OPENCHAIN

OPENCHAIN Compliance Program Specification

Version 2.0 (DRAFT) 2019.03.02

Trusting the Open Source from which Software Solutions are built

***DRAFT**: This is the draft of the next version 2.0 of the OpenChain specification. Changes made over the previous 1.2 version can be found in a special marked up version of this draft. Section 0 provides a high level summary of changes made. The most recent updates made to the text that have been generally accepted can be found in blue highlights. Recommended updates still under discussion can be found in yellow highlights. We are targeting to release a new version of the specification in April 2019.*

Contents

0) What Changed Log (temporary section)

We provide a summary of changes made to version 1.2 of the Specification. This is a temporary section that will be removed in the final version. The updates include:

- Changed version number from 1.2 to 2.0
- Overhauled the Introduction section to make it more concise and up to date.
- Overhauled the training section to move from a more prescriptive to a less prescriptive requirement. Now gives more flexibility to an organization to define the roles and responsibilities and how to ensure those fulfilling the roles are properly trained. We replaced the term training with the two concepts used by the ISO 9001:2015 standard: Competency and Awareness.
- Removed section title terms for Goal 1, Goal 2, … Goal 6. They didn't represent goals as much as they represented facets of a program.
- Standardized on the term Open Source. The previous specification uses both terms "Open Source" and the "FOSS (Free and Open Source)" interchangeable. It was acknowledged that the term Open Source is more widely recognized and understood for the following reasons: i) Some users of the current specification have pointed out there was confusion between the term Open Source Software and FOSS. This was particularly true for first time readers; ii) Open Source represents a superset (inclusive); iii) The large majority of major foundations use the de facto term Open Source (*e.g.*, Apache, Eclipse, OSI, Linux, OpenStack, Cloud Foundry, …); iv) Most commercial organizations externally use the de facto term Open Source; v) It is also consistent with the term "Open Source Program Office" which represents a major target audience of the specification; vi) The term Open Source is de facto in Asia, largely used in North America and mixed yet dominant in Europe; Translations would be simplified by the use of a single term;
- Program Scope Declaration (section 1.4) was added. Provides the flexibility to construct a compliance program that best fits the scope of a given organization's needs. Some organizations could choose to maintain a compliance program for a specific product line while others could choose the program scope to govern software releases of the entire organization. Large organizations may prefer the former example while smaller organizations may prefer the latter.
- Changes Section 2 to deemphasize specific roles and emphasize expected tasks to be performed instead. For instance the Open Source Liaison role was removed. Because we are moving in the direction of ISO 9001, we needed to make the spec less prescriptive. However the responsibility remains which are described in requirements 2.1 and 2.2.
- Update the definition for Identified Licenses to make clear that the supplied software can range in open source composition from 0% to 100% open source.
- Updated requirement 3.1 text to make clear the bill of materials consist of open source components.
- Added a tag line to the title page to succinctly describe the essence of the spec in one sentence.
- Minor edits were made throughout the document to improve readability.

1) Introduction

This specification defines the key requirements of a quality open source compliance program (Program). The objective is to provide a benchmark that builds trust between organizations exchanging software solutions that are comprised of open source software. Specification conformance provides assurance that a Program has been designed to produce the required Compliance Artifacts (*i.e.*, legal notices, source code and so forth) for each software solution. The OpenChain Specification focuses on the "what" and "why" aspects of a Program rather than the "how" and "when". This ensures flexibility for different organizations of different sizes in different markets to choose specific policy and process content that fits their size, goals and scope. For instance, an OpenChain Conformant program may address a single product line or the entire organization.

This introduction provides the context for all potential users. Section 2 defines key terms used throughout Specification. Section 3 defines the requirements that a Program must satisfy to achieve conformance. A requirement consists of one or more Verification Materials (*i.e.*, records) that must be produced to satisfy the requirement. Verification Materials are not required to be made public, though organizations may choose to provide them to others, potentially under a Non-Disclosure Agreement (NDA).

The Specification is developed as an open initiative with feedback received from over 150 contributors. Insight into its historical development can be obtained by reviewing the Specification mailing list and Frequently Asked Questions (FAQs).

2) Definitions

"Compliance Artifacts" - a collection of artifacts that represent the output of the Program for the Supplied Software. The collection may include (but are not limited to) one or more of the following: source code, attribution notices, copyright notices, copy of licenses, modification notifications, written offers, Open Source component bill of materials, and SPDX documents.

"Identified Licenses" - a set of Open Source Software licenses identified as a result of following an appropriate method of identifying Open Source components from which the Supplied Software is comprised.

"OpenChain Conformant" - a Program that satisfies all the requirements of this specification.

"Open Source" - software subject to one or more licenses that meet the Open Source Definition published by the Open Source Initiative (OpenSource.org) or the Free Software Definition (published by the Free Software Foundation) or similar license.

"**Program**" – the set of policies, processes and personnel that manage an organization's Open Source compliance activities.

