

## Dealing with Patents in Software Licenses

by Lawrence Rosen

Many members of the open source and free software community oppose software patents. Software patents, they say, hinder the advancement of software art, and as such counter the beneficial effects of open and published source code.

The risk of infringing a copyright is much less than the risk of infringing a patent. You can avoid infringing a copyright by practicing good clean-room practices and writing your own independent version of copyrighted software. On the other hand, a software patent can stop you from making, using or selling the patented invention even if you didn't copy the inventor's software. This means that you may not be able to avoid infringing a patent no matter how careful you are. Someone you never even heard of can inform you that he or she has a patent and, if you cannot "invent around" that patented invention, your open source project may be stopped dead in its tracks.

Like them or not, however, software patents are a reality. Software patents have been blessed repeatedly by Congress and the courts, and by the laws of many other countries. Given this reality, it is important to understand the implications of patents for software licenses, so that you can select a license that meets your philosophy and goals.

Consider, from the viewpoint of a licensee of open source software, three kinds of patents. These are (1) patents owned by the software licensor, (2) patents owned by third parties, and (3) patents owned by the licensee or by the licensee's downstream sublicensees.

**Licensor patents:** Suppose you took a license for some open source software and then discovered that the licensor has a patent on that software that was not included in the license grant. Without a license to that patent, you could not make, use or sell the software. Whenever I review a software license for a licensee, I make sure there is an express grant "under claims of patents now or hereafter owned or controlled by licensor, to make, use, sell, offer for sale, have made, and/or otherwise dispose of licensed software or portions thereof." Many open source licenses, including the BSD license, contain no such provision. In those cases, a patent license may be implied, but I don't recommend relying on an implied license. Wherever possible, make sure you have an explicit license to any necessary patents held by the licensor.

**Third party patents:** A software licensor may not be aware of all patents that apply to its software. Some third party may suddenly announce that it owns a patent that covers some aspects of the software. As a licensee of infringing software, you may have to stop using the software despite the license. To deal with this situation, proprietary software licenses often include an *indemnity* clause, by which the software licensor indemnifies its licensees against third party patent claims, promising to refund license fees, provide non-infringing versions of the software, or obtain licenses to third party patents, if third party patents come to light. But open source licenses don't usually contain indemnity clauses, because licensors of open source software usually do not collect license fees sufficient to cover the potential costs of the indemnification. So, for most open source software, the license is "AS IS" without any warranty of non-infringement. Open source licensees beware! The risk of third party patents is usually borne by the licensee.

**Licensee patents:** This situation is more subtle than the previous two. Here, we are dealing with a licensee's own patents or, perhaps even more important, the patents of downstream sublicensees of the open source software. Licensors sometimes include a "patent

retaliation” clause in their licenses to help defend themselves against infringement claims. In essence, a patent retaliation clause says that if a licensee (or his downstream sublicensee) sues the software licensor for patent infringement, the license to the software terminates; a licensee or sublicensee can’t both use the software and also sue the licensor for patent infringement. One open source software licensor justified its use of a patent retaliation clause this way: “Maintaining the defensive use of patents will minimize unfairness.”

There are two general forms of patent retaliation clause. The first, a so-called **weak** patent retaliation clause, says something like this: If you take a license to my software, and you later assert your patent against me relating to this software, then your license to my software is terminated. The same patent retaliation clause usually applies to your sublicensees; if one of your sublicensees sues your licensor for patent infringement, your sublicensee’s license to the software is terminated. I personally support weak patent retaliation clauses in licenses, because I think they balance the interests of the licensor and licensee. Licensees and their sublicensees should not be able to benefit from free and open source software while at the same time forcing the licensor to pay royalties for patents embodied in that very software.

A so-called **strong** patent retaliation clause says something like this: If you take a license my software, and you later assert your patent against me relating to anything at all, then your license to my software is terminated. Again, this patent retaliation clause also usually applies to your sublicensees. I personally believe that strong patent retaliation clauses harm the open source community far more than they helps licensors, because companies with large patent portfolios will resist adopting open source software if the software licenses effectively allow licensors to infringe the licensees’ (or sublicensees’) other unrelated patents with impunity. Indeed, as a licensee of open source software, you may find that a strong patent retaliation clause inhibits your ability to disseminate derivative works, because your downstream sublicensees may refuse to accept such a “virus” that affects their ability to enforce their unrelated patents against your licensor.

Apple is an example of a company that insists upon including strong patent retaliation clauses in its licenses. If a licensee adopts certain Apple software, the licensee’s use of that Apple software is conditioned upon the licensee not suing Apple for patent infringement in any other matter. So to the extent that Apple’s software becomes widely used -- perhaps even indispensable -- in a licensee’s company, Apple can now infringe any of that licensee’s other patents without fear of an infringement suit. A company with a large patent portfolio might be reluctant to adopt Apple software if it means effectively cross-licensing all of its other patents to Apple. In fact, I would probably warn any client of mine to consider avoiding Apple’s open source software for that very reason.

Whatever your philosophy about software patents, it is important to understand the ways that patents can affect software licenses. When you license your software to others, or when you take licenses to software for your use, you should make sure that the license fairly expresses your own philosophy and goals relating to patents.

*Lawrence Rosen is an attorney in private practice in Redwood City, California ([www.rosenlaw.com](http://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](http://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.