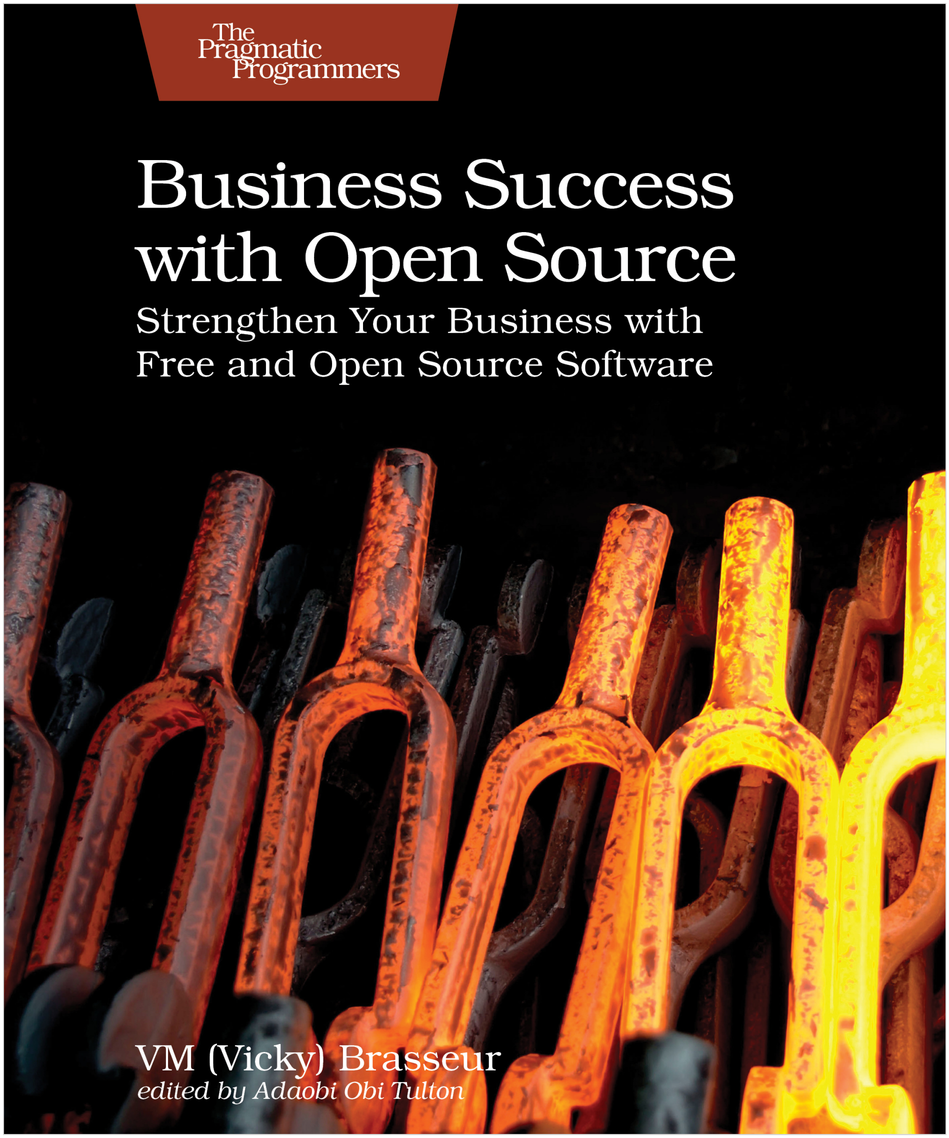

The Pragmatic Programmers

Business Success with Open Source

Strengthen Your Business with Free and Open Source Software

VM (Vicky) Brasseur

edited by Adaobi Obi Tulton

Basic License Compliance

Yes, obligations vary across the FOSS licensing landscape. That doesn't mean license compliance needs to be a convoluted mess. In fact, three basic steps will cover the majority of your license compliance needs (and one of them is optional for a subset of licenses): checking for compatibility, giving attribution, and providing source code.