Software Staff - any organization employee or contractor that defines, contributes to or has responsibility for preparing Supplied Software. Depending on the organization, that may include (but is not limited to) software developers, release engineers, quality engineers, product marketing and product management.

SPDX - the format standard created by the Linux Foundation's SPDX (Software Package Exchange) Working Group for exchanging license and copyright information for a given software package. A description of the SPDX specification can be found at www.spdx.org.

Supplied Software - software that an organization distributes to third parties (*e.g.*, other organizations or individuals).

Verification Materials - materials that demonstrate that a given requirement is satisfied.

3) Requirements

1.0 Program Foundation

1.1 Policy

A written Open Source policy exists that governs Open Source license compliance of the Supplied Software. The policy must be internally communicated.

Verification Material(s):

- ☐ 1.1.1 A documented Open Source policy.
- ☐ 1.1.2 A documented procedure that makes Software Staff aware of the existence of the Open Source policy (*e.g.*, via training, internal wiki, or other practical communication method).

Rationale:
To ensure steps are taken to create, record and make Software Staff aware of the existence of an Open Source policy. Although no requirements are provided here on what should be included in the policy, other sections may impose requirements on the policy.

1.2 Competence

The organization shall:

- **Identify the roles and the corresponding responsibilities of those roles that affects the performance and effectiveness of the Program;**
- **Determine the necessary competence of person(s) fulfilling each role**
- **Ensure that these persons are competent on the basis of appropriate education, training, and/or experience;**
- **Where applicable, take actions to acquire the necessary competence; and**
- **Retain appropriate documented information as evidence of competence.**

Verification Material(s):

- ☐ 1.2.1 A documented list of roles with corresponding responsibilities for the different participants in the Program.
- ☐ 1.2.2 A document that identifies the competencies for each role.
- ☐ 1.2.3 Documented evidence of assessed competence for each Program participant.

Rationale:
To ensure that the identified participants fulfilling Program roles have obtained a sufficient level of competence for their respected roles and responsibilities.

1.3 Awareness

The organization shall ensure that Program participants are aware of:

a) The Open Source policy;
b) Relevant Open Source objectives;

c) Their contribution to the effectiveness of the Program; and
d) The implications of not following the Program's requirements.

Verification Material(s):

- ☐ 1.3.1 Documented evidence of assessed awareness for each Program personnel including the Program's objectives, ones contribution within the Program and implications of Program non-conformance.

Rationale:
To ensure Program personnel have obtain a sufficient level of awareness for their respected roles and responsibilities within the Program.

1.4 Program Scope

Different Programs may be governed by different levels of scope. For example, a program could govern a single product line, an entire department or an entire organization. The scope designation needs to be declared for each Program.

Verification Material(s):

- ☐ 1.4.1 A written statement that clearly defines the limits and scope of the Program.

Rationale:
To provide the flexibility to construct a Program that best fits the scope of an organization's needs. Some organizations could choose to maintain a Program for a specific product line while others could implement a Program to govern the Supplied Software of the entire organization.

1.5 License Obligations

A process exists for reviewing the Identified Licenses to determine the obligations, restrictions and rights granted by each license.

Verification Material(s):

- ☐ 1.5.1 A documented procedure to review and document the obligations, restrictions and rights granted by each Identified License.

Rationale:
To ensure a process exists for reviewing and identifying the license obligations for each Identified License for the various use cases an organization may encounter.

2.0 Relevant Tasks Defined and Supported

2.1 Maintain a process to effectively respond to external Open Source inquiries. Publicly identify a means by which a third party can make an Open Source compliance inquiry.

Verification Material(s):

- ☐ 2.1.1 Publicly visible method that allows any third party **to make an Open Source license compliance inquiry** (*e.g.*, via a published contact email address, or the Linux Foundation's Open Compliance Directory).
- ☐ 2.1.2 An internal documented procedure for responding to third party Open Source license compliance inquiries.

Rationale:
To ensure there is a reasonable way for third parties to contact the organization with regard to Open Source compliance inquiries and that the organization is prepared to effectively respond.

2.2 Identify and Resource Program Task(s).

- **Assign accountability to ensure the successful execution of Program tasks.**
- **Program tasks are sufficiently resourced:**
  - **Time to perform the tasks have been allocated; and**
  - **Adequate funding has been allocated.**
- **A process exists for reviewing and updating the policy and supporting tasks;**
- **Legal expertise pertaining to Open Source license compliance is accessible to those who may need such guidance; and**
- **A process exists for the resolution of Open Source license compliance issues.**

Verification Material(s):

- ☐ 2.2.1 Document with name of persons, group or function in Program role(s) identified.
- ☐ 2.2.2 The identified Program roles have been properly staffed and adequate funding provided.
- ☐ 2.2.2 Identification of legal expertise available to address Open Source license compliance matters which could be internal or external.
- ☐ 2.2.3 A documented procedure that assigns internal responsibilities for Open Source compliance.
- ☐ 2.2.4 A documented procedure for handling the review and remediation of non-compliant cases.

