Extracted from:

Business Success with Open Source

Strengthen Your Business with Free and Open Source Software

The Pragmatic Bookshelf

Dallas, Texas

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

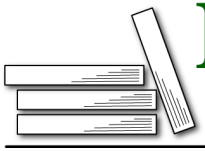
Business Success with Open Source

Strengthen Your Business with Free and Open Source Software

VM (Vicky) Brasseur

The Pragmatic Bookshelf

Dallas, Texas



For our complete catalog of hands-on, practical, and Pragmatic content for software developers, please visit *https://pragprog.com*.

For sales, volume licensing, and support, please contact *support@pragprog.com*.

For international rights, please contact *rights@pragprog.com*.

ISBN-13: 979-8-88865-049-3
Encoded using the finest acid-free high-entropy binary digits.
Book version: B1.0—October 12, 2023

Basic License Compliance

Yes, terms and conditions vary across the FOSS licensing landscape. That doesn't mean license compliance needs to be a convoluted mess. In fact, there are three basic steps that 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.

Checking for Compatibility

FOSS licenses generally mingle pleasantly together in a software codebase, but there are a few that don't get on well at all. For instance, Apache-2.0 and GPL-2.0 do not play nicely together, which can be problematic as these two FOSS licenses are quite popular. When reviewing the licenses represented in your SSC, make sure they're compatible with each other. The license compatibility matrix in Chapter 3, Licenses: The Rules of IP Engagement, 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 bit of a pain. As you'll learn later in this chapter, automated policies can help to avoid this particular problem.

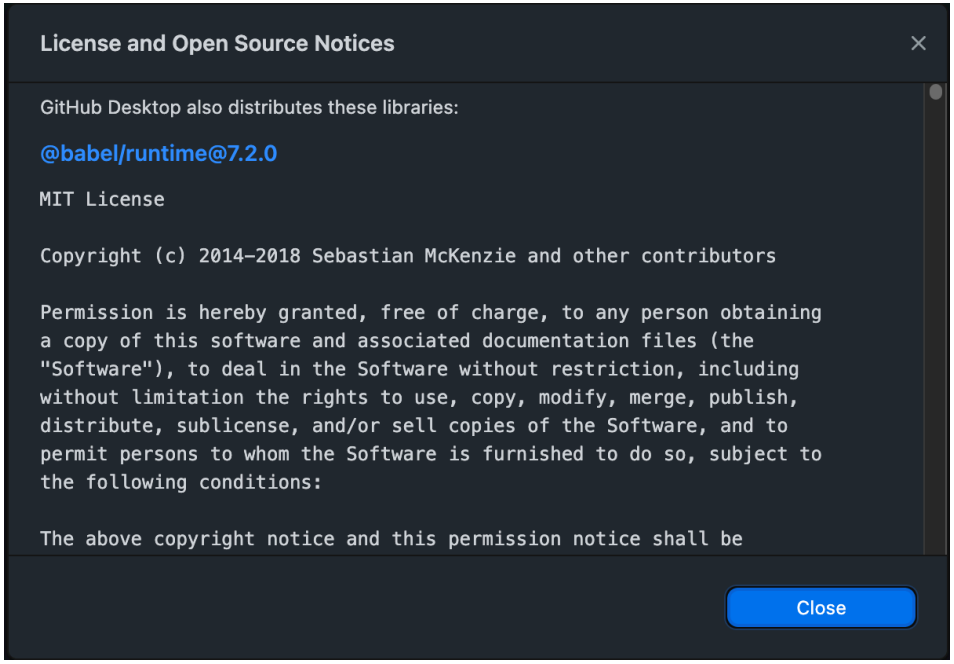
Giving Attribution

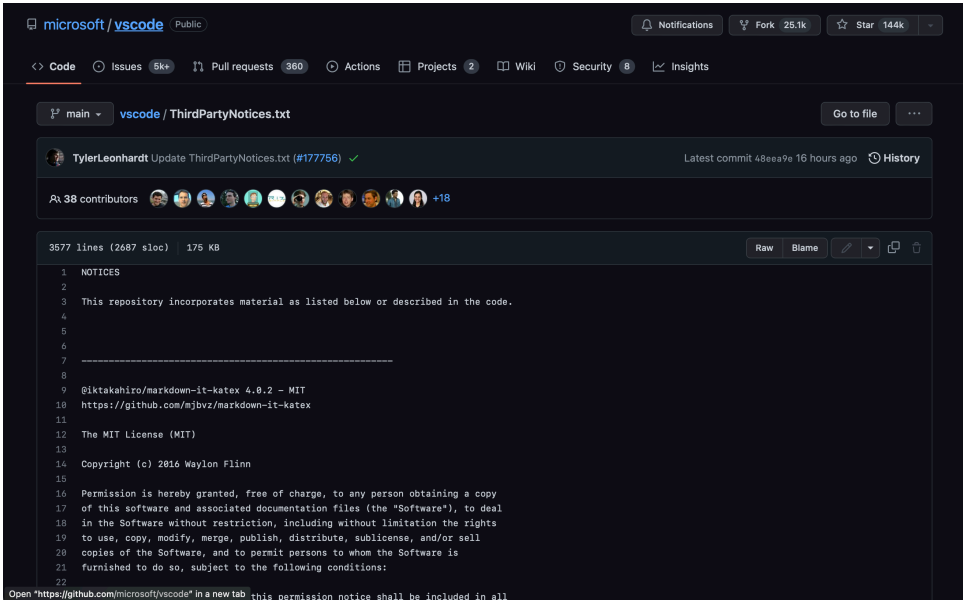
When you benefit from the work of someone else, it's polite to give them credit (attribution) for their contributions. In FOSS, it's not only polite but it's also required by the terms of most licenses, both permissive and reciprocal. For instance, the highly permissive MIT license has but one license 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 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 MIT example above, the copyright notice and the text of the license) and must be posted somewhere users of the software can access it. For example, the open source GitHub Desktop git version control client includes the notices in a link that the user 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.





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

In Chapter 11, Know the Links in Your Software Supply Chain, 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.

Providing Source Code

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

What constitutes a "derivative work" is a complex matter, as you learned in Chapter 3, Licenses: The Rules of IP Engagement, 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 users can access the source code for that derivative work. There are any number of ways to do this, but 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 are 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 knowledgable layperson—read and understand the licenses in your SSC. Also, make sure they verify that those licenses are actually open source, i.e. are reviewed and approved by the Open Source Initiative. There are a lot of licenses out there that 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, as well.

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 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 recreate the container internally in a way that complies with all relevant FOSS licenses.