OPENCHAIN

OpenChain Conformance 2016-H1 Specification

Contents

Introduction

The OpenChain Initiative began in 2013 when a group of software supply chain open source practitioners observed two emerging patterns: 1) significant process similarities existed among organizations with mature open source compliance programs; and 2) there still remained a large number of organizations exchanging software with less developed programs. The latter observation resulted in a lack of trust in the consistency and quality of the compliance artifacts accompanying the software being exchanged. As a consequence, at each tier of the supply chain, downstream organizations were frequently redoing the compliance work already performed by other upstream organizations.

A study group was formed to consider whether a standard program specification could be created that would: i) facilitate greater quality and consistency of open source compliance information being shared across the industry; and ii) decrease the high transaction costs associated with open source resulting from compliance rework. The study group evolved into a work group, and in April 2016, formally organized as a Linux Foundation collaborative project.

The Vision and Mission of the OpenChain Initiative are as follows:

- **Vision**: A software supply chain where free/open source software (FOSS) is delivered with trusted and consistent compliance information.

- **Mission**: Establish requirements to achieve effective management of free/open source software (FOSS) for software supply chain participants, such that the requirements and associated collateral are developed collaboratively and openly by representatives from the software supply chain, open source community, and academia.

In accordance with the Vision and Mission, this specification defines a set of requirements that if met, would significantly increases the probability that an open source compliance program had achieved a sufficient level of quality, consistency and completeness; although a program that satisfies all the specification requirements does not guarantee full compliance. The requirements represent a base level (minimum) set of requirements a program must satisfy to be considered OpenChain Conforming. The specification focuses on the “what” and “why” qualities of a compliance program as opposed to the “how” and “when” considerations. This ensures a practical level of flexibility that enables different organizations to tailor their policies and processes to best fit their objectives.

Section 2 introduces definitions of key terms used throughout the specification. Section 3 presents the specification requirements where each one has a list of one or more Verification Artifacts. They represent the evidence that must exist in order for a given requirement to be considered satisfied. If all the requirements have been met for a given program, it would be considered OpenChain Conforming in accordance with version 2016-H1 of the specification.

Definitions

Distributed Compliance Artifacts - the set of artifacts that an Identified License requires be provided with Supplied Software. They include (but are not limited to) the following: copyright notices, copies of licenses, modification notifications, attribution notices, source code, written offers and so forth.

FOSS (Free and Open Source Software) - 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.

FOSS Liaison - a designated person who is assigned to receive external FOSS inquires.

Identified Licenses - a set of FOSS licenses identified as a result of following an appropriate method of identifying such licenses.

OpenChain Conforming – a program that satisfies all the requirements of this specification.

Software Staff - any 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 or Software Package Data Exchange – the format standard created by the SPDX Working Group for exchanging license and copyright information for a given software package. A description of the SPDX specification can be found at ww.spdx.org.

Supplied Software – software that an organization delivers to third parties (e.g., other organizations or individuals).

Verification Artifacts - evidence that must exist in order for a given requirement to be considered satisfied.

Requirements

G1: Know Your FOSS Responsibilities

1.1 A written FOSS policy exists that governs FOSS license compliance of the Supplied Software distribution where, as a minimum, it must be internally communicated.

Verification Artifact(s):

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

Rationale:
Ensure steps were taken to create, record and make Software Staff aware of the existence of a FOSS policy. Although no requirements are provided here on what should be included in the policy, other requirements in other sections may.

1.2 Mandatory FOSS training for all Software Staff exists such that:

- **The training, as a minimum, covers the following topics:**
  - **The FOSS policy and where to find a copy;**
  - **Basics of IP law pertaining to FOSS and FOSS licenses;**
  - **FOSS licensing concepts (including the concepts of permissive and copyleft licenses);**
  - **FOSS project licensing models;**
  - **Software Staff roles and responsibilities pertaining to FOSS compliance specifically and the FOSS policy in general; and**
  - **Process for identifying, recording and/or tracking of FOSS components contained in Supplied Software.**
- **Software Staff must have completed FOSS training within the last 24 months (to be considered current). A test may be used to allow Software Staff to satisfy the training requirement.**

Verification Artifact(s):

- ☐ 1.2.1 FOSS course materials covering the above topics exists (e.g., slide decks, online course, or other training materials).
- ☐ 1.2.2 Method of tracking the completion of the course for all Software Staff.
- ☐ 1.2.3 At least 85% of all Software Staff are current, as per definition in above section.

Rationale:
Ensure the Software Staff have recently attended FOSS training and that a core set of relevant FOSS topics are covered. The intent is to ensure a core base level set of topics are covered but a typical training program would likely be more comprehensive than what is required here.