Rationale:
To ensure: i) Program responsibilities are effectively supported and resourced and ii) policies and supporting processes are regularly updated to accommodate changes in Open Source compliance best practices.

3.0 Open Source Content Review and Approval

3.1 A process exists for creating and managing a bill of materials that includes each Open Source component (and its Identified Licenses) from which the Supplied Software is comprised.

Verification Material(s):

- ☐ 3.1.1 A documented procedure for identifying, tracking, reviewing, approving and archiving information about the collection of Open Source components from which the Supplied Software is comprised.
- ☐ 3.1.2 Open Source component records for the Supplied Software that demonstrates the documented procedure was properly followed.

Rationale:
To ensure a process exists for creating and managing a Open Source component bill of materials used to construct the Supplied Software. A bill of materials is needed to support the systematic review and approval of each component's license terms to understand the obligations and restrictions as it applies to the distribution of the Supplied Software.

3.2 The Program must be capable of managing common Open Source license use cases encountered by Software Staff for Supplied Software, which may include the following use cases (note that the list is neither exhaustive, nor may all of the use cases apply):

- **distributed in binary form;**
- **distributed in source form;**
- **integrated with other Open Source such that it may trigger copyleft obligations;**
- **contains modified Open Source;**
- **contains Open Source or other software under an incompatible license interacting with other components within the Supplied Software; and/or**
- **contains Open Source with attribution requirements.**

Verification Material(s):

- ☐ 3.2.1 A documented procedure for handling the common Open Source license use cases for the Open Source components of the Supplied Software.

Rationale:
To ensure the program is sufficiently robust to handle an organization's common Open Source license use cases. That a procedure exists to support this activity and that the procedure is followed.

4.0 Compliance Artifact Creation and Delivery

4.1 A process exists for creating the set of Compliance Artifacts for the Supplied Software.

Verification Material(s):

- ☐ 4.1.1 A documented procedure that documents the process under which the Compliance Artifacts are prepared and distributed with the Supplied Software as required by the Identified Licenses.

- ☐ 4.1.2 A documented procedure for archiving copies of the Compliance Artifacts of the Supplied Software - where the archive is planned to exist for a reasonable period of time since the last offer of the Supplied Software or as required by the Identified Licenses (whichever is longer). Records exist that demonstrate the procedure has been properly followed.

Rationale:

To institute reasonable commercial efforts in the complete collection of Compliance Artifacts accompany the Supplied Software as required by the Identified Licenses.

5.0 Understand Open Source Community Engagement

5.1 A written policy exists that governs contributions to Open Source projects by the organization. The policy must be internally communicated.

Verification Material(s):

- ☐ 5.1.1 A documented Open Source contribution policy.
- ☐ 5.1.2 A documented procedure that makes all Software Staff aware of the existence of the Open Source contribution policy (*e.g.*, via training, internal wiki, or other practical communication method).

Rationale:
To ensure an organization has given reasonable consideration to developing a policy with respect to publicly contributing to Open Source. The Open Source contribution policy can be made a part of the overall Open Source policy of an organization or be its own separate policy. In the situation where contributions are limited or not permitted at all, a policy should exist making that position clear.

5.2 If an organization permits contributions to Open Source projects then a process exists that implements the Open Source contribution policy outlined in Section 5.1.

Verification Material(s):

- ☐ 5.2.1 Provided the Open Source contribution policy permits contributions, a documented procedure that governs Open Source contributions.

Rationale:
To ensure an organization has a documented process for how the organization publicly contributes Open Source. A policy may exist such that contributions are not permitted at all. In that situation it is understood that no procedure may exist and this requirement would nevertheless be met.

6.0 Verify Adherence to OpenChain Requirements

6.1 In order for a Program to be deemed OpenChain Conforming Program, the organization must affirm that the program satisfies the requirements presented in this specification.

Verification Material(s):

- 6.1.1 A document affirming the Program designated in requirement 1.4 satisfies all the requirements of this document, OpenChain Specification version 2.0.

Rationale:
To ensure that if an organization declares that it has a program that is OpenChain Conforming, that such program has met all the requirements of this specification. The mere meeting of a subset of these requirements would not be considered sufficient.

6.2 A Program that is OpenChain Conformant with this version of the specification will last 18 months from the date conformance validation was obtained. The conformance validation registration procedure can be found on the OpenChain project's website.

Verification Material(s):

- 6.2.1 A document affirming the Program meets all the requirements of this document, OpenChain Specification version 2.0, within the past 18 months of obtaining conformance validation.

Rationale:
It is important for the organization to remain current with the specification if that organization wants to assert program conformance over time. This requirement ensures that the program's supporting processes and controls do not erode if an organization continues to assert program conformance over time.

Appendix I: Language Translations

To facilitate global adoption we welcome efforts to translate the specification into different languages. Because OpenChain functions as an open source project translations are driven by those willing to contribute their time and expertise to perform translations under the terms of the CC-BY 4.0 license and the project's translation policy. The details of the policy and available translations can be found on the OpenChain project specification webpage.