**Open-Source Compliance**

Table of Contents

[Open-Source Software 3](#_Toc83393798)

[OSI Approved OSS Licenses 4](#_Toc83393799)

[Permissive and Copyleft Licenses 5](#_Toc83393800)

[Non-Traditional Licenses 5](#_Toc83393801)

[Compliance 7](#_Toc83393802)

[Objectives 8](#_Toc83393803)

[Benefits 8](#_Toc83393804)

[Consequences of Violations 9](#_Toc83393805)

[Compliance Management 11](#_Toc83393806)

[Identification 12](#_Toc83393807)

[Audit 13](#_Toc83393808)

[Resolve Issues 13](#_Toc83393809)

[Review 14](#_Toc83393810)

[Approvals 14](#_Toc83393811)

[Registration 14](#_Toc83393812)

[Notices 15](#_Toc83393813)

[Pre-Distribution Verification 15](#_Toc83393814)

[Distribution 15](#_Toc83393815)

[Final Verification 16](#_Toc83393816)

[General Guidelines 17](#_Toc83393817)

## Open-Source Software

**Open-Source Software** (OSS), also called Free Open-Source Software (FOSS), is software that is publicly available. They can be viewed by anyone, but to use them, we must comply with some rules.

An **OSS License** is a license for OSS that allows us to use it. It is a **legal and binding contract** between the author and user, declaring that the content can be used in commercial applications under certain conditions.

An OSS License allows us to use, modify and/or share the OSS under defined terms and conditions. In general, OSS licenses allow for modification and redistribution **without having to pay** the original author.

OSS Licenses may impact:

* Use of the software
* Modification of the software
* Maintenance of the software
* Distribution of the software and its derivatives
* Intellectual property rights (IPR)

On the other hand, it may permit:

* Modification of the source code
* Recompilation of the software
* Redistribution of the original source code, modified source code and/or
* binaries
* Integration of the software with proprietary software
* Redistribution of the resulting software as part of, or within, proprietary products

### OSI Approved OSS Licenses

A popular set of OSS Licenses are approved by the **Open-Source Initiative** (OSI), which is based on the **Open-Source Definition** (OSD):

* Apache License 2.0
* BSD 3-Clause "New" or "Revised" license
* BSD 2-Clause "Simplified" or "FreeBSD" license
* GNU General Public License (GPL)
* GNU Library or "Lesser" General Public License (LGPL)
* MIT license
* Mozilla Public License 2.0
* Common Development and Distribution License
* Eclipse Public License version 2.0

### Permissive and Copyleft Licenses

**Permissive** licenses are the ones that are **least restrictive**. Essentially, we can do whatever we want at no cost. There may be some simple requirement, such as including some header text that gives credit to the original author. One popular example of such an OSS License is the MIT License.

**Copyleft** Licenses, also called **viral** licenses or **protective** licenses, requires that the software we create using the open-source code must also be made open-source. This is based on the thought process that software should be free. GPL and AGPL are Copyleft licenses.

In general, permissive licenses tend to be much more popular than copyleft licenses.

### Non-Traditional Licenses

OSS Licenses in general tend to be **short and simple**. They look a lot less like proper legal text.

MIT License:

“Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions…”

However, some licenses are written by programmers, who are famous for having a sense of humour. This leads to licenses that are even less formal.

*/\*  
 \* -------------------------------------------------------------------------  
 \* "THE BEER-WARE LICENSE" (Revision 42):  
 \* <phk@FreeBSD.ORG> wrote this file. As long as you retain this notice you  
 \* can do whatever you want with this stuff. If we meet some day, and you  
 \* think this stuff is worth it, you can buy me a beer in return.  
 \* Poul-Henning Kamp  
 \* -------------------------------------------------------------------------  
 \*/*

C++

## Compliance

**Compliance** refers to the fact that users of OSS must observe all copyright notices and satisfy all license obligations. Companies that use OSS in **commercial products** want to not only comply with the licenses, but to also ensure that their and their third-party suppliers’ **intellectual property** is protected from **unintended disclosure**.

Compliance practices mitigate risks through:

* Identification of the origin and license of used software
* Identification of license obligations
* Fulfillment of license obligations when product ships

Compliance consists of:

* Policies
* Processes
* Training
* Tools

These allow the organization to use OSS and contribute to open communities while also:

* Respecting copyrights,
* Complying with license obligations
* Protecting the organization's intellectual property and that of its customers and suppliers.

