

Open Source Guidelines

Cisco Suppliers

A printed or duplicate soft copy is considered uncontrolled. Refer to the original online version for the latest revision.
Based on Template Version 8

Table of Contents

Open Source Guidelines 1
1. Purpose 3
2. Overview 3
3. Scope 4
4. Policy Statements 4
4.1 Open Source Compliance Materials 4
5. Technical Instructions (How to Comply with these Guidelines) 9

1. Purpose

These ***Open Source Guidelines for Cisco Suppliers*** ("Guidelines") specify what procedures must be followed, and what information and materials must be provided, when a supplier provides any deliverable to Cisco that may include Open Source Software or that requires Open Source Software for the use of the deliverable (i.e. a supplier deliverable which relies on dependencies not included in the deliverable). For example, in these common situations:

2. Overview

For purposes of these Guidelines, by "Open Source Software" we mean any technologies, and all of the IP rights therein, that are subject to the terms of an "open source license." An open source license is generally any license that will allow anyone to copy, modify, and distribute software for free. Examples of open source licenses include all versions of: the BSD license (BSD), the MIT license (MIT), the Mozilla Public License (MPL), the Eclipse Public License (EPL), and the Apache Software License (ASL), as well as all of the licenses sponsored by the Free Software Foundation, such as all versions of the GNU General Public License GPL), the GNU Lesser General Public License (LGPL), and the GNU Affero General Public License (AGPL). Open source licenses can apply to both source and binary/object code forms of the technology.

As a Cisco supplier, it's important that you understand all of the open source technology in your products and solutions, regardless of how it was included. You may have included Open Source Software in your product intentionally, such as by using open source components, packages, libraries, files, scripts, or snippets of code, or less intentionally, such as by using open source tools, open standards, or open APIs. You may also have Open Source Software in your product simply because one of your suppliers included open source. Another consideration is that some open source licenses apply their terms to modifications and additions that you make to the original code, and certain licenses – such as the GPL and the LGPL – may apply their terms to code that links to, is used by, or interacts with the original code. For the purposes of these Guidelines, all such code is considered Open Source Software.

3. Scope

Cisco values the open source community as a resource and partner in innovation, whose works must be respected and used appropriately. While Open Source Software allows us to develop new products quickly and to leverage widely accepted platforms and standards, we recognize that Open Source Software also comes with responsibilities, obligations, and some risk.

Cisco intends to comply with its open source license obligations, manage risk in a sensible manner, satisfy the needs of our customers, and be a positive contributor back to the open source community. To this end, we have enacted policies and procedures that our engineers and product teams must follow when creating or interacting with products, solutions, and services that could contain Open Source Software. For this to work at scale, our suppliers must align with this approach as well, as they are placing obligations and risks directly on Cisco, typically without our visibility or control. These Guidelines exist solely to ensure that we have what is necessary to address these requirements.

Due to the critical nature of this task, Cisco will monitor each supplier's compliance with these Guidelines. Failure to comply may breach the supplier's agreement with Cisco and cause a supplier to be removed from Cisco's Approved Vendor List or otherwise be eliminated from consideration for future business with Cisco.

4. Policy Statements

All suppliers shall provide the following to Cisco:

4.1 Open Source Compliance Materials

The Open Source Compliance Materials enable Cisco to meet its obligations as required by each open source license that covers any technology in your deliverables, including the obligations to publish copyright notices, license notices, copies of licenses, and source code. You must provide *all* compliance materials that are mandated by the license(s) for each piece of open source technology in your deliverables or that requires Open Source Software for the use by the deliverable (i.e a supplier deliverable which relies on dependencies not included in the deliverable). The timeline is set forth in the agreement between the parties but the default period is a minimum net 15 day prior to the release of the Cisco product **.**

It is <u>your</u> obligation to review each license and ensure that all materials required for complete compliance (including corresponding notices and source code) are provided to Cisco. Please consult with your Legal representative if you need help understanding your obligations.

The **Open Source Compliance Materials** will be composed of **two** elements:

1. **Compliance Documentation**

The required Compliance Documentation includes the corresponding copyright and license notices and copies of open source licenses as mandated by the open source licenses applicable to your product deliverables. For example, this may include:

a. Copyright and license notices found in the source code for BSD, MIT, and similarly licensed open source components; *and*
b. Copies of the AGPL, GPL, LGPL, and other open source licenses that mandate inclusion of a copy of the license with a distribution.

