

# Open Source Software Licensing

*Edward Kelly, JD*

---

*Edward Kelly, JD, is a partner at Ropes and Gray in Boston, Massachusetts.*

---

## Introduction

*Open source* describes a software development strategy; essentially it is a way to write computer programs. The OS model develops software in an open environment with the source code distributed for peer review and peer modification: See Eric Raymond's *Cathedral and Bazaar* (<http://www.tuxedo.org/~esr/writings/cathedral-bazaar>). The idea is to publish the *source code*, the human readable form of the program, for review and comment by anyone and everyone interested. The open and public scrutiny of the code will result in bugs being found more quickly and solved with greater elegance.

Thus, the intrinsic parallelism and free idea exchange inherent in the open source software development process has benefits over the traditional proprietary software development process. To get these benefits, the source code needs to be freely available and free to use and modify. These needs have led to the development of license agreements that make freely available the source code and the right to use and modify the source code: open source licenses.

But there is more to the OS model. There is a philosophy promoted by most open source developers that the knowledge and information provided in the published code is for public use, and the public should have the right to use the source code as the public sees fit. Still others go further. They take the position that copyright laws have failed to benefit actual software developers and instead benefit only corporations that take ownership of copyrights from employee-developers and use the copyright to control distribution of the software for company profit.<sup>1</sup> To address this, these developers promote releasing code to the public to be used by the public but only with an agreement from those using the code that they too will release any work derived from the code for public review and use.

To this end, the open source community employs copyright laws and software licenses to distribute code for public use, and some in the community have taken significant effort to leverage copyright law and licenses to ensure that code released to the public remains publicly available, even if that code is subsequently improved, modified, or expanded. This chapter discusses two families of OS licenses that differ in the requirements placed on the licensee and which, consequently, serve different members of the software development community.

## Commonly Used Terms in OS Licenses

Before going any further, it may be helpful to review some of the terms commonly used when discussing open source licensing of software. Some of the terms are technical and some are legal. It may be best to discuss these terms in the context of a simple example.

When computer programmers first start learning to program they often start with the simple example computer program called Hello World. Written in one popular computer language, the program looks like this:

```
#include <stdio.h>
main()
{
printf("Hello World!\n");
}
```

The Hello World program will cause the words “Hello World!” to be written on the screen. Indeed, the line `printf(“Hello World!\n”)` clearly suggests that this program will print out the phrase “Hello World!” This program code set out above is called the *source code*, a version of the program that a human can read and understand and change. For example, you can easily see that if you wanted this program to welcome AUTM, you could edit the program to read `printf(“Hello AUTM!\n”)`; and now the phrase “Hello AUTM!” will print out on to the screen.

The modification you made is called a *derivative work*, which is a term provided by the U.S. Copyright Statute that says that the holder of a copyright (which is the right to copy) is also the holder of all rights to prepare derivative works; that is, altered, amended, or supplemented versions of the original work. What is and is not a derivative work is often a complex question, but the analysis often begins by looking to see whether the new work was made by modifying a copy of the original work. If it was, this is a large factor in finding the new work to be a derivative work prepared from the original.

Although you can read this source code, a computer cannot. To put this program into a form that a computer can understand and run, the program sends the above source code for this Hello World program to a special program called a compiler that will compile down the source code into *object code*, essentially the 1s and 0s (binary form) that a computer can understand, but that is all but impossible for a person to understand.

Looking again to the source code for the program, you can see that the very first line is `#include <stdio.h>`. This is a statement that tells the compiler to link to a library called `stdio`. The `stdio library` is the standard input/output library, and it contains all the computer code necessary to carry out the `printf` command used in the Hello World program. A library contains sets of common computer operations, like printing to a computer screen, or math functions like quadratic equations.

Using standard libraries reduces the development time of software products by eliminating the need to repeatedly code these common operations and debug the result. The compiler will *link* the `stdio` library with the Hello World program and generate all the object code the computer needs to actually print the phrase “Hello World!” to the screen.

One or more of the terms *source code*, *object code*, *derivative work*, *library*, and *linking* appear frequently in most open source licenses, and the preceding should provide some useful description of the meaning of these terms.

