



Legal Issues in Open Source Software

Open source is increasingly popular with software developers. Having once been the preserve of geeks and hobbyists, open source is now part of the mainstream of software development and offers numerous advantages for developer SMEs. But when it comes to licensing, there are a whole set of legal issues for developers to consider.

Open source (OS) challenges many of the conventional notions of intellectual property (IP). Rather than wrap everything up in legal straightjackets that prevent others from accessing the technology, open source makes the key asset – the source code – freely available for others to use and modify as they wish. Redistribution of the new, modified solution is allowed subject to certain licensing conditions.

An open source license assigns copyright (and patents, if held by the author) to the end-user and re-distributor without compensation. Thus, for example, SMEs can leverage an application at no cost, use it in the course of commercial business, and profit by it in interactions with their customers.

But as open source becomes more popular and starts to collide with the world of 'closed' or proprietary software, it brings a whole new set of legal issues:

- How can developers choose between the huge range of OS licenses available?
- What can developers do to ensure they don't infringe upon existing software patents?
- Are the provisions of OS licenses enforceable and meaningful in law?

The process for choosing a license, reviewing code and launching a product without liability concerns becomes more vexing as the open source model expands.

Types of Open Source License

Over fifty open sources licenses are now in existence. Research shows around two-thirds of open source projects utilize the GNU General Public License (GPL). Other popular licenses are the Berkeley Software Distribution (BSD) and Mozilla licenses.

Open source licenses fall into four distinct types:

1. **Academic licenses:** Representing the most 'free' of open source licenses, academic licenses place no requirements whatsoever on the license user - there's not even a requirement for the user to share modifications or redistribute them. However, they prohibit the leveraging of the original licensor's name as an endorsement in marketing efforts. Licenses in this category include the BSD, MIT and Apache licenses.
2. **Reciprocal licenses:** These require that any derivatives of the software be released under the same license, and that the source code must be released. The resulting new software must also be free. The intent of reciprocity is to ensure that a growing universe of free software emerges, and that original works - as well as modified and new efforts - remain free to users. The centerpiece of this category is the GPL, which is used in software such as Linux and MySQL.
3. **Standards licenses:** Standards licenses seek to create a standard code base so that the originating author can come back to it and evolve it without difficulty. Modified and redistributed sources usually have to be distributed as plug-ins or patches, so as not to modify the core.
4. **Content licenses:** These cover elements aside from code, such as art, copy and audio/video. The most well known is the Creative Commons, although this is not strictly an OSS license.





Legal Issues in Open Source Software

Legal Implications of Open Source

Whatever the licensing terms, open source as a whole raises a variety of legal issues, some of which have yet to be tested in the courts.

- **Validity and enforcement:** The most important legal aspect of OS licenses is whether the licenses are valid and enforceable under national laws. As with proprietary 'shrink wrap' licenses, the end-user does not sign any license agreement but receives information alongside the program. Sometimes this may only comprise a reference to the license conditions. Some experts question whether such agreements can be legally binding when the user has no opportunity to negotiate terms, especially for consumer markets.
- **Moral rights:** This is a legal term for the personal rights of authors and creators. The most important moral rights are the right to be identified as an author and to object to derogatory treatment of the work. Although some open source licenses require the original authors to be identified, moral rights may require this too.
- **Warranties:** Open source licenses usually contain provisions that disclaim warranties and many attempt to limit liabilities. However, complications can arise in consumer relations and situations where the licensor charges for the service. Requirements can also vary from one country to another. Although these types of provisions are also very common in proprietary software, legally it's still a bit of a grey area.
- **Software patents:** Since all the source code of open source programs is usually freely available, it is very easy to find programs which may infringe software patents. But as open source programs are often created by many programmers, it can be very difficult to find exactly who was responsible and start legal proceedings against them.
- **Derivative works:** Probably the most important difference between various open source licenses is in their approaches to derivative works. This issue may also have significant legal implications, in particular for companies building a business on open source. For instance, the GPL – generally considered to have the strictest approach – requires that every modification to the source code must also be under the GPL.

Staying Legal

Open source is increasingly popular with developers, who value its adaptability and flexibility. But it is not necessarily as 'open' as some make out. While using OS for your own purposes is fairly straightforward, IP and license considerations become critical if the source is being modified, packaged into another solution, and distributed. You need to undertake regular reviews to ensure you comply with the provisions of the OS licenses and any copyrighted and patented code – and document what you do so as to prove it later on.

In understanding the licensing process, it's important to distinguish between copyright, trademark and patents. All of these elements play a role in the software we use every day. In some cases, the freedom from patent risk has been included in the license (e.g. Sun Microsystems' CDDL). Many of the licenses protect users from patent requirements of the original software, but cannot necessarily extend that protection once you modify and redistribute the source code or binary.

The choice of license has a major impact on risk. As explained above, each license category has its own particular purpose, whether it's to ensure end-user freedom, prevent commercial use, or preserve a standard code base. Users can switch licensing schemes after they've made their selection and distributed software; but applying such a change retrospectively to existing code and license agreements is difficult. Hence, developers need very carefully to select a license they can live with for the long term.

For example, if a developer foresees only selling support and customisation services over the long term, choosing a reciprocal license that may prevent the sale of the software itself would be sufficient. However, if there's a chance that a future application may become partially proprietary while including original or modified open source, an academic license may be a better route.

Useful Links

Open Source Licensing: Software Freedom and Intellectual Property Law, by Lawrence Rosen, published by Prentice Hall, 2004.

OpenAdvantage, the West Midlands Open Source Solutions Centre, www.openadvantage.org