EDITOR DIRK RIEHLE
Friedrich Alexander-University of Erlangen Nürnberg;
dirk.riehle@fau.de

Free and Open Source Software Licenses Explained

Miriam Ballhausen, Bird & Bird, LLP

This installment of Computer's series exploring free and open source software confronts a pressing issue, free and open source software licenses: what they are, the rights they convey, and the restrictions they impose.

Free and open source software (FOSS) has been important to the software industry for a few decades. By now, the percentage of FOSS in the average application exceeds the amount of proprietarily licensed code (according to Black Duck's 2018 Open Source Security and Risk Analysis Report, available at https://www.blackducksoftware.com/open-source-security-risk-analysis-2018, applications, on average, contain roughly 57% FOSS). That is easily understandable given that FOSS allows engineers to save time, effort, and money while still solving the (coding) issue at hand. By contrast, FOSS licenses are found to be dull, and the task of having to consider the law and licensing is regarded as a bug rather than a tool, a safety net to ensure that the software remains free.[1]

However, once considered from that angle, FOSS licenses become an essential instrument, which, if understood, may be employed to achieve a set objective. To foster such understanding, this article first looks into what FOSS licenses actually are and then focuses on software freedom as a key differentiator between them and other software licenses (see the next section), before turning to the tools they employ to secure this liberty (see the section "It's All About Preserving the Freedom").

THE DEFINITION OF FOSS LICENSES

Initially, all FOSS was referred to as *free software*. The term *open source* was introduced in 1998 with the intention of clarifying that the software was not free but, instead, gave users more flexibility because the source code was readily available. The definitions of free software and open source software largely align and essentially include the same license terms (see "The Four

Digital Object Identifier 10.1109/MC.2019.2907766
Date of publication: 4 June 2019

FROM THE EDITOR

Welcome back to this column on open source software and how it is changing the world! After our start with an article about the innovations of open source, we will now tackle, head on, one of the more baffling aspects of open source: its licenses and how to use the software correctly. Often overlooked by researchers, this is still, after so many years, the topic at the top of many practitioners' minds.

The first author in this theme is Miriam Ballhausen, of the Bird & Bird law firm. In this article, she explains what free and open source software licenses are, the rights they typically grant, and the obligations they place on users as well as common prohibitions. This article is a solid foundation for the first in a series of articles on managing the (corporate) use of open source software. Future articles will discuss how to select open source components, to be license compliant, to work with the supply chain, and so forth. If you have comments or suggestions for future themes and articles, email me at dirk@riehle.org. Happy hacking! —*D. Riehle*

Essential Freedoms" section for details). To cover both aspects, software licensed under such terms is referred to as *FOSS*.

Simply put, FOSS licenses are distribution terms for software. They are necessary because software is protected under copyright laws (see the "Protection of Software Under Copyright Law" section). Unless rights of use are granted (licensed, see the "Software Licensing" section), third parties are not in a position where they can lawfully use the software. This applies irrespective of whether software is licensed as FOSS or otherwise (for example, proprietarily), but if software is licensed as FOSS, the distribution terms meet particular requirements (see the "Free and Open Source Licensing" section).

Protection of software under copyright law

In line with model provisions made available by the World Intellectual Property Organization (WIPO), Geneva, Switzerland, a computer program is "a set of instructions capable, when incorporated in a machine-readable medium, of causing a machine having information-processing capabilities to indicate, perform, or achieve a particular result." (The model provisions are available at www.wipo.int/mdocsarchives/AGCP_NGO_IV_77/AGCP_NGO_IV_8_E.pdf.) Software is usually referred to as *computer programs* in the respective legal texts and is widely protected as literary work (the same as books and this article) under applicable copyright law, provided that the software embodies an author's original creation. This protection was agreed to in the WIPO Copyright Treaty and is, for example, reflected in the United States by the Copyright Act of 1976 and in the European Union by the so-called Computer Programs Directive (Directive 2009/24/EC of the European Parliament and the Council of 23 April 2009 on the legal protection of computer programs) and its integration into member states' laws (for example, the German Copyright Act).

