

The Virtual Lawyer

Briefings on Technology and Business Law

October 2008



Open Source—The Agony and the Ecstasy

Legal Disclaimer

- Law is ever-changing. This briefing is a synopsis only and cannot substitute for personal legal advice. Everyone's facts and circumstances are different and you should not rely on the contents of this publication to make substantive legal decisions. Please contact me for a further consultation.

Open source software is both a blessing and curse. It's truly a blessing for the software developer because the huge and growing open-source libraries provide tested and standardized functionalities that make a geek's life that much easier. Using open-source, companies are able to reduce the time to market for new products, lower costs for maintenance and licensing and save development resources for unique and proprietary functionality that sets one product apart from another.

Even "proprietary" software these days incorporates open-source to a greater or lesser degree. For instance, I am counsel to a U.S. software company that offshores a good deal of its development. The proprietary product contains a goodly collection of open source which, when I dutifully listed it in the license, took an entire page. This is by no means unusual. When this same company when through a venture capital round, the use of open source was accepted as a matter of course. But you can be sure that counsel for the VCs worked over my list.

But open source can also be a curse – for the software company's general counsel, that is. But who cares about the lawyer, anyway? Well, you should, because he is the guy that's going to keep you geeky behind out of trouble when you license your product, obtain financing or sell your company. Think of the headaches in a large organization: in procurement terms, your client might obtain "proprietary" code to sell, OEM or incorporate in a larger application. It's our job to sort this all out. So, where to begin?

As originally envisaged, the open source movement was a rebellion against the basics of copyright as applied to software: the idea that the author could absolutely control the distribution and derivations of a protected work. For the movement's founders, the idea was to break the perceived stranglehold of copyright to provide open distribution and use of tested software – and force those who used it to themselves share their own works with the development community. Hence, the software communists tried to create a hook – offer goodies in return for more sharing down the line.

While that was the original intention, what has evolved instead is a veritable Babel of licenses. The Free Software Foundation lists fifty or more. These licenses run the gamut of complexity and applicability. The Berkeley Software Distribution (BSD) license is very liberal, letting the licensee use the particular in just about any way he or she pleases, subject to an "as-is, where-is" warranty. Other licenses tend to be peculiar to the individual developer. Apache, for instance, has a family of similar, permissive, licenses.

The one "free" software that seems to stick out more than any other is the "General Public License". The GPL is the brainchild of Richard Stallman, one of the founders of the Free Software Foundation. Version 2.0 has some peculiar provisions that give earnest lawyers a migraine. While the GPL doesn't restrict the running of the licensed program, section 2 of the license does control the creation of derivative works. Specifically, section 2b pro-

Attorney at Law
David P. Crocker
Solicitor of England and Wales



Business Address
Russell House
158 Pleasant Avenue
Portland, Maine 04103-3204 USA

Phone: 207.879.0708
Fax: 207.221.6417
Email: dpc@davidcrocker.com
www.davidcrocker.com

Open Source

vides that any work that “contains or is derived from the Program” must itself be licensed under the GPL while distribution of “independent and separate works” do not.

Proprietary developers have a natural aversion to disclosing their own code (much to the righteous disgust of Stallman’s acolytes). But it’s that section 2 that’s the issue: what does “contain” or “derived from” really mean, after all and what’s an “independent and separate work”? Well, it all comes down to “linking”. In early programming, the “contain” part of the equation was more easily understood because code was a unitary work that had a beginning and an end and the open source would have been baked into the whole using a “static” link. For coders today, most open source would be hung off the executable using a “dynamic” link. But the Free Software people have, ironically enough, a proprietary interest in how their own license is interpreted. They wouldn’t want just anyone to interpret it, don’t you know?

In FSF’s view of the world, whether GPL open source is baked in or linked depends on the exact nature of the dynamic link – and that means the mode of communication and information communicated through the link. When a developer uses the normal open source API to do the linking (therefore providing “standard” communication), the FSF takes the position that the open source and the larger proprietary application are separate. What developers have done in many cases is to interpose proprietary code (known as a “shim”) between the open source API and other code to provide “special” communication capabilities. This sort of programming is often found in mass market hardware like DVD recorders and the like. The FSF views the use of shims as creating an integrated work subjecting the proprietary code to the GPL license.

First admitted to practice in 1989, David P. Crocker manages an international law practice in licensing, intellectual property, information technology and business law. He has experience in domestic and international licensing, technology development and general intellectual property law, including copyright, trademark and trade secret protection.



Crocker provides direct and timely assistance not only to technologists, but also to writers, artists, photographers and all creative people who wish to protect and license their work.

Crocker is currently qualified to practice in the United States and England and Wales. He advises United States, United Kingdom and other international clients in intellectual property, technology and business matters.

All Content © 2008 David P. Crocker. All Rights Reserved

The FSF’s position creates a dilemma for companies like nVidia and ATI. Stallman in particular is reported annoyed at the “Tivoization” of GPL code by folks who dare to use GPL code without taking their own trousers down. Interestingly, no less a personage than Linus Torvald’s seems unbothered by this because the essential goal of disseminating open source has been met. But this is not Stallman’s view, apparently. He and his minions have released version 3.0 of the GPL which is considerably more prolix and confusing. In the prologue, his primary beef is software patents and imbedded code that can’t be replaced by a user. So, version 3.0 seeks to tamp down patent cross licensing while free up “general purpose computers.”

Needless to say, this can make life interesting for the lawyers. In general, it’s necessary to audit open source use and establish corporate policies. How to warrant software becomes more difficult in an open source world. A vendor would necessarily warrant that the application contains no open source or that there is no open source or reciprocal licenses. Or, as is my practice, I simply exhaustively list all open source with references to the licenses. But one question still remains: in an open source world, is a standard third party IP indemnification provision adequate? That’s a subject for another paper.