Check for Compatibility

FOSS licenses generally mingle pleasantly together in a software codebase, but a few don't get on well at all. For instance, Apache-2.0 and GPL-2.0 don't play nicely together,[6] which can be problematic since these two FOSS licenses are popular. When reviewing the licenses represented in your SSC, make sure they're compatible with each other. The license compatibility matrix in FOSS License Compatibility, on page ?, will help here, but you may also need legal assistance for anything more than the most basic case.

Should you locate potential incompatibilities, see how the relevant components are used in the software. They may not be used in a way that triggers the licenses' terms that lead to incompatibilities. If those terms *are* triggered, there's nothing for it but to replace one of the components, thus eliminating the incompatibility. This, as you can imagine, can be a pain. As you'll learn later in this chapter, automated policies can help to avoid this particular problem.

Give Attribution

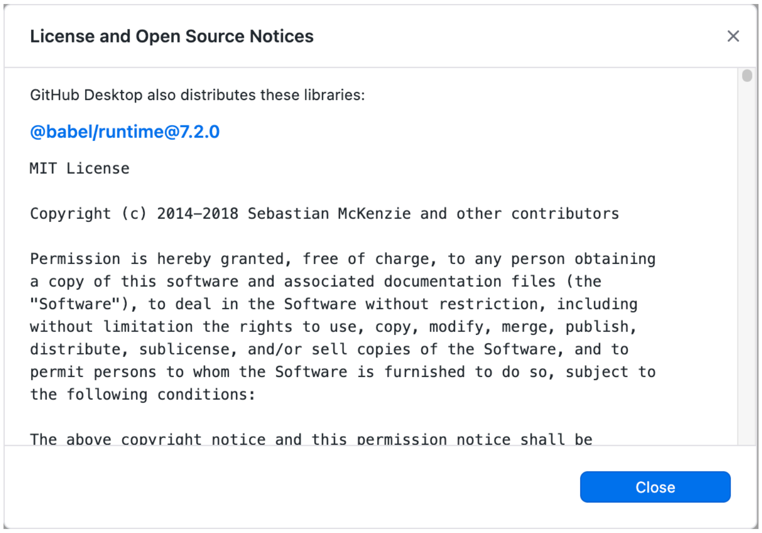
When you benefit from the work of someone else, it's polite to give them credit (attribution) for their contributions. In FOSS, it's not only polite but also required by the terms of most licenses, both permissive and reciprocal. For instance, the highly permissive MIT license has but one license term, and it includes attribution for the copyright holder:

> The above copyright notice and this permission notice (including the next paragraph) shall be included in all copies or substantial portions of the Software.

We usually call these attributions *notices* or *NOTICE files*, but you'll find them under a variety of names and also in a variety of formats. These notices should include the information required by the license (usually, as shown in the preceding MIT example, the copyright notice and the text of the license) and must be posted somewhere users of the software can access it.

6. https://www.apache.org/licenses/GPL-compatibility.html

For example, as shown in the following figures, the open source GitHub Desktop Git version control client includes the notices in a link that someone can reach via the About dialog box for the software, while Microsoft's open source VS Code editor lists the notices in a file named ThirdPartyNotices.txt on its public version control source code repository.[7]



GitHub Desktop also distributes these libraries:

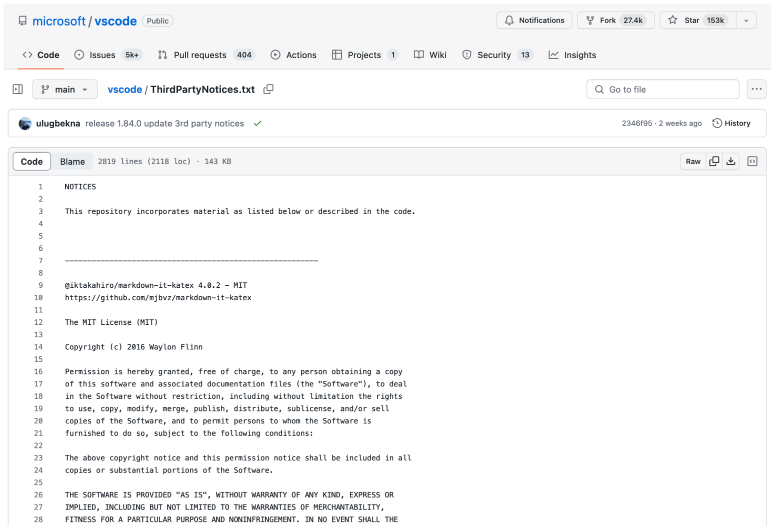
@babel/runtime@7.2.0

```
MIT License

Copyright (c) 2014-2018 Sebastian McKenzie and other contributors

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:

The above copyright notice and this permission notice shall be
```



```
NOTICES

This repository incorporates material as listed below or described in the code.



---------------------------------------------------------

@iktakahiro/markdown-it-katex 4.0.2 - MIT
https://github.com/mjbvz/markdown-it-katex

The MIT License (MIT)

Copyright (c) 2016 Waylon Flinn

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:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
```

