License Information Management: Zephyr Case Study

Kate Stewart & Steve Winslow
Linux Foundation
Opening Up Your Source Code

So you picked a license...
So you picked a license...

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION

1. Definitions. "License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of ...

...now what?

“Confetti Girl” image by Scout; used under CC0-1.0; https://openclipart.org/detail/232158/confetti-girl
Opening Up Your Source Code

What licenses are already inside your source code?
Opening Up Your Source Code

What licenses are already inside your source code?

(potentially more than you expected)
Opening Up Your Source Code

An existing code base might contain:

- your own code

```python
def _getFinalConfigValue(self, kwValue):
    if kwValue is not None:
        return str(kwValue)
    try:
        value = self.db.get_value()
        return str(value)
    ```
Opening Up Your Source Code

An existing code base might contain:
- your own code
- third-party proprietary code
Opening Up Your Source Code

An existing code base might contain:
- your own code
- third-party proprietary code
- incompatible open source licenses
Opening Up Your Source Code

An existing code base might contain:

- your own code
- third-party proprietary code
- incompatible open source licenses
- missing open source licenses

Pencil image by TheUjulala; used under CC0-1.0
Opening Up Your Source Code

An existing code base might contain:

- your own code
- third-party proprietary code
- incompatible open source licenses
- missing open source licenses
- puzzling license statements
Opening Up Your Source Code

An existing code base might contain:

- your own code
- third-party proprietary code
- incompatible open source licenses
- puzzling license statements
  
  “See LICENSE in LICENSE”
  (with no LICENSE file in repo)

  “Licensed under the Creative Commons Attribution 4.0 International License, titled CC-BY-SA-4.0”
Opening Up Your Source Code

An existing code base might contain:

- your own code
- third-party proprietary code
- incompatible open source licenses
- missing open source licenses
- puzzling license statements
- your own confidentiality notices
Opening Up Your Source Code

An existing code base might contain:

- your own code
- third-party proprietary code
- incompatible open source licenses
- missing open source licenses
- puzzling license statements
- your own confidentiality notices
- code with snarky licenses
Opening Up Your Source Code

An existing code base might contain:

• your own code
• third-party proprietary code
• incompatible open source licenses
• your own confidentiality notices
• code with snarky licenses

“This is free software; you can redistribute it and/or modify it under the terms of the BSD License. Use by owners of Che Guevarra parafernalia is prohibited, where possible, and highly discouraged elsewhere.”
Opening Up Your Source Code

An existing code base might contain:
- your own code
- third-party proprietary code
- incompatible open source licenses
- missing open source licenses
- puzzling license statements
- your own confidentiality notices
- code with snarky licenses
- code with secret keys or passwords

“Cles de serrure – lock keys” image by enolynn; used under CC0-1.0 https://openclipart.org/detail/190821/cles-de-serrure-lock-keys
Opening Up Your Source Code

An existing code base might contain:

- your own code
- third-party proprietary code
- incompatible open source licenses
- missing open source licenses
- puzzling license statements
- your own confidentiality notices
- code with snarky licenses
- code with secret keys or passwords
- code with security vulnerabilities

Heartbleed logo image by Synopsys, Inc.; used under CC0 1.0
http://heartbleed.com/
Opening Up Your Source Code

An existing code base might contain:

- your own code
- third-party proprietary code
- incompatible open source licenses
- missing open source licenses
- puzzling license statements
- your own confidentiality notices
- code with snarky licenses
- code with secret keys or passwords
- code with security vulnerabilities
Opening Up Your Source Code

An existing code base might contain:

- your own code
- third-party proprietary code
- incompatible open source licenses
- missing open source licenses
- puzzling license statements
- your own confidentiality notices
- code with snarky licenses
- code with secret keys or passwords
- code with security vulnerabilities
- dependencies with any of the above
Opening Up Your Source Code

An existing code base might contain:

- your own code
- third-party proprietary code
- incompatible open source licenses
- missing open source licenses
- puzzling license statements
- your own confidentiality notices