### Objectives

* Comply with **third party software supplier contractual obligations** in light of FOSS licensing obligations
* Facilitate **effective usage** of FOSS in commercial products
* **Protect commercial product differentiation** while complying with FOSS contractual obligations

### Benefits

* Increased understanding of the **benefits** of FOSS and how it impacts your organization
* Increased understanding of the **costs and risks** associated with using FOSS
* **Better relations** with the FOSS community and FOSS organizations
* **Increased knowledge** of available FOSS solutions
* **Be prepared** for possible acquisition, sale, new product or service release, where compliance assurance is mandatory before the completion of any of these transaction
* Improve your overall **FOSS strategy** using the results from your compliance program

## Consequences of Violations

* Company to publish licensing notice on their website
* Company to provide additional notices in product publications
* Company to make available the complete and corresponding source code used in their product freely available on its website
* Company to cease binary distribution of the FOSS in question until it has published complete corresponding source code on its web site
* Company to pay an undisclosed amount of financial consideration to the plaintiffs
* Company to make available the complete and corresponding source code used in their product, releasing code that contains their product differentiation as open source under the GPL.
* In almost all cases, the failure to comply with the FOSS license obligations has also resulted in:
  + Public embarrassment
  + Negative press
  + Damaged relationships with some of their customers, suppliers and most notably the FOSS community

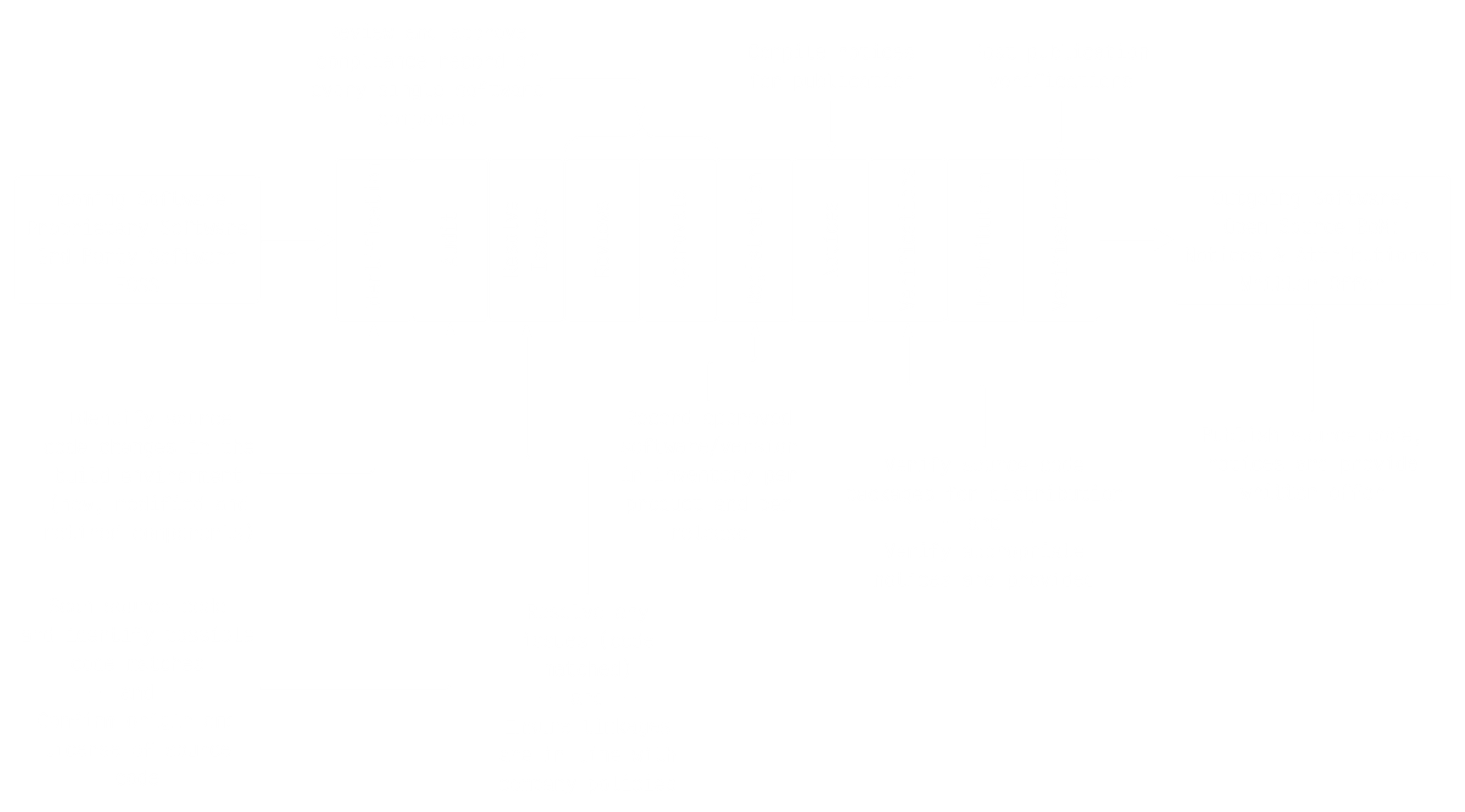
To avoid such consequences, not only must companies make compliance a **priority**, but they must also establish and maintain **consistent compliance policies** and procedures for third-party software they use.