7. https://github.com/microsoft/vscode/blob/main/ThirdPartyNotices.txt

Most SCA software can provide this information, and some (such as Open Source Review Toolkit (ORT)) can generate the notice files for you.

In Software Bill of Materials (SBOM), on page ?, you learned about SBOMs. If your company either generates or receives SBOMs, then you're in luck! A correctly formatted SBOM will include all of the information required to generate notice files, making the process considerably easier.

Provide Source Code

If the software your company creates includes reciprocally licensed components, you're required to provide the source code for the derivative works created using those components. This is the "reciprocal" nature of these FOSS licenses: you've benefitted from this FOSS project, so others should be able to benefit from your own work that's based on it.

What constitutes a "derivative work" is a complex matter, as you learned in Derivatives, on page ?. I feel it's safest to interpret it as broadly as reasonable rather than attempting to walk the thin line of "derivation or not." As in so many cases where licenses are concerned, you may need to bring in the lawyers to sort it all out.

Regardless, once someone has decided that the software is a derivative of a reciprocally licensed FOSS component, you'll need to ensure that people can access the source code for that derivative work. Out of any number of ways to do this, the easiest (especially in the case of software embedded in hardware) may be to provide a link to the source code as well as the instructions and support utilities necessary for building or compiling the code to create a functional version of the derivative work.

Ignorance Is No Excuse

Regardless of what you read here or elsewhere, *always read and understand the licenses in your software supply chain*. Remember: by using the software you're agreeing to the terms of those licenses. Would your company sign a contract without reading it first? Probably not, but that's similar to what happens when people use FOSS components without being familiar with the terms in their licenses.

Always have someone—a lawyer or knowledgeable layperson—read and understand the licenses in your SSC. Also, make sure they verify that those licenses are actually open source—that is, reviewed and approved by the Open Source Initiative. A lot of licenses out there claim to be open source but actually include terms that violate the standard Open Source Definition.

These *fauxpen source* licenses can cause unexpected and unpleasant surprises for your company, its business, and its customers.

The Container Complication Redux

You learned in The Container Complication, on page ?, containers hold all the software needed to run a program, they'll run just about anywhere, and they're a big pain in the butt when you're trying to get insight into your software supply chain. Therefore it shouldn't surprise you to learn that this also makes containers a big pain in the butt where license compliance is concerned.

Many of the containers used in software development are created externally and brought into your company by way of a *container registry*, which is a sort of search and storage engine for container images. Because the container is created elsewhere, it means that someone somewhere has distributed that container. You know what distribution does, right? Right: it triggers FOSS license terms. Many container creators and distributors aren't aware of this (but, again, ignorance is no excuse), so by the time the container reaches your software development process, it may already be out of compliance with the licenses of the software it's distributing.

Now, if you're only going to be using the container for software that's used internally at your company, you probably don't have much to worry about here. After all, you weren't the one who distributed the container and its software, so you're not the one out of compliance with the licenses. But if there's any chance at all that the software will be available outside of your company? Well, you got trouble, my friend. With a capital T, that rhymes with C, and that stands for Containers.

It's OK, this is fixable. The Tern open source tool[a] allows you to have a look inside of a container and can create an SBOM of the contents. If your inspection of that SBOM reveals any potential FOSS licensing problems, you can either ask the container creator to come into compliance or re-create the container internally in a way that complies with all relevant FOSS licenses.

a. https://github.com/tern-tools/tern