## The Legal Structures of the Open Source Model

As noted above, there are two basic and related philosophies that drive open source software licensing. The first is that the value of the information and technology built into computer code is realized most effectively when it is published for everyone to see and released for everyone to use. This philosophy guided the development of several common licenses used by the OS community, which arose in the late 1990s, in response to the concerns of the hacker and free software communities. These licenses, discussed more fully below, grant to the license holder a right to use the code as the license holder wishes and disclaims all warranties and liabilities.

A second related philosophy is that programmers, at least certain types of programmers, are best-served when source code is freely available for review and use, and, further, that traditional notions of copyright restrict creativity and innovation and fail to compensate the creative party. This philosophy led to the development of the *copyleft license*, a license providing published code for use but requiring any modifications, amendments, or derivative works resulting from that published code remain publicly available for review and use. To this end, the copyleft license requires that, to the extent that any derivative work is released, that derivative work will be released with the published source code and with the right to use that derivative work for any purpose.

In essence, the open source licensing environment breaks down along these two types of licenses, with the first type being referred to generally and in this chapter as *less restrictive* and the second type being referred to generally and in this chapter as *copyleft*.

Moreover, the open source licensing environment includes a robust and vigorous membership that takes seriously these licensing issues and, to this end, have established organizations that offer examples of open source licenses and suggestions on how to use the correct license. Further, the open source community has added some teeth to these agreements by establishing organizations like the Free Software Foundation Inc. (<http://www.fsf.org>) that will pursue infringers, particularly companies that fail to comply with copyleft obligations.

As such, unlike other areas of licensing, the open source community has a rich body of available template licenses offered by organizations and communities for the purpose of helping developers create code under the open source model. These template licenses are commonly reviewed and selected by developers, including those at universities and research centers, as the licenses under which code is distributed.

## Less-Restrictive Open Source Licenses

The less-restrictive open source license provides the source code to the licensee with the licensee agreeing to disclaim all warranties. Typically these licenses are publicly and freely available and are delivered as shrink-wrap licenses that the licensee obtains upon download of the code. As the point of the OS license is to give the licensee the actual source code, it is understood that many of the licensees will use the source code to operate computers, machines, and other devices. A good example is the Apache Web server software (<http://www.apache.org>). This code is released as open source code. The vast majority of licensees merely use the code to run Web servers.

However, a substantial number of licensees have actually made derivative works from the Apache server source code. These modified versions of Apache can be used for any number of applications, including running Web servers from small handheld devices. As the modified code actually takes control of the hardware and may have access to files stored on the hardware, it is important to disclaim all liability, as there is always the possibility that the software may damage the hardware and data on the system. But for the concern over liability, the source code could be released freely and without concern. However, under the adage of “No good deed goes unpunished,” developers releasing their code to the public with copies of the source are quite wise to use a shrink-wrap license to disclaim all warranties.

The Massachusetts Institute of Technology, Berkeley Software Distribution (BSD), and Apache licenses, classic open source software licenses widely used in many open source projects, are examples of the less-restrictive open source licenses: The text of these licenses can be found at the Open Source Initiative, <http://www.opensource.org>. These licenses are very flexible and compatible with almost every form of open source license.

The most well-known of such projects are probably the BSDNet and FreeBSD Unix-like operating systems and the Apache HTTP server.<sup>2</sup> These licenses, as applied to the original licensed code, allow the code to be used in proprietary software and do not require that open source versions of the code be distributed. Projects created under these licenses, or derived from such code, may go closed and released under a proprietary license.<sup>3</sup>

The following is a closer look at the BSD license template. As previously noted, the BSD license, and templates for licenses like the BSD license, are available from several organizations, including the Open Source Initiative.

A BSD-like license begins with a notice of copyright:

Copyright (c), Year., <owner>

All rights reserved

The next clause provides for the use and redistribution of the source code, along with certain minimal attribution obligations:

Redistribution and use in source form and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of the <ORGANIZATION> nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

The license ends with a section of standard disclaimers that disclaim all warranties and liabilities:

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The effectiveness of these disclaimers may vary from jurisdiction to jurisdiction, with different countries having different rules as to the extent warranties may be disclaimed. Consulting with an attorney is important when crafting disclaimers.

