



## A Glossary of Open Source Legal Terms for Engineers

By Michael Barr

*Updated November 27, 2018*

*When using software developed by others, such as Linux, within an embedded system design, it is recommended that developers consult with an intellectual property lawyer to read license agreements.*

The following is a brief glossary of licensing terminology to help engineering teams understand legal terms related to open source software.

### Proprietary software

Proprietary Software is owned exclusively by an individual or company and is protected under copyright law. Heavy restrictions are usually established for the use, modification and distribution of proprietary software, and the source code is typically kept secret.

In the most traditional licensing model for commercial software, no customer is allowed to examine the source code for the product. The one exception to this is a customer willing to pay an additional source code license fee. If a developer does buy the source code for such a product, the purchaser cannot publish that code or otherwise cause it to fall into the possession of anyone outside the licensed group. The rights to modify the source code may be restricted.

### Shareware

Shareware is software that is distributed for free, and may even be passed on to others. Users of shareware products are honor-bound to pay the developer's registration fee, whatever it may be. A variant on this licensing scheme, termed crippleware, will not function fully until the registration fee is paid. The source code is typically not included.

### Freeware

Freeware is shareware with no registration fee.

### Free software

Free software is software that is distributed free of charge whereby users can modify and redistribute without restrictions. Focused on access to the source code, the free software movement is as much a political organization as anything else. Under the free software licensing model, developers are even allowed to charge for their distribution of the software. However, these broad rights are conditioned upon the users' commitment to provide similar access to their code modifications and to never narrow the licensing rights as a condition of distribution.

Proponents of free software generally believe that all information, especially source code, has a right to be free. Therefore, free software is being in the "free speech" sense—not as in "free lunch." They advocate attachment of a "copyleft", which says that "anyone who redistributes the software, with or without changes, must pass along the freedom to further copy and change it."

### Public domain software

Public domain software is similar to free software, but less restrictive. Public domain software is source code that can be used, modified, or redistributed in any way. Users are free to make changes to the software and keep those changes proprietary. Developers can even choose to charge for the original code or a derivative, without providing any access to the source code. In that way, anyone can make use of public domain software in any way, without needing a lawyer.

## **GNU General Public License (GPL)**

Since 1984, the goal of the GNU project has been to develop a complete UNIX environment that is licensed as free software. Although some of the code involved is public domain, the vast majority is licensed under GPL. GPL is a specific implementation of copyleft. This is analogous to copyright, in which there is a general right that is implemented in various ways in different contracts and print and electronic publications.

General Public License (GPL) prohibits proprietary patents related to modifications of the software, prohibits royalties, and requires that the same terms be attached when redistributing the software or a derivative of it. Developers can create software and then license it under these same terms. Use of the GPL language is not restricted to GNU-related projects. (Their copyleft is not copyrighted!)

The popular GNU compiler and associated tools are licensed under GPL. This means that anyone making a new and improved GNU compiler must give their new source code back to the community. However, it is important to note that this does not mean that software built with the compiler must also become free. It is legal to use a free software tool to produce proprietary software.

## **GNU Lesser General Public License (LGPL)**

The LGPL is used to license free software so that it can be incorporated into both free software and proprietary software. The rules are basically the same as GPL, with one major exception: the requirement that developers open up the source code to their own extensions to the software is removed. So, while LGPL components remain free software, they can be included within a larger proprietary software package.

The downside of the GPL, particularly for embedded developers, is that it's designed to discourage the creation of proprietary software and to encourage free software. If developers want to build their firmware around a GPL package or library, they are forced to give away the source code to their firmware. But this is not a problem with an LGPL package, like the GNU C standard library, which can be legally included as part of proprietary software.

There's more info about working with GPL and LGPL software in [Embedded Linux](#) and [Copyright Law](#).

## **Open source**

There is no clear definition for "open source software" and no standard license. Many companies are using the term open source these days, but in far different ways. While the idea is similar to that of free software (developers can generally still use, modify, and redistribute the software), there is far less emphasis on the right of the source code to be free.

While not as true with respect to Linux, many open source companies seem to be unwilling to give up central control of their software. (Free software has no owner.) It is important to note about open source software, particularly for embedded developers, that the licensing terms are more like LGPL than GPL. Developers are typically free to add their own proprietary software to the open source code and produce a proprietary result. While this is not in keeping with the goals of the free software movement, open source software is more aligned with the newer open source movement than in opposition to it.

You can learn more about this subject from the [Software Freedom Law Center](#) website.

Barr Group provides testifying expert witnesses and software source code analysis teams to support complex litigation, including litigation involving product liability and infringement of intellectual property such as patents and software copyrights. [CONTACT US](#)