G2: Assign Responsibility for Achieving Compliance

2.1 Identify FOSS Liaison Function ("FOSS Liaison").

- **Assign individual(s) responsible for receiving external FOSS inquiries;**
- **FOSS Liaison must make commercially reasonable efforts to respond to FOSS compliance inquiries as appropriate; and**
- **Publicly identify means of contacting the FOSS Liaison by way of electronic communication.**

Verification Artifact(s):

- ☐ 2.1.1 FOSS Liaison function is publicly identified (e.g., via an email address and/or the Linux Foundation's Open Compliance Directory).
- ☐ 2.1.2 A documented procedure exists that assigns responsibility for receiving FOSS compliance inquiries.

Rationale:
Ensure there is a reasonable way for third parties to contact the organization with regard to FOSS compliance inquiries.

2.2 Identify Internal FOSS Compliance Role(s).

- **Assign individual(s) responsible for managing internal FOSS compliance. The FOSS Compliance role and the FOSS Liaison can be the same individual.**
- **FOSS compliance management activity is sufficiently resourced:**
  - **Time to perform the role has been allocated; and**
  - **Commercially reasonable budget has been allocated.**
- **Assign responsibilities to develop and maintain FOSS compliance policy and processes;**
- **Legal expertise pertaining to FOSS compliance is accessible to the FOSS Compliance role (e.g., could be internal or external); and**
- **Escalation path is available for resolution of FOSS compliance issues.**

Verification Artifact(s):

- ☐ 2.2.1 Name of persons, group or function in FOSS Compliance role(s) identified.
- ☐ 2.2.2 Identify source of legal expertise available to FOSS Compliance role(s).
- ☐ 2.2.3 A documented procedure exists that assigns responsibilities for FOSS compliance.
- ☐ 2.2.4 A documented procedure exists that identifies an escalation path for issue resolution.

Rationale:
Ensure certain FOSS responsibilities have been effectively assigned.

G3: Review and Approve FOSS Content

3.1 A process exists for identifying, tracking and archiving a list of all FOSS components (and their respective Identified Licenses) from which Supplied Software is comprised.

Verification Artifact(s):

- ☐ 3.1.1 A documented procedure exists used to identify, track, and archive a list of FOSS components and their Identified Licenses from which the Supplied Software is comprised.

Rationale:
To ensure a process exists for identifying and listing all FOSS components used to construct the Supplied Software. This inventory must exist to support the systematic review of each component's license terms to understand their respective distribution obligations and restrictions applicable to the Supplied Software. The recorded inventory also serves as evidence that the process was followed.

3.2 The FOSS program must be capable of handling typical FOSS use cases encountered by Software Staff for Supplied Software, which may include the following use cases - when parts of the Supplied Software (note that the below list is neither exhaustive, nor may all of the below use cases apply depending on the organization):

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

Verification Artifact(s):

- ☐ 3.2.1 A process has been implemented that is capable of addressing the typical FOSS use cases encountered by Software Staff for Supplied Software.

Rationale:
To cause the FOSS program to be sufficiently robust to address that organization's typical use cases as a result of that organization's business practices.

G4: Deliver FOSS Content Documentation and Artifacts

4.1 Prepare the following Distributed Compliance Artifacts to accompany the Supplied Software as required by the corresponding Identified Licenses which might include (but is not limited to) the required:

- **copyright notices**
- **copies of Identified Licenses**
- **modification notifications**
- **attribution notices**
- **prominent notices**
- **source code**
- **required build instructions and scripts**
- **written offers**

Verification Artifact(s):

- ☐ 4.1.1 A documented procedure exists describing a process that ensures the Distributed Compliance Artifacts be distributed with Supplied Software as required by the Identified Licenses.
- ☐ 4.1.2 Copies of the Distributed Compliance Artifacts of the Supplied Software are archived and easily retrievable (e.g., legal notices, source code, SPDX documents), and the archive is planned to exist for at least as long as the Supplied Software is offered or as required by the Identified Licenses (whichever is longer).

Rationale:
Ensure the complete collection of compliance artifacts accompany the Supplied Software as required by the Identified Licenses that govern the Supplied Software.

G5: Understand FOSS Community Engagement

5.1 A written policy exists that governs contributions to publicly accessible FOSS projects by employees on behalf of the organization where, as a minimum, it must be internally communicated.

Verification Artifact(s):

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

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

5.2 Provided the FOSS contribution policy permits such contributions, a process exists for confirming contributions adhere to the FOSS contribution policy, which might include (but is not limited to) the following considerations:

- **legal approval for license considerations**
- **business rationale or approval**
- **technical review of code to be contributed**
- **community engagement and interaction, including a project's Code of Conduct or equivalent**
- **adherence to project-specific contribution requirements**

Verification Artifact(s):

- ☐ 5.2.1 Provided the FOSS contribution policy permits contributions, a documented procedure exists that describes the FOSS contribution process.

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

G6: Certify Adherence to OpenChain Requirements

6.1 In order for an organization to be OpenChain certified, it must affirm that it has a FOSS program that meets the criteria described in this OpenChain Conformance 2016-H1 Specification.

Verification Artifact(s):

- ☐ 6.1.1 The organization affirms that a program exists that meets all the requirements of this OpenChain Conformance 2016-H1 Specification.

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 to warrant a program be OpenChain certified.