In the end, this license grants the right to use the software in any manner the license holder wants, including the right to make modifications to the code and freely use and sell the modified code. Thus, this less-restrictive license gives the right to make derivative works and keep that work as proprietary code for commercial sale.

Others of these less-restrictive licenses have similar language, some being quite clear that the license holder can use the code for his or her own proprietary, commercial endeavors, such as the Apache license, which states:

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

## Granting Patent Rights under Less-Restrictive OS Licenses

To encourage the software development community to work with code being distributed under an OS model, some commonly used less-restrictive licenses contain grants to any patent rights held by the developer and necessary to use the code being distributed. For example, the Apache license includes a grant to any necessary patent rights held by the contributors, a term defined in the license to include the licensor:

### 3. Grant of Patent License.

..., each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. ...

The problem with this clause for a university is quite clear. It can reduce the value of any patents the university owns on inventions that are embodied in the open source code as it freely grants licenses to any party that also takes a license to the software. In the end, it is important to understand that many forms of less-restrictive open source licenses are offered for use by different organizations and communities. These licenses will differ in terms substantially, and the correct license for your purpose will depend upon your facts and intentions.

## Copyleft Licenses

Although there is no single copyleft license, the GNU General Public License (GPL) is generally considered the prototypical license for copyleft licensing of open source software. The GNU GPL is the license under which the Free Software Foundation Inc. (FSF) licenses GNU software. Many other organizations follow its lead and release software under the GPL. The GPL has been available in one form or another for more than fifteen years; the latest version, GPLv3, was released in the summer of 2008 and is available from the FSF at <http://gplv3.fsf.org/>.

The GPL includes a preamble, terms and conditions of licensing, warranty disclaimers, terms and conditions that govern software produced by the licensee using the GNU software, and other miscellaneous terms and conditions. The preamble sets forth the purpose of the GPL, namely: “to guarantee your freedom to share and change free software—to make sure the software is free for all its users.” The GPL is intended to apply to most of FSF’s software and any other program whose authors commit to using it.

The GPL permits users to distribute copies of programs covered by the GPL (including derivative works) and charge a fee for the service of providing copies of such works. GPL licensees may copy and distribute copies of the source code of the software that is covered by the GPL.

In exchange for the licenses granted to them, licensees are required under the GPL to include specified copyright notices, warranty disclaimers, and licenses with each licensed software program and modified work that is based upon or that incorporates the licensed software. Licensees must also provide prominent notices of any modifications they make to software.

However, the key feature of the GPL, and any copyleft license, is that the GPL requires that any derivative work of the licensed software be licensed as a whole at no charge to all third parties under the terms of the GPL; that is, a licensee’s customers must be permitted to view and work with source code and to copy, modify, and distribute derivative works of the licensee’s programs without payment of a royalty (other than the permitted service fee for transferring a physical copy). This clause prevents the licensee from using the source code to create a derivative work that can be licensed as proprietary software under a commercial software license.

More specifically, the GPL includes the following statements, terms, and conditions:

Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program—to make sure it remains free software for all its users.

The GPL goes on to say:

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

The terms of the GPL allow you to charge for your software, although given that the GPL allows copies to be made freely, it is difficult to understand why someone would pay a meaningful royalty.

Turning to terms and conditions, the GPL implements the copyleft philosophy by using licensing clauses that restrict how you can *convey* copies of the original work or copies of any derivative work. The GPL, in Section 2, grants any person who “comes into possession” of the software the right to use the software, modify it, make copies for that person’s use, and provide it to others with the express purpose of having those others modify the work and provide the modified work back. Specifically, Section 2 of the GPL states:

#### 2. Basic Permissions.

... You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force...

All this can be done and none of it requires the release of the source code for the work or modifications to the work as all these activities are essentially related to mere use of the software by a person who has a copy of the software.

However, once a party chooses to begin *conveying* (i.e., providing) copies of the software, or modified versions of the software, to others for the others to have and use, then Sections 4 and 5 of the GPL become relevant and active. Sections 4 and 5 of the GPL require that any conveyance of the software or modified versions of the software be accompanied by copies of the source code, or by instructions on how to obtain the source, and with a grant of the right to use, modify, and copy the software or modified versions of the software.