2. **Compliance Source Code**

The Compliance Source Code is the complete and corresponding source code for all Open Source Software under a license that requires mandatory publication of source code (for example, the GPL, LGPL, MPL, EPL, and other licenses with similar obligations). Depending on the open source license, the source code may include your modifications, scripts, build and install information (toolchain used, encryption keys, etc.), and non-open source binary objects, depending on the obligations of each license.

Note: If you do not have any Open Source Software under a license that requires mandatory publication of source code ("Required Source Code License") this requirement does not apply to you.

See the Technical Instructions at the end of these Guidelines for more detail on how to properly assemble the Open Source Compliance Materials.

3. Open Source Verification Materials

The Open Source Verification Materials are required by Cisco's internal open source process and tooling, and are used to substantiate the accuracy of the Open Source Compliance Materials described above.

a. The **Open Source Verification Materials**
   - **a.** Verification Source Code provided by you can be either:
     - **i.** the original, unmodified source code for each open source component utilized in the deliverables, as you originally acquired it (before modification or integration); *or*
     - **ii.** The current version of the source code for each open source component, as actually being utilized in the product deliverables (which may include your modifications).

In other words, you may provide the original unmodified source code for an open source component if you choose not to share any modifications or changes that you have made to it unless the open source license requires that you provide the source code of the modifications. In the alternative, you may provide the current version of the source code for an open source component if the original version is no longer obtainable, or you have made no proprietary modifications to it. Please choose whichever option is most convenient for you.

Note: This Verification Source Code differs from the Compliance Source Code above in that the Verification Source Code requirement applies to *all* open source technology, not just the open source under a Required Source Code License, and it is for informational rather than compliance purposes. For example, for the Verification Source Code you must provide the source for a BSD-licensed open source component included in your product, even though the BSD license does not require mandatory publication of such code for license compliance, because our management system requires it in order to substantiate certain information about each component (including applicable licenses, copyright holders, notices, etc.) and perform our internal review. Additionally, the Compliance Source Code may include some materials beyond strictly source code as the Verification Source Code that are required for proper open source license compliance (described above), which is not required under these open source licenses.

4. Managing External Requests

a. Requests for Source Code

If Cisco receives a request from a third party to provide source code related to your deliverables due to an alleged open source license obligation, and such source code was not already provided to Cisco (a "request"), Cisco will notify you and refer the requestor to you. You must respond to the request within five (5) business days of notification by providing Cisco either:

i. the requested source code, if you are obligated to provide all or part of the code under the terms of an open source license or your agreement with Cisco; *or*

ii. A written explanation describing in sufficient technical detail why you reasonably believe that the requested source code is not subject to release.

If, despite providing the above, the request should remain unresolved, you agree to meet with Cisco within five (5) business days of Cisco's written request in order to discuss how you plan to address the request. At this time, Cisco will have the option to require that you provide a Build Archive (described below), for the sole purpose of helping to evaluate the unresolved request.

b. Build Archive

If a request still remains unresolved after following the procedure outlined above, Cisco may request that you provide a Build Archive to help evaluate the situation. A Build Archive contains the additional materials that would be sufficient to build the product image using an accurate Build Log and Compliance Source Code bundle.

A proper Build Archive does not need to include any proprietary source code, only the binary objects necessary to build the product image from the open source code that you are required to publish. The purpose of the Build Archive is only to assist in substantiating the accuracy of the other materials that you've provided under these Guidelines (for example, your Compliance Source Code bundle), and only when faced with an unresolved external allegation that those materials are incorrect or incomplete. In such a situation, the Build Archive would help you demonstrate the completeness and accuracy of what you've provided.

You are granting no further rights or permissions to the Build Archive materials or the resulting product image.

If one is ever needed, please see the Technical Instructions at the end of these Guidelines for more detail on how to properly assemble a Build Archive.

5. Updates

When Open Source Software in any deliverables provided to Cisco is changed or updated in new versions of such deliverable, you must promptly notify Cisco and ensure that Cisco has accurate and up-to-date versions of all the materials required by these Guidelines, including component information, notices, and source code. Likewise, if any Open Source Software is added or removed, Cisco must be promptly notified and provided the proper corresponding materials as required under these Guidelines. In other words, the materials we've obtained from you must always correspond to the current state of the Open Source Software in your deliverables. The timeline for updates is set forth in the contract but the default period is within 30 days of the event that necessitated the update.

5. Technical Instructions (How to Comply with these Guidelines)

General Instructions

Reference your specific agreement for the exact date each required item under these Guidelines is due

Please work with your Cisco Contact to provide and maintain the text files for **Compliance Documentation (Section 2.A)** and deposit the **Verification Source Code (Section 3.A)**. These must be provided together in order to allow Cisco to import this data into our system. Please follow the step-by-step instructions included in the template when assembling this import archive.

