

Which open source license should I use for my software?

by Lawrence Rosen

OK, I'll admit it, I wrote this question myself because I've been asked it so many times I wanted to see it answered in print. As an attorney for open source companies and projects, I am often requested to select a license (or to bless my client's selection) from among the OSI approved open source licenses. [All the licenses described here are listed on the OSI website at www.opensource.org.]

The question I asked above puts the cart before the horse. What drives the license selection process is the client's business strategy, and not the other way around. Here is how the issue should really be addressed:

- Do you intend to make money from licensing the software or from providing ancillary services like installation and training? There's nothing illegal about using a proprietary software license if that's what your business model dictates. Of course, as an advocate of open source, I'd try to convince you that there are many advantages to non-proprietary business models -- but the client is the one to make the final decision.
- What degree of freedom are you willing to grant to your licensees to modify your software? There are open source licenses (e.g., BSD-type) that impose virtually no restrictions on licensees; they can modify the licensed software and create proprietary versions without restriction. There are other open source licenses (e.g., GPL-type, more typically known as "free software" licenses) that require the licensee's modifications to be licensed back under that same license; this "inheritance" characteristic is an advantage if you want your licensees to have to reciprocate if they benefit from your contribution to the community. There are still other open source licenses (e.g., MPL-type) that impose an intermediate level of freedom; modifications to individual files containing licensed code must be licensed back, but new files that merely work with the licensed code need not be.
- Are you willing to grant warranties that the software will be "merchantable" or "fit for a particular purpose"? If your software is royalty-free, you probably can't afford a warranty. On the other hand, you may want to charge for your open source software and use the profits to provide a warranty and other forms of service.
- Is your software so well known that the main asset you need to protect is your trademark rather than your code? An excellent example of this is Apache. Their license allows you to do almost anything you want with the Apache code, but you'll have to change the name. For example, I can't imagine anyone wanting to buy Larry's Webserver, even if it works just like Apache. If you have a trademark to protect, make sure your license contains appropriate terms relating to that.
- Have you considered the possibility of dual licensing? The owner of a copyright in a software program always has the option to use multiple licenses. For example, you may want to license your software under the GPL and simultaneously provide a proprietary version for those of your customers who are afraid of the GPL's inheritance features; that unreasonable fear can be treated as a revenue opportunity.

- Have you considered using different licenses for different parts of your software? For example, client software might be distributed under an MPL-like license, but server software might be distributed under a proprietary license. That way, you could make money from the bigger customers that will pay to license your server software, and simultaneously build a large customer base with free client software.
- What is it about your software that you are really trying to protect? Is it the code itself, or the standards that are implemented using that software? A license like SSISSL allows anyone to develop modifications of licensed software as long as the licensee complies with all requirements set out by a standards body; a licensee who elects not to comply with the specification must publish a royalty-free reference implementation of the modifications so that the standard cannot be abducted by another company.
- Are there any patents that relate to your software? If so, you will have to consider licensing your patents along with your code. You may also want retaliate against any licensee who takes your free software and then turns around and sues you for patent infringement. Various licenses on the OSI approved license list take different approaches to this problem. Some include a strong retaliation clause, others a weaker version that may be less threatening to customers with a large patent portfolio.

This is not an exhaustive list of considerations. You and your attorney should understand your business situation thoroughly before you decide on a license.

Even after you answer these questions, you will still need to decide whether to invest in the attorney resources necessary to create your own license, or simply have your attorney modify an existing license to meet your needs. If you choose to create your own license, your attorney will be able to tailor your license to your unique business requirements. On the other hand, modifications to an existing license may be sufficient. Consult an attorney familiar with your business to advise you.

Remember that your business objectives guide the choice of license. Anyone who ignores your business needs and whose first words to you are “use this license” is the wrong horse to push your cart.

Lawrence Rosen is an attorney in private practice in Redwood City, California (www.rosenlaw.com). He is also executive director and general counsel for Open Source Initiative, which manages and promotes the Open Source Definition (www.opensource.org).

Legal advice must be provided in the course of an attorney-client relationship specifically with reference to all the facts of a particular situation and the law of your jurisdiction. Even though an attorney wrote this article, the information in this article must not be relied upon as a substitute for obtaining specific legal advice from a licensed attorney.