#### 4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; ...

#### 5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

- a) The work must carry prominent notices stating that you modified it ...
- b) The work must carry prominent notices stating that it is released under this License ...
- c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. ... This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.

As set out in the above clause c), additional terms and restrictions to use, modify, or copy the software or modifications of the software are strictly forbidden, or at least controlled. Section 7 of the GPL allows some very minimal modifications to the license, such as allowing for better warranties.

The result is that the GPL licenses the software with the right to freely use, copy, and modify the software, but with a restriction that prevents a user from redistributing the software or any derivatives as proprietary code under a traditional commercial license. Importantly, this restriction also applies to software that is incorporated into hardware devices, such as MP3 players, personal digital assistants, cell phones, and other devices.

The GPL explicitly requires that software licensed under the GPL and incorporated into a hardware device must be accompanied by the source code for that software and instructions for changing the software within the device. In this regard, companies incorporating GPL-licensed software must take care; a recent settlement between the Software Freedom Law Center and Verizon Communications Inc. required Actiontec Electronics, a supplier of Verizon hardware that violated the terms of the GPL, to appoint an open source compliance officer and to pay an undisclosed amount.

## **The Lesser GPL**

The strictness of the GPL raises some concerns within the open source software development community. In particular, the use of software libraries that are licensed under the GPL has caused some particular concerns. Recall that the Hello World example included the `stdio` library, which was compiled along with the program. The concern is that using a library licensed under the GPL (such as a licensed version of the `stdio`) would bring the entire Hello World program under the terms of the GPL. In many cases, the library provides merely standard code and technology and its inclusion into the program is more for the convenience of not having to rewrite these standard libraries than to take advantage of any particularly innovative programming present in the library.

To address this concern, the GNU project offers a second principal license that can be used with any code, but is particularly suited to use for libraries. As opposed to the GPL, the GNU Lesser GPL (LGPL) permits use of the library in proprietary programs without having the entire program brought under the GPL (for the text of this license, see <http://www.opensource.org>).

The LGPL provides definitions to help more clearly explain the limits of the license. Specifically, the LGPL helpfully defines certain terms:

“The Library” refers to a covered work governed by this License, other than an Application or a Combined Work as defined below.

An “Application” is any work that makes use of an interface provided by the Library, but which is not otherwise based on the Library. Defining a subclass of a class defined by the Library is deemed a mode of using an interface provided by the Library.

A “Combined Work” is a work produced by combining or linking an Application with the Library. The particular version of the Library with which the Combined Work was made is also called the “Linked Version.”

In Section 4, the LGPL allows the library to be used with another application to create a combined work, and the combined work may be released under the terms of “your choice:”

#### 4. Combined Works.

You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the portions of the Library contained in the Combined Work and reverse engineering for debugging such modifications, if you also do each of the following:

- a) Give prominent notice with each copy of the Combined Work that the Library is used in it and that the Library and its use are covered by this License.
- b) Accompany the Combined Work with a copy of the GNU GPL and this license document.
- c) For a Combined Work that displays copyright notices during execution...
- d) Do one of the following:
  - 1) Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.

- 2) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time a copy of the Library already present on the user's computer system, and (b) will operate properly with a modified version of the Library that is interface-compatible with the Linked Version.
- e) Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding Source and Corresponding Application Code. If you use option 4d1, you must provide the Installation Information in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.)

Thus, the LGPL provides copyleft restrictions on the library code itself, but does not apply copyleft restrictions to a program that merely uses the library.

## Conclusion

The open source development model, whether under less-restrictive licenses, the GPL, or the LGPL, appears to be here to stay, and it may work well for many projects and efforts where a royalty stream is unlikely or unnecessary. However, copyright holders should take care to understand the open source model before embarking on an open source strategy. In particular, universities should evaluate how their community currently uses open source software, whether the advantages of open source software development apply to them, and whether their licensing goals are consistent with the GPL.

## Notes

1. See St. Laurent, *Open Source & Free Software Licensing* (O'Reilly 2004).
2. See St. Laurent, p.14.
3. See St. Laurent.