This can be ensured by using an **end-to-end FOSS management infrastructure**, which ensures:

* Identify all FOSS it is using in its products
* Collect the applicable FOSS licenses for review by the legal department
* Develop FOSS use and distribution policies
* Institutionalize FOSS and compliance training to ensure that all employees are aware of the legal risks involved with using FOSS and aware of company policies
* Ensure that your software vendors, suppliers and subcontractors are adhering to FOSS license requirements
* Furthermore, companies need to know not only which FOSS they are using, but also how they are using them.

## Compliance Management

When we get incoming software, it may include proprietary software, third-party software and FOSS. We need to go through several steps to ensure the outgoing software does not break any of the above rules.



The steps above are:

1. Identification of FOSS
2. Auditing source code
3. Resolving any issues uncovered by the audit
4. Completing reviews
5. Receiving approval to use FOSS
6. Updating software inventory
7. Updating end user documentation
8. Performing verification of all previous steps prior to distribution
9. Distributing FOSS including any modifications (if any) when Applicable
10. Performing final verifications in relation to distribution

They work together to ensure that:

* FOSS used in the product has been identified, reviewed and approved
* The product implementation includes only the approved FOSS
* FOSS used in the product have been registered in the FOSS inventory system
* All obligations related to the use of licensed material have been identified
* Appropriate notices have been provided in the product documentation: these include a written offer to provide source code, attributions and copyright notices
* Source code including modifications (when applicable) have been prepared and ready to be made available once the product ships
* Verifications of all the steps in the process

### Identification

Prerequisites:

* A discovery of a FOSS being used via a platform scan
* A discovery of a FOSS being used as part of third party software

Outcome:

* A compliance record is created (or updated) for the FOSS
* An audit is requested to scan the source code

### Audit

This step involves scanning source code using an automated source code analysis tool to discover source code that matches FOSS source code.

The goals are:

* Identify the bill of material (BoM) of the component in question
* Confirm the origin(s) of the source code
* Flag dependencies, code matches and licensing conflicts
* Understand the licenses that govern its use, modification and distribution
* Identify the obligations of the various licenses

### Resolve Issues

Prerequisites:

* A source code scan has been completed
* An audit report is generated identifying the origins and licenses of the source code and flagging source code files that were not identified and that need further investigation.

Outcomes:

* A resolution for each of the flagged files in the report and a resolution for any flagged license conflict

### Review

Prerequisites:

* Source code has been audited
* All identified issues have been resolved

Outcome:

* Open-Source Review Board (OSRB) members perform an architecture review and a linkage analysis for the specific component and mark it as ready for the next step (i.e. Approval) if no issues were uncovered

### Approvals

Approval comes from the OSRB.

### Registration

The compliance ticket is updated to reflect the approval. It is added to the software inventory that tracks FOSS that is used in products.

### Notices

Companies that use FOSS in commercial products must:

* Acknowledge the use of FOSS by providing full copyright and attribution notices
* Inform the end user of their product on how to obtain a copy of the FOSS source code (when applicable, for example in the case of GPL and LGPL)
* Reproduce the entire text of the license agreements for the FOSS code included in the product.

### Pre-Distribution Verification

* FOSS packages destined for distribution have been identified and approved
* The source code packages (including modifications) have been verified to match the binary equivalence shipping in the product
* All appropriate notices have been included in the product documentation to inform end-users of their right to request source code for identified FOSS

### Distribution

The FOSS package is uploaded to the distribution website, identified with labels for the product and version it corresponds to.

### Final Verification

Prerequisites:

* The source code is published on the web site

Outcomes:

* Verify that the source code is:
  + Uploaded correctly
  + Corresponds to the same version that was approved
  + Accessible for download for the public

## General Guidelines

* Request formal approval for each open-source software you are using in product or in SDK (refer to your company’s Usage Policy)
* Save the web site from which you downloaded the open-source package and save a mint copy of the package you downloaded
* Consult with your manager when you upgrade your open-source software version. License changes can occur between versions.
* Don’t change or eliminate existing comments in headers
* Do not re-name open-source modules
* Do not discuss coding or compliance practices with persons outside the company
* Do not copy/paste FOSS code into proprietary or third-party source code or vice versa without OSRB approval.
* Mixing of different FOSS licenses in a derivative work must be avoided.
* When in doubt, always refer to the FSF resource page on license compatibility available [here](http://www.fsf.org/licensing/licenses/index.html).
* Do not remove or in any way disturb existing FOSS licensing copyrights or other licensing information from any FOSS components that you use.
* All copyright and licensing information is to remain intact in all FOSS components.