All rights of use to the software (such as the right to distribute and the right to modification) initially lay with the software developer. Exceptions to this rule may apply depending on the applicable copyright law. For example, in the case of employment relationships, the rights of use may lay with the employer. Third parties (anyone who is not the software developer) may only use the software if, and to the extent that, rights of use are granted (licensed) to them.

Software licensing

Software licensing means that rights of use to computer programs are granted. Such rights may be granted in various ways, including simple/single rights of use with all others having the same privileges or exclusively so that the licensee is the only one who may lawfully exercise a particular right. A license may be territorially restricted (for example, the European Union only) or granted worldwide; it may be restricted in time (such as what happens with hardcover books before they become available as paperbacks), or it may be perpetually granted. Finally, licenses may be granted for varying types of use, including reproduction in whole or in part; translation, adaptation, arrangement, and other modifications; and distribution of a computer program and making software publicly available.

FOSS licensing

These licensing options always exist irrespective of whether software is licensed proprietarily or as FOSS. All FOSS licenses simply make use of these options in a particular way. Rights of use are granted to the furthest extent possible to give users the four essential freedoms detailed in the next section. The rights of use only vary in how they are worded. Some are simple and straightforward, such as the rights-of-use provision in the Berkeley Source Distribution (BSD) licenses, as illustrated by this example from the BSD-2-Clause[2]:

> *Redistribution and use in source and binary forms, with or without modification, are permitted […].*

Others, such as the provision in the MIT license, are more elaborate but still directly worded[3]:

> *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, [...].*

Yet others, such as the Apache-2.0 license, use more legalese to describe the same, extensive rights of use[4]:

> *Subject to the terms and conditions of this license, each contributor hereby grants to you a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, sublicense, and distribute the work and such derivative works in source or object form.*

The definitions of free software and open source software largely align and essentially include the same license terms.

The four essential freedoms

By extensively granting rights of use, FOSS licenses ensure that users have the freedom to use, study, share, and modify software. Three of these four essential freedoms were first set out in Richard Stallman's definition of free software, which, after adding freedom zero, the Free Software Foundation (FSF), Boston, still relies on today. (See the FSF's "What is free software?" article available at https://www.gnu.org/philosophy/free-sw.en.html. For additional detail, see the FSF's definition of free software available at www.gnu.org/philosophy/free-sw.html.en#f1.) Accordingly, for software to be considered free, users must be granted the following freedoms:

1. to run the program as desired for any purpose
2. to study how the program works and change it so that it performs computing tasks as desired
3. to redistribute copies
4. to distribute copies of modified versions, thus giving the whole community a chance to benefit from the changes that were made.

Although the term *free software* was always intended to reference software liberty and not price, the ambiguity of the word *free* was widely assumed to prevent the business adoption of FOSS and FOSS licenses. In reaction to such a misconception, the term *open source* was coined in the late 1990s.

The open source definition

In comparison to the rather simplistic definition of free software, open source is defined through 10 criteria that software distribution terms must meet (see the Open Source Initiative's definition available at https://opensource.org/osd). Despite the term *open source*, these criteria go beyond the mere accessibility of source code.

Criteria one, three, and four essentially mirror the four freedoms Richard Stallman and the FSF defined. The second criterion requires the accessibility of the software's source code, which, previously, had been perceived as a precondition for the four freedoms.

1. The free redistribution of the software as a component of an aggregate software distribution that contains programs from several different sources may not be restricted. This includes a stipulation that the license grant may not require payment.
2. The source code of the software must be available, preferably, upon distribution or, at least, through a well-publicized means of obtaining the source code for no more than a reasonable reproduction cost, ideally by downloading via the Internet without charge. This includes a stipulation that deliberately obfuscated source code or intermediate forms, such as the output of a preprocessor or translator, are not allowed.
3. Modification and creating derived works must be allowed as well as distribution under the same terms as the license of the original software.
4. The right to distribute modified versions of the software may only be restricted to preserve the integrity of the author's source code. Distributing software built from modified source code may not be restricted. If the license restricts source code from being distributed in modified form, it must allow the dissemination of patch files with the source code so that the program can be modified at build time.