After delivering these materials, Cisco may contact you to discuss how the Open Source Software is used if this information is not evident from the materials alone. For example, you might be asked whether Open Source Software was linked in a certain way where such detail cannot be determined by Cisco from just your Guideline and product deliverables.

How to assemble the Compliance Source Code (Section 2.B)

Along with the complete and corresponding source code (and other materials, as required) for the Compliance Source Code, you must provide a top-level README file at the root of the Compliance Source Code bundle that contains instructions (such as an ordered list of Unix shell commands) for building the source code into installable object code. This README file should also contain a table of requirements identifying the name, version number, and origin (such as a URL) for each software package required for compilation, installation, and execution of the code in the Compliance Source Code bundle, if those packages are not themselves present. For example, a Compliance Source Code bundle containing only a “Hello, World!” program source file (“hello.c”) and a simple Make file for generating an executable program, might provide a requirements table like the following:

Package Name	Version	Origin
GNU Compiler Collection (gcc)	4.4.1	http://gcc.gnu.org/gcc-4.4/
GNU binutils	2.19.1	http://www.gnu.org/software/binutils/
GNU C Library	2.10.1	http://www.gnu.org/software/libc/
GNU Make	3.81	http://www.gnu.org/software/make/

If the software package required for development of the code in the Compliance Source Code originated from a distributor (such as Red Hat Enterprise Linux, Debian GNU/Linux, or Ubuntu), then it is acceptable to prepare the table using the output of the corresponding package management commands, such as “rpm - qa” for RPM-based systems or “dpkg-l” for DEB-based systems. Please prune such lists of all software packages that are *not* required.

If a required package is proprietary and available only on a commercial basis, then the name of the licensor or distributor, along with a URL, contact address (email or traditional, or both), and/or phone number.

Note: If you have modified any of the required technology such that the modifications are required to correctly compile, install, or execute any of the code in the Compliance Source Code, then (if you have a license to distribute the modified versions) you must provide the modified versions, in a subdirectory of the root called “REQUIREMENTS”, which should in turn contain one archive file or unpacked directory of modified source code for each modified requirement. If you do not have license to distribute the requirements in modified source form, then you should note their modified status by placing the word “MODIFIED” after the version number in the “README” file. The README file should be in plain-text format.

How to assemble a Build Log (Section 3.B)

In order to verify the completeness of the Compliance Source Code bundle, you must provide a Build Log for that code. An acceptable Build Log must satisfy the following list of requirements (we provide Unix shell commands as examples to guide you in developing your own procedures):

- **Demonstrate as part of the log a standard, secure cryptographic hash of the archive:**
  - md5sum myarchive.tar.gz, *or*
  - shasum –a 256 myarchive.tar.gz
- **Demonstrate the unpacking of the archive:**
  - tar xzvf myarchive.tar.gz
- **Demonstrate the build process:**
  - cd myarchive
  - ./configure
  - make
- **Demonstrate the installation of the archive results (not necessarily at device level, but at least at filesystem level):**
  - make install
- **Demonstrate as part of the log the standard, secure cryptographic hash of the resultant artifact(s), as for example:**
  - find $INSTALL_DIR -print | xargs md5sum, *or*
  - find $INSTALL_DIR –print | xargs shasum –a 256

- **If you are providing Cisco with a product image, demonstrate building the product image with the results of the Compliance Source Code build.**
- **If you are providing Cisco with a product image, demonstrate or provide as part of the log the standard, secure cryptographic hash of the resultant product image.**
- **Include the complete output of the above demonstrations.**

How to assemble a Build Archive (Section 4.B)

If required, a Build Archive will contain sufficient additional materials to allow Cisco to follow the Build Log and build the resulting product image using the materials provided in your Compliance Source Code bundle. This can achieve in either of the two following ways:

i. The simplest and recommended way to achieve this is to provide a virtual machine, executable by VMWare Player (or similar) and containing all necessary tools, Compliance Source Code, binary objects for vendors' proprietary technology, and instructions for a simple entry point to build the image from the Compliance Source Code and the provided binary objects. (For example, cd $START_DIR; make; make install). The instructions should be provided in a file named README at the root directory of the filesystem of the VM; *or*

ii. If you do not wish to provide such a VM, you should provide all necessary tools, binary objects, supporting libraries, instructions, and so forth such that Cisco can replicate the build of the product image using the Compliance Source Code.

Questions?

Your Cisco Contact is available to point you in the right direction. Please get in touch with him or her if you need further guidance.