Licensing: How to open-source your work

Skylar Chan John Evans Jeffrey Fisher

Introduction

Introduction

This presentation is brought to you by the Linux Club at UMD.

What this presentation is

We intend that after this presentation you:

- Know what permissions popular licenses grant you
- Have a good idea of what licenses you may want to use for your projects
- (maybe) Have some understanding of the issues if you want to make money from open-source work

What this presentation is **not**

We will avoid promoting a particular choice. However, we may mention pros and cons of particular choices. Also, this presentation is focused on free and open-source licenses.

Disclaimer

We are not lawyers.

If you want to use a custom or less-understood license, you should probably consult a lawyer.

Terminology

Terminology

Copyright, works, licenses, oh my!

Work

noun something done or made.

Examples:

- a painting
- a book
- software source code
- CAD design files

Copyright Notice

Statement of who owns the copyright for something.

Example: “Copyright © 2012 Jane Doe”

- Include year of first publication.
- A year range may be included for works that are frequently updated: “Copyright © 2000-2004 Jane Doe”

License

An agreement that grants permissions, possibly with some restrictions.

Free and Open Source Software (FOSS)

The free software movement and the open-source software movement are separate but closely related.

Most licenses that fit the free software definition also fit the open-source software definition, and vice versa. ("Categories of Free and Nonfree Software," n.d.)

The commonalities between the two are access to source code and freedom to use, modify, and share copies of the code for any purpose.

Proprietary

For this presentation, “proprietary” means code that does not provide all of these freedoms.

Just stick it online, right?

No license = All rights reserved

When you make a creative work (which includes code), the work is under exclusive copyright by default.

Unless you include a license that specifies otherwise, nobody else can copy, distribute, or modify your work without being at risk of take-downs, shake-downs, or litigation.

Once the work has other contributors (each a copyright holder), “nobody” starts including you. (“Choose an Open Source License,” n.d.)

- If you are the only user, this is probably fine
- It's not required, but it's a good idea to put a copyright notice and statement of "all rights reserved" in an easy-to-find place like a file named README.
- If you want other people to reuse and share your work, this is probably not what you want
- Choose a license to specify your terms more clearly

Permissive licenses

I want it simple and permissive

Permissive licenses let people do almost anything they want with your project, like making and distributing closed source versions. (“Choose an Open Source License,” n.d.)

- Includes public domain (there is no copyright owner)

Examples

- MIT: NodeJS
- BSD: scikit-learn
- Apache: Qiskit
  - has additional terms on patent rights
- Unlicense: yt-dlp

Copyleft licenses

Copyleft licenses lets people do almost anything they want with your project, **except** distributing closed source versions. ("Choose an Open Source License," n.d.)

- Changes must be redistributed with a copyleft license and with source code
- Some copyleft licenses (GPL) may be incompatible with proprietary code
  - GPL extends to linking libraries in compiled code
  - LGPL projects can be linked to proprietary code

Examples

- GPL: Linux kernel (v2), LLaMA (v3)
  - v3 provides additional clarifications on license compatibility, digital rights management, and patents
- AGPL: Canvas LMS
  - requires servers hosting the source code to share any changes
- LGPL: QT (most of it)
  - most of it can be linked in proprietary projects without source redistribution

Non-software works

Creative Commons

Use Creative Commons to license creative, non-software works like pictures, video, and music.

- CC0 = public domain
- CC-BY = attribution (like MIT)
- CC-BY-SA = share alike (like GPL)

3D Printing

Home 3D printing has made sharing 3D models common practice.

Similar to source code files, 3D model files can be considered creative works, and can be licensed as such.

Note: This does not mean that the design or the physical prints are protected. Patents would be required for that.

Working with others and in a community

Making a new project

- Your license conveys to the community how you intend your project to be used and shared
- Permissive licenses are easier to use and understand
- Copyleft licenses encourage publicly sharing changes, and have broader legal protections (in theory)

How to apply a license

GitHub's https://choosealicense.com/ is a great resource, and heavily inspired this presentation.

For each license it tells you how to apply it and gives a summary of the permissions and conditions.

Add a license

- Create a file at the top-level folder of your project named `LICENSE` or `LICENSE.txt`, so it is easy to find.
- Copy the text of the license you are using into that file.

Copyright notice

- Add a copyright notice at the top of the license file.
  - Copyright (c) [year] [name of author]
- (maybe) In a README file and at the top of every code file, add a copyright notice and a statement of what license the file is licensed under.
  - This is not strictly necessary, but can reduce chances that someone copies the code and forgets to provide a copy of the license.
  - At the end of the GPL there is shorter sample text that you can choose to include in every file.