The other six criteria are usually less apparent, although they are often the main differentiator between FOSS and other licensing models under which software may be distributed free of charge (for example, freeware).

5. The rights of use must be granted equally to everyone.
6. The license must allow the software to be used in all fields of endeavor.
7. The rights of use granted by the license must be available to every recipient of the program without having to execute an additional license.
8. The rights of use must apply irrespective of, and separately from, the product the software is used in.
9. The license must be impartial to other software that is distributed with the licensed software. This includes a stipulation that it may not be mandatory for all other software to be open source also.
10. The license may not prescribe the individual technology or style of interface for the software.

IT'S ALL ABOUT PRESERVING THE FREEDOM

The focus of both definitions is on preserving software freedom. That is hardly surprising given the roots of the free-software movement (see the next section). Once software freedom is understood to be at the core of everything, many FOSS license obligations (or the lack thereof) become clear, comprehensible, and, for the most part, predictable.

The emergence of FOSS licensing

In the beginning and until the 1970s, all relevant software was FOSS. On the one hand, freely sharing software aligned best with the standard academic approach of publishing all research and results. On the other hand, software was only distributed as a part of hardware because a market for it, alone, would only be established in the mid-1970s once computer programs had been recognized as protected literary work under applicable copyright laws (see the "Protection of Software Under Copyright Law" section). This recognition also enabled copyright holders to charge license fees when they allowed third parties to use their software, meaning that, although the software's source code remained generally available, payments were becoming necessary to actually gain access to it, thus, instantaneously restricting software freedom, for which free access to source code is a prerequisite. The FOSS community grew from a countermovement against these developments and aimed to preserve software freedom, although freedom was interpreted differently by various FOSS licensing models.

By extensively granting rights of use, FOSS licenses ensure that users have the freedom to use, study, share, and modify software.

FOSS licensing models: Copyleft versus noncopyleft licenses

FOSS licenses can generally be divided into two models: copyleft licenses and noncopyleft licenses. (A list of FOSS licenses that was approved by the Open Source Initiative is available at https://opensource.org/licenses; a list of FOSS licenses commonly found in software and projects is available from the Linux Foundation's Software Package Data Exchange project at https://spdx.org/licenses/; and a list, in English, of categorized licenses is available from the Institut für Rechtsfragen der Freien und Open Source Software at http://www.ifross.org/en/license-center.)

Copyleft licenses are characterized by the obligation to distribute derivative works of the software (granted to the licensee under the copyleft authorization) under the terms of the respective copyleft license. While this general rule applies to all copyleft licenses, they each define independently which modification, alteration, or combination constitutes a derivative work (which requires them to be licensed under the copyleft terms). In practice, this means, for example, that changes made to files authorized under Mozilla Public License (MPL) 2.0 must also be licensed under the terms of MPL 2.0, and self-developed software that is linked to software licensed under General Public License 3.0 must be licensed under the same terms.

Copyleft licenses can further be categorized as strong copyleft licenses, on the one hand, and weak copyleft licenses, on the other. In the former case, the copyleft generally broadly applies to all derivative works (the term *derivative work* must still be interpreted for each FOSS license individually, determining, for example, on a case-by-case basis, if dynamically linked components qualify as derivative). In the latter instance, the copyleft only applies in certain situations (for example, only to changes made in the same file, as in the case of MPL-2.0). All copyleft licenses, irrespective of whether the copyleft is strong or weak, aim at preserving software freedom, with the focus, in this case, being on ensuring that the source code remains freely accessible.