- code with snarky licenses
- code with secret keys or passwords
- code with security vulnerabilities
- (sub)dependencies with any of the above
Tempting Response: Ignore it

Drop in a LICENSE.txt file and declare yourself done
License Management

General process:

• Identify licenses
• Address incompatibilities
• Address compliance
• Communicate licenses
License Management

General process:

- **Identify licenses**
- Address incompatibilities
- Address compliance
- Communicate licenses

Focusing on these two in this talk

(getting these right enables meaningful conversations about the other two)
License information can be managed!

This is not an insurmountable challenge

Tackling it benefits projects and benefits the whole ecosystem (and not just by making lawyers happier!)

Avoid making “perfect” the enemy of “better”

There are gaps in today’s tooling but there is also forward progress
Identifying Licenses

Goal 1: Determine which licenses are relevant to your project

Goal 2: Do so in an automated, scalable way
Identifying Licenses

Different types of scans:
- license scanning
- code scanning
- dependency scanning
Scanning Tools

Quick and dirty; no tooling needed

Look for relevant words / fragments:

- “licen”
- “redist”
- “copyright”
- common license fragments: “bsd”, “gpl”, “general public”, “cddl”, ...

Manual searches

grep -nri
(or your favorite command line args)

Ctrl-F
(or your favorite editor’s equivalent)
FOSSology is used to scan a codebase for licenses

Performs textual analysis and regular expression scanning to identify likely license notices and references

Supplemented with manual review to remove false positives and investigate unusual findings
Scanning Tools

Version 3.3 released in May 2018

Since 3.2 it includes:

- SPDX file imports
- Obligation analysis and summaries

https://www.fossology.org/
https://github.com/fossology/fossology
Scanning Tools

From ScanCode’s README:
ScanCode is a suite of utilities used to scan a codebase for license, copyright, package manifests and dependencies and other interesting information that can be discovered in source and binary code files.

https://github.com/nexB/scancode-toolkit
Scanning tools

Various other scanning tools and services, including open source and proprietary / commercial options

Some include security vulnerability detection

Some include initial free tiers for open source projects (read carefully how they define “open source” and “projects”)

Scanning tools

Keep in mind:

However automated the tooling is, some manual review will likely be required
Goal 1: Let others know what licenses are relevant to your project

Goal 2: Do so in an automated, scalable way
Communicating License Information

From the specification:

- The Software Package Data Exchange (SPDX®) specification is a standard format for communicating the components, licenses, and copyrights associated with software packages.

Current version:
https://spdx.github.io/spdx-spec/

Prior Versions:
https://spdx.org/specifications
Communicating License Information

SPDX Documents comprise manifests of files from software packages

Includes checksum hashes per file, license information and other optional data

Two official formats:
- **XML** – easier for automated consumption
- **Tag-value** - easier for human consumption

Translation tools can convert to spreadsheets, JSON, YAML, XML etc., and next revision of spec (2.2) will make them official

https://spdx.github.io/spdx-spec/
Communicating License Information

SPDX Documents comprise manifests of files from software packages. Includes checksum hashes per file, license information and other optional data.

Two official formats:
- **XML** – easier for automated consumption
- **Tag-value** - easier for human consumption

Translation tools can convert to spreadsheets, JSON, YAML, XML etc., and next revision of spec (2.2) will make them official.

https://spdx.github.io/spdx-spec/
Communicating License Information

From the License List:

“...a list of commonly found licenses and exceptions used in free and open source and other collaborative software or documentation.”

“The purpose of the SPDX License List is to enable easy and efficient identification of such licenses and exceptions in an SPDX document, in source files or elsewhere.”

SPDX License List

https://spdx.org/licenses
Communicating License Information

From the License List:

“...a list of commonly found licenses and exceptions used in free and open source and other collaborative software or documentation.”

“The purpose of the SPDX License List is to enable easy and efficient identification of such licenses and exceptions in an SPDX document, in source files or elsewhere.”

Examples:
- BSD-2-Clause
- BSD-3-Clause
- GPL-2.0-only
- GPL-3.0-or-later
- MIT
- MPL-2.0

https://spdx.org/licenses
Communicating License Information

One-line comment in each source code file to unambiguously designate the applicable license(s)

Examples:

/* SPDX-License-Identifier: GPL-2.0-only */

// SPDX-License-Identifier: BSD-2-Clause OR MIT

# SPDX-License-Identifier: Apache-2.0 AND MIT

SPDX Short-Form IDs

Usage example:
Communicating License Information

One-line comments can unambiguously designate applicable license(s).

Examples:

/* SPDX-License-Identifier: GPL-2.0-only */

// SPDX-License-Identifier: BSD-2-Clause OR MIT

# SPDX-License-Identifier: Apache-2.0 AND MIT

If a file’s license ID looks like this, maybe rethink that file’s structure:

GPL-3.0 AND GPL-2.0+ AND GPL-2.0 AND LGPL-2.1+ AND LGPL-2.1 AND MIT AND BSD-3-Clause AND (AFL-2.1+ OR BSD-3-Clause) AND (MIT OR LicenseRef-BSD OR LicenseRef-GPL) AND (MIT OR LicenseRef-GPL) AND (MPL-1.1 OR GPL-2.0 OR LGPL-2.1) AND LicenseRef-MIT-style

Usage example:
Communicating License Information

The REUSE Initiative (from Free Software Foundation Europe) provides **best practices** in communicating license information for an entire package, and **tools** to assist in confirming compliance with those practices.

Includes recommendations for how and where to place copyright notices, license references and license texts

 Makes use of SPDX short-form identifiers

https://reuse.software

The REUSE website and logo are copyright © FSFE e.V. The REUSE logo is licensed under Creative Commons Attribution-ShareAlike 4.0.
Contribution instructions for your project:

• Include a file (CONTRIBUTIONS.md) which explains that contributions are required to be made under the project’s license

• In that file, also include:
  – the Developer Certificate of Origin (https://developercertificate.org/)
  – a statement that “Signed-off-by:” lines in commit messages signal an affirmation to the DCO
Related Suggestions

Location for third-party software:

- Whenever possible, where third-party software is included within your repository, keep it in a separate “third-party/” or “ext/” or similar folder
  - May already be a standard or semi-standard, e.g. “vendor/” folder for many Golang projects; “node_modules/” for NPM projects

- Helps flag to downstream users that licenses may differ

- Also provides a good place to focus when looking for security vulnerabilities in dependencies
Current Status and Gaps

How well do all these pieces fit together?

...disparate tools; it’s a work in progress

Focus is now turning to developing centralizing tools to unify these different parts of the licensing story

• e.g. Quartermaster (http://qmstr.org/)
To Learn More…

Free publication available from The Linux Foundation website:

Now available in Chinese!

Free publication available from The Linux Foundation website:


DOWNLOAD THE PAPER (CHINESE)
Case Study: Zephyr

https://www.zephyrproject.org/

https://github.com/zephyrproject-rtos/zephyr
Case Study: Zephyr

The Zephyr project is Apache-2.0 licensed

The project leaders and developers have intentionally focused on improving management of the license information for their code
Case Study: Zephyr

Zephyr license processes:

• License review (in addition to code review) for all commits not fully under Apache-2.0
  – Currently a manual process
  – Would prefer to have checking IDs automatically
Case Study: Zephyr

Zephyr license processes:

- Each Zephyr source code file has a one-line SPDX-License-Identifier comment

/* SPDX-License-Identifier: Apache-2.0 */
Case Study: Zephyr

Zephyr license processes:

- Anything not under the project’s Apache-2.0 license is in a separate “ext/” directory
  - Might not have SPDX-License-Identifier for these files
  - Keeping third party files unmodified makes it easier to refresh updates
  - Process for contributing is documented, and expectation that a README will provide appropriate licensing information as part of initial commit before it is accepted. Expectation is it will reflect any updated licensing. https://github.com/zephyrproject-rtos/zephyr/blob/master/doc/contribute/contribute_non-apache.rst
Case Study: Zephyr

Zephyr license details:

• Apache-2.0 license text in LICENSE file

• Details about choice of license, processes and use of DCO in CONTRIBUTING.rst file
Case Study: Zephyr

Zephyr license details:

- Project page with clear details about non-Apache licenses in the codebase: [http://docs.zephyrproject.org/LICENSING.html](http://docs.zephyrproject.org/LICENSING.html)

- “SPDX-License-Identifiers” in all other files make it easy to auto-generate license details

- Will be generating .spdx file with first LTS release, and all releases after.