How to check the license of other code

- Look for comments at the top of the file that mention the license.
- Look for common filenames like `LICENSE`, `LICENSE.txt`, `COPYING`, `README`
- Sites like GitHub and GitLab may detect the license and display it to you in the interface.

If you can't find anything, contact the author to ask.

Make sure your plans on how to use the code fit the license. If it is a dependency of your project, make sure your project is licensed under a compatible license.

Dependencies and using other people's code

- License compatibility: you must comply with the licenses of code you depend on
- Rule of thumb:
  - copyleft code can only be reused in copyleft code
  - permissive code can be reused almost anywhere
  - proprietary code may have restrictions (frequently noncommercial)

Examples for how to check for dependencies

You can check the licenses of your code's dependencies for various languages:

- Rust: cargo-about
- Python: pip-licenses
- JavaScript: license-checker

Contributing to an existing project

- Review the license
- Read the contributor agreements, if there are any. You may be required to assign your copyright to the copyright holder for them to use your contributions and enforce the license.

Example contributor agreements:

- To contribute to GNU Emacs, you must assign your copyright to the Free Software Foundation.
- Developer Certificate of Origin (DCO): Used by the Linux kernel. In brief, the certificate verifies that the developer wrote the open-source changes, and allows the project to use them in a way consistent with the license.

Contributor License Agreement (CLA)

Typically used in corporate projects. Different CLAs may have different terms. Take note of terms that allow the owner of the project to relicense your patches, as they can relicense open-source code *you* own into proprietary code *they* own:

When a CLA requires a contributor to assign unrestricted republishing rights to the project, contributed code can be relicensed at the discretion of the project, even when the CLA does not assign copyright to the project. (“Contributor License Agreement: Relicensing Controversy,” n.d.)

Non-FOSS licenses

Non-FOSS licenses

They may provide some of the freedoms and use terminology associated with FOSS, but do not fit all the requirements.

An important requirement of FOSS is that it can be used for any purpose (commercial or otherwise).

- Just because you can look at the code doesn't mean it's FOSS
- Leaked/reverse-assembled proprietary code is still proprietary

Source-available licenses

- Business Source License (“permissive”)
  - Releases are source available with a promise to open-source it later
  - MariaDB, Codon, CockroachDB
- Server Side Public License (“copyleft”)
  - Copyleft extends much farther than the AGPL
  - MongoDB

Noncommercial Creative Commons licenses

In recent years, the Creative Commons family of licenses have become popular.

3D printing repository websites such as Thingiverse and Printables have these licenses available for users to select from when uploading their designs.

While it is not well explained on these sites, the Non-Commercial versions of these licenses are not free or open source. (“Creative Commons 4.0 BY and BY-SA Licenses Approved Conformant with the Open Definition,” n.d.)

This is because Non-Commercial licenses limit usage to Non-Commercial usage only.

Ethical Source license

Ethical Source[1] licenses[2] have provisions that restrict uses for ethical purposes.

- JSON license: "The Software shall be used for Good, not Evil."
  - Considered non-free by multiple organizations because of this restriction ("JSON Non Free License," n.d.)
  - How do you define "good" or "evil"? What counts as "ethical"?
  - Maybe unenforceable

[1] https://ethicalsource.dev/
[2] https://ethicalsource.dev/licenses/

Practical and economic considerations

- Making money from FOSS is difficult but possible
- One option is to sell commercial licenses and software support

Further reading: Uncurled

Legal considerations

Consult a lawyer if:

- You want a custom license
- You want to litigate a license violation

Note: we (the authors) are not lawyers.

Further reading: Open Source Software Litigation and Risks

The end

Now you know the basics of software licensing!

Bibliography

Bibliography

"Categories of Free and Nonfree Software." n.d. GNU Project. https://www.gnu.org/philosophy/categories.html.

"Choose an Open Source License." n.d. GitHub, Inc. https://choosealicense.com/.

"Contributor License Agreement: Relicensing Controversy." n.d. Wikimedia Foundation, Inc. https://en.wikipedia.org/wiki/Contributor_License_Agreement?lang=en#Relicensing_controversy.

"Creative Commons 4.0 BY and BY-SA Licenses Approved Conformant with the Open Definition." n.d. Creative Commons. https://creativecommons.org/2013/12/27/creative-commons-4-0-by-and-by-sa-licenses-approved-conformant-with-the-open-definition/.

"JSON Non Free License." n.d. Debian QA. https://wiki.debian.org/qa.debian.org/jsonevil.