Noncopyleft licenses, in turn, do not focus mainly on the free accessibility of the source code but, instead, on the licensees' freedom to determine the license conditions for works they create, even if they are derivative works of a third party's code. Accordingly, weak copyleft licenses do not oblige the licensee to apply the same license terms to derivative works but, instead, allow him or her to license them under freely chosen conditions (for instance, derivative works of BSD-3-Clause-licensed software can be distributed under the terms of the licensee's choice, including proprietarily).

Other typical license obligations

Apart from the copyleft provision (or lack thereof), there are other obligations, which are typically found in all FOSS licenses, including the duty to provide

1. the full text of the applicable license(s), thus ensuring the licensee is aware of all rights and obligations
2. the attribution notices.

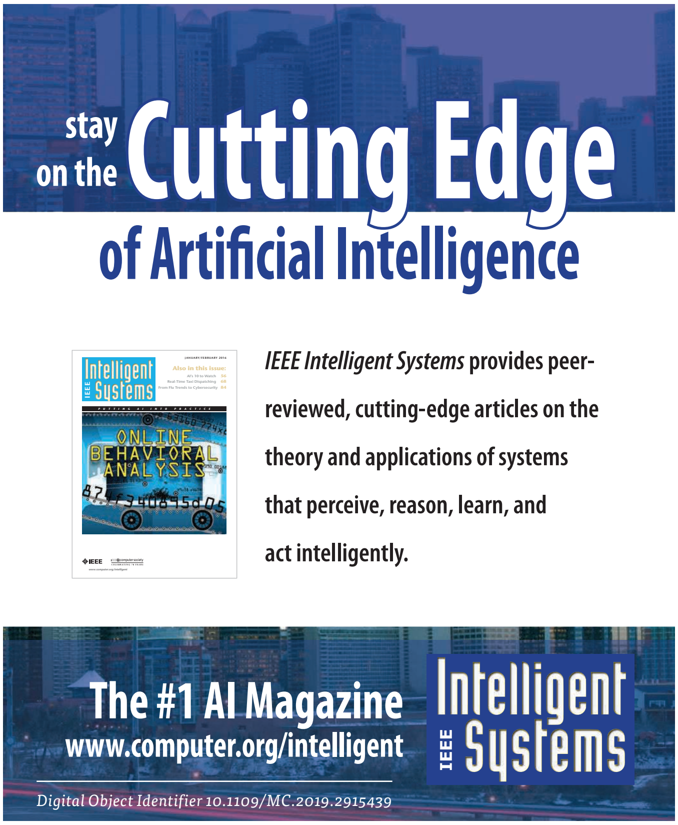
These (and other) license obligations, and how to comply with them in practice, will be the subject of the next article in this series. Their interpretation in the light of software freedom has, therefore, only been outlined here.

From a legal perspective, FOSS licenses are no different from any other copyright license. They, rather, only make use of the options available under applicable copyright law in a way that grants and protects software freedom to the furthest extent possible. Ensuring software freedom was the initial trigger for creating FOSS licenses and remains the strategic focus. Keeping that in mind allows one to easily understand and work with FOSS and FOSS licenses.

REFERENCES

1. D. Neary, "Gray areas of software licensing," LWN, 2012. [Online]. Available: https://lwn.net/Articles/481386/
2. Open Source Initiative, "The 2-clause BSD license," Opensource. Accessed on: Apr. 26, 2019. [Online]. Available: opensource.org/licenses/BSD-2-Clause
3. Open Source Initiative, "The MIT license," Opensource. Accessed on: Apr. 26, 2019. [Online]. Available: opensource.org/licenses/MIT
4. Open Source Initiative, "Apache license, version 2.0," Opensource. Accessed on: Apr. 26, 2019. [Online]. Available: opensource.org/licenses/Apache-2.0

MIRIAM BALLHAUSEN is a lawyer at Bird & Bird, LLP, Hamburg, Germany. Contact her at Miriam.Ballhausen@twobirds.com.



Digital Object Identifier 10.1109/MC.2019.2915439