Giving Everyone Access To Open Source Best Practices

OpenChain Project - The Linux Foundation
Available under the CC Attribution-NoDerivatives 4.0 International license.
The value of open source is efficiency:

1. Efficient code creation
2. Efficient code support
3. Efficient code evolution
Open source license compliance is an opportunity for efficiency.
Open source license compliance reduces:

1. Legal and reputation risk
2. Costs of custom process management
3. Costs related to compliance violations
Many companies do a great job of reducing this inefficiency internally.
However, most companies rely on a supply chain.
We can significantly improve efficiency around open source license compliance in the supply chain.
OpenChain Project addresses the question of “how do I trust open source compliance in my supply chain?”
There are three parts to OpenChain Project:

1. Specification
2. Conformance
3. Curriculum
The OpenChain Specification defines a core set of requirements every quality compliance program must satisfy.

Learn more here: https://www.openchainproject.org/spec
OpenChain Conformance allows organizations to display their adherence to these requirements.

Learn more here: https://www.openchainproject.org/conformance
The OpenChain Curriculum provides the educational foundation for open source processes and solutions.

Learn more here: https://www.openchainproject.org/curriculum
Digging Deeper
The core of the OpenChain Curriculum is a reference deck to help people meet the requirements of the OpenChain Specification.
The deck can also be used for general training.
The deck is *not* intended to cover everything.
It is intended to be a useful starting point.
Contents

1. What is Intellectual Property?
2. Introduction to FOSS Licenses
3. Introduction to FOSS Compliance
4. Key Software Concepts for FOSS Review
5. Running a FOSS Review
6. End to End Compliance Management (Example Process)
7. Avoiding Compliance Pitfalls
8. Developer Guidelines
What is “Intellectual Property”?  

- **Copyright**: protects original works of authorship  
  - Protects expression (not the underlying idea)  
  - It covers software, books, and similar works  
- **Patents**: useful inventions that are novel and non-obvious  
  - Limited monopoly to incentivize innovation  
- **Trade secrets**: protects valuable confidential information  
- **Trademarks**: protects marks (word, logos, slogans, color, etc.) that identify the source of the product  
  - Consumer and brand protection; avoid consumer confusion and brand dilution  

*This chapter will focus on copyright and patents, the areas most relevant to FOSS compliance.*
Check Your Understanding

• What type of material does copyright law protect?
• What copyright rights are most important for software?
• Can software be subject to a patent?
• What rights does a patent give to the patent owner?
• If you independently develop your own software, is it possible that you might need a copyright license from a third party for that software? A patent license?
Example Small to Medium Company Checklist

Ongoing Compliance Tasks:
1. Discover all FOSS early in the procurement/development cycle
2. Review and Approve all FOSS components used
3. Verify the information necessary to satisfy FOSS obligations
4. Review and approve any outbound contributions to FOSS projects

Support Requirements:
1. Ensure adequate compliance staffing and designate clear lines of responsibility
2. Adapt existing Business Processes to support the FOSS compliance program
3. Have training on the organization’s FOSS policy available to everyone
4. Track progress of all FOSS compliance activities

You can get detailed checklists for these items here: https://www.linuxfoundation.org/projects/opencompliance/self-assessment-compliance-checklist
Example
Enterprise Process

Identify FOSS components for review
Scan or audit source code – and – Confirm origin and license of source code
Resolve any audit issues in line with company FOSS policies
Record approved software/version in inventory per product and per release
Verify source code packages for distribution – and – Verify appropriate notices are provided
Publish source code, notices and provide written offer

Review and approve compliance record of FOSS software components
Compile notices for publication
Post publication verifications

Outgoing Software
Notices & Attributions
Written Offer

Example of Compliance Management End-to-End Process
CHAPTER 7

Avoiding Compliance Pitfalls
### Intellectual Property Pitfalls

<table>
<thead>
<tr>
<th>Type &amp; Description</th>
<th>Discovery</th>
<th>Avoidance</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Unplanned inclusion of copyleft FOSS into proprietary or 3rd party code:</strong></td>
<td>This type of failure can be discovered by scanning or auditing the source code for possible matches with:</td>
<td>This type of failure can be avoided by:</td>
</tr>
<tr>
<td></td>
<td>• FOSS source code</td>
<td>• Offering training to engineering staff about compliance issues,</td>
</tr>
<tr>
<td></td>
<td>• Copyright notices</td>
<td>the different types of FOSS licenses and the implications of</td>
</tr>
<tr>
<td></td>
<td>Automated source code scanning tools may be used for this purpose</td>
<td>including FOSS in proprietary source code</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Conducting regular source code scans or audits for all the source</td>
</tr>
<tr>
<td></td>
<td></td>
<td>code in the build environment.</td>
</tr>
</tbody>
</table>

This type of failure occurs during the development process when engineers add FOSS code into source code that is intended to be proprietary in conflict with the FOSS policy.
Check Your Understanding

• What types of pitfalls can occur in FOSS compliance?
• Give an example of an intellectual property failure.
• Give an example of a license compliance failure.
• Give an example of a compliance process failure.
• What are the benefits of prioritizing compliance?
• What are the benefits of maintaining a good community relationship?
CHAPTER 8
Developer Guidelines
Developer Guidelines

• Select code from high quality, well supported FOSS communities
• Seek guidance
  • Request formal approval for each FOSS component you are using
  • Do not check un-reviewed code into any internal source tree
  • Request formal approval for outside contributions to FOSS projects
• Preserve existing licensing information
  • 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
  • Do not re-name FOSS components unless you are required to under the FOSS license (e.g., required renaming of modified versions)
• Gather and retain FOSS project information required for your FOSS review process
Check Your Understanding

• Name some general guidelines developers can practice when working with FOSS.
• Should you remove or alter FOSS license header information?
• Name some important steps in a compliance process.
• How can a new version of a previously-reviewed FOSS component create new compliance issues?
• What risks should you address with in-bound software?

Learn more through the free Compliance Basics for Developers hosted by the Linux Foundation at:
https://training.linuxfoundation.org/linux-courses/open-source-compliance-courses/compliance-basics-for-developers
We also have a GitHub repository with more material:
https://github.com/OpenChain-Project/curriculum
Everything is Creative Commons 0 (public domain).
This means the knowledge can be used, shared, remixed and incorporated anywhere in any way.
We are always looking for donations of material or time to improve our curriculum.
Let’s Talk

Shane Coughlan
coughlan@linux.com
+818040358083

www.openchainproject.org