COSCA/NACM JOINT TECHNOLOGY COMMITTEE NCSC National Center for State Courts

Open Source Licensing Overview

Overview

The Open Source community is a movement towards making software available to others. Those “others” can be end users, who use the software as-is. Those “others” can also be developers who then continue development on the software or integrate it into software of their own. Open Source software covers a wide area. There are Open Source word processors, spreadsheets, web browsers, and entire operating systems. Pretty much any sort of software is available in an Open Source variety.

The Open Source world has a wide variety of licenses available. The Open Source Initiative currently lists 58 different Open Source licenses (www.opensource.org/licenses/). While 58 is a large number from which to choose, most of these licenses share a number of features.

Commonalities

All Open Source licenses seek to make code easy and desirable to use and share. To this end, they have many commonalities:

- *They allow bundling.* Software may be bundled with other software and this entire bundle may be either sold or given out for free.
- *They do not restrict bundled software.* Other software distributed in the bundle need not be covered under the same license.
- *They allow modifications.* The software may be modified.
- *They allow distributions of modifications.* Modified software may be distributed.
- *They allow derived works.* Creation of derived works that extend the software are allowed.
- *They do not discriminate as to recipients or contributors.* There are no restrictions on persons, groups, or fields of endeavor. Anyone can use. Anyone can modify. Anyone can distribute.

Differences

Again, all Open Source licenses seek to make code easy and desirable to use and share. Unfortunately, these aims are often in conflict and the different licenses available seek to reconcile this conflict in different ways.

On one hand, Open Source developers in general want others to be able to use their work, either as end-users or as developers. On the other hand, Open Source developers want their work to stay open. That desire may extend to wanting any derivative works created from their work to also be kept open.

These two desires can be in conflict. A commercial developer may want to use some open source project as a starting point, but still be able to profitably sell the resulting derivative work, as well as retain rights to the intellectual property (IP) that they added to the derivative work. In these cases, requiring that derivative works remain open may limit adoption of a work by other developers.

The balance is between encouraging the adoption and use of software versus ensuring that the software remains open. Different open source licenses deal with this conflict in three different ways.

"Encouraging Adoption" Over "Protecting Openness"

This class of license seeks to make software as easy as possible for others to take and use. The primary form of this license is simply donating the software to the "Public Domain." When an author puts her work in the Public Domain, she gives up any and all rights to it. Anyone else can take the software and use it, modify it, or even sell it as their own. If they sell it, they can sell their copy under any sort of license they desire. It can be open. It can be highly restrictive. Releasing software to the Public Domain is the easiest way to make software available to all and to encourage adoption and use by other developers. The downside is that there are no means to ensure that those developers will make their work freely available. Once you place something in the Public Domain, it is no longer yours and you have absolutely no control over how it is used.

"Protecting Openness" over "Encouraging Adoption"

The GNU General Public License (GPL) is the main license of this type. Under the GPL, if derivative works are distributed to others, the derivative work must be licensed under the GPL. It prevents others from adding restrictions to copies that they in turn distribute. This ensures that GPL-ed software always remain open.

Additionally, the GPL requires that if a distribution is in the form of binary code, then the source code must either be included or be freely available.

The downside to the GPL is that commercial developers are reluctant to use GPL-ed code because it requires them to release their derivative works with the same license. Including a small amount of GPL-ed code in a much larger piece of proprietary software can render the entire package GPL-ed. If your business is dependent on selling that piece of software, this can be a huge problem.

The full text of the GPL is available from the Open Source Initiative:
www.opensource.org/licenses/gpl-license.php

Balancing "Protecting Openness" With "Encouraging Adoption"

Most other open source licenses strike a balance between releasing to the Public Domain and releasing under the GPL. Most of these licenses allow for other developers to include the released software in their own works without subjecting their own intellectual property to "restrictions" of the GPL. Typically, a copyright notice giving credit to earlier authors is required. Examples of these alternative licenses include:

Lesser General Public License (LGPL)

The LGPL is a special version of the GPL designed for code libraries. Unlike the GPL, it allows software to link in LGPL-ed software without the software doing the linking becoming GPL-ed itself. Like the GPL, the LGPL does require that software that *incorporates* LGPL-ed also be under the LGPL. It also has the same requirements that source code be made available in those cases. The full text of the LGPL is available from the Open Source Initiative: www.opensource.org/licenses/lgpl-license.php

Berkeley Software Distribution (BSD) License

The BSD allows redistribution, but requires that copies include the BSD copyright notice. Additionally, derivative works cannot claim that those earlier copyright holder endorse the modified version. The full text of the BSD License is available from the Open Source Initiative at: www.opensource.org/licenses/bsd-license.php

Massachusetts Institute of Technology (MIT) License

The MIT License allows redistribution. It only requires that the copies include the MIT copyright notice. The full text of the MIT License is available from the Open Source Initiative at: www.opensource.org/licenses/mit-license.php

Open Source Initiative

The Open Source Initiative promotes Open Source and provides a full list of Open Source licenses at:
http://www.opensource.org/licenses/

Other Issues

Open Source Misconceptions

There are two main misconceptions about Open Source software. The first misconception is that you cannot sell Open Source software. In fact, you can, for as much as you want. Even the GPL allows you to sell software. However, when you sell it, you pass along those same GPL rights of redistribution. This makes it difficult to rely on selling Open Source software as a revenue stream. You can sell your software to Customer A. Then you can try to sell it to Customer B. However, Customer A is free to give the software to Customer B for free. Basically, you're only assured of one sale. Because of this, most companies that produce Open Source software rely on installation, customization, and support fees for revenue.

The second misconception is that if you modify Open Source software, you must provide it to others. In fact, the various protections of Open Source software apply only if you decide to distribute the software. You are, however, under no obligation to distribute the software at all. You can take a piece of Open Source software and modify it all you like. You can even incorporate it into your own software. And you are not required to tell anyone at all about it. It's only if you redistribute your changes that the various license provisions trigger. There is no Open Source mechanism that requires changes to be returned to the community.

Moderation

While Open Source projects are usually community driven, some amount of moderation is needed, particularly as projects grow in size. Projects need some entity that provides direction and prevents projects from proceeding down non-productive avenues. Individuals often direct even large-scale Open Source projects; however a steering committee may be more appropriate for large projects or projects with a widely diverse community.

In many projects, if a large segment of a community disagrees with the direction of a project, they can also start their own version, a process known as “forking.” The process allows those with alternate ideas to try them out without delaying or harming the main project.

Hosting

Open Source projects generally require an online home where interested parties can access code and documentation, as well as provide feedback and contributions. Hosts can be as simple as individual web sites where code is posted. Generally, large projects are hosted on content and code management systems specifically designed for collaborative development. Here are two examples.

Government Open Code Collaborative

The Government Open Code Collaborative is a voluntary collaboration between public sector entities and non-profit academic institutions for the purpose of encouraging the free sharing of computer code. More information is available at:
www.gocc.gov

SourceForge

SourceForge is the world's largest Open Source software development web site. It acts as a centralized resource for managing projects, issues, communications, and code. It hosts projects at no charge. SourceForge is geared towards technically adept developers and can be confusing for newcomers to Open Source development. More information is available at:
sourceforge.net