

Is OSS for You

Hot Topics in Open-Source Software Licensing

Businesses and software developers who incorporate new or pre-existing open-source software (OSS) code as part of their hardware or software solutions are on the leading edge of a new dynamic in software licensing. Where the historical, proprietary model was based on protection and enforcement of the original developer's ability to control how and by whom the software could be copied or modified, the new framework is based instead on protection and enforcement of subsequent developers' and users' abilities to freely copy, modify and re-distribute the original work. The OSS paradigm therefore presents exciting new opportunities for innovation and business development. However, it is not free from legal considerations and pitfalls, and those interested in using OSS are well advised to at least familiarize themselves with some of the basic concepts of open-source licensing and also with some of the legal issues that have arisen under the new framework.

The Open-Source Model

Open-source licensing seeks to achieve the goal of freely distributable and modifiable software by taking advantage of copyright law – the same tool that is used to restrict downstream distribution and modification under the proprietary framework. Some refer to this open-source re-appropriation of copyright law as “copyleft.” In the copyleft framework:

1. A developer asserts a legal copyright interest in software he or she has authored.
2. The developer then licenses that software under the terms of an open-source license, such as the Free Software Foundation's (FSF's) GNU General Public License (GPL).¹
3. Licensees of that software are generally free to use, modify and re-distribute the software, provided that they also freely distribute the source code and take no steps to prevent downstream users from using, modifying and re-distributing the licensee's derivative work, if any. If the licensees fail to do so, their copying and/or use of the software could be determined to be outside

the scope of the license and an infringement of the original author's copyright.

As discussed in greater detail below, penalties under the Copyright Act for infringement can be sizeable, giving the copyleft framework significant potential power to achieve the goal of many OSS advocates: “promoting and preserving software freedom.”²

Many, if not most, of the more noteworthy current developments and publicized disputes in OSS licensing concern the scope of a downstream user's or developer's obligations when including open-source code in a proprietary product, be that product primarily software or hardware. Different provisions of the GPL (the most popular open-source license in use today)³ define the circumstances under which a proprietary product can incorporate GPL-licensed software. It is essential for downstream developers to have a confident understanding of the ramifications of these provisions prior to using any GPL-licensed software in one of their proprietary products.

Open-Source Software as Part of a Larger Software Solution

The details differ, but the substance of GPLv2 and of GPLv3 is essentially the same with regard to what happens when GPL-licensed software is used as part of a larger software solution. However, neither version provides any bright-line tests, making this aspect of GPL licensing, in many cases, the most difficult to resolve to any degree of certainty. Developers must pay careful attention to the terms of the applicable license and to any additional guidance available from the license's authors.

Both versions of the GPL require that any modified version of GPL-licensed software be distributed, if at all, also under the GPL or under a license that is at least as permissive as the GPL. In defining the conditions under which a developer need not license their resulting software under the GPL, each version sets forth substantively the same, nebulous test:

GPLv2:

If identifiable sections of [the resulting] work are not derived from the [GPL-licensed software], and



can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the [GPL-licensed software], the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the [GPL-licensed software].

In addition, mere aggregation of another work not based on the [GPL-licensed software] with the [GPL-licensed software] (or with a work based on the [GPL-licensed software]) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

GPLv3:

A compilation of [GPL-licensed software] with other separate and independent works, which are not by their nature extensions of the [GPL-licensed software], and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of [GPL-licensed software] in an aggregate does not cause this License to apply to the other parts of the aggregate.

Neither version of the GPL makes any significant effort to define terms like "independent" and "separate" (in GPLv2) or "extensions" and "combined" (in GPLv3). Therefore, a developer's determination of whether he or she can consider the resulting work an "aggregate," and eligible for more restrictive licensing terms, if that is the goal, can be very intricate and fact-intensive.

Guidance on this issue is fairly scarce. There has yet to be a published opinion by a U.S. court discussing the circumstances under which a resulting work can be classified as an "aggregate" under either version of the GPL, and it is possible that, even if or when a court does issue such an opinion, its effect never will reach beyond the bounds of the state where the dispute is centered. The FSF has a number frequently asked questions published on its web site, several of which touch on this issue. In response, the FSF states the following:

Where's the line between two separate programs,

and one program with two parts? This is a legal question, which ultimately judges will decide. We believe that a proper criterion depends both on the mechanism of communication (exec, pipes, rpc, function calls within a shared address space, etc.) and the semantics of the communication (what kinds of information are interchanged).

If the modules are included in the same executable file, they are definitely combined in one program. If modules are designed to run linked together in a shared address space, that almost surely means combining them into one program.

By contrast, pipes, sockets and command-line arguments are communication mechanisms normally used between two separate programs. So when they are used for communication, the modules normally are separate programs. But if the semantics of the communication are intimate enough, exchanging complex internal data structures, that too could be a basis to consider the two parts as combined into a larger program.⁴

While helpful, that explanation is possibly biased in favor of more expansive GPL licensing. It is also far from universally applicable and likely will be subject to change as software architecture evolves over time. In close cases, it makes sense for technical experts to consult with a knowledgeable attorney regarding what license terms may apply to a resulting work.

Open-Source Software as Firmware

Hardware developers face a different set of potential problems when they use OSS as "firmware" – read-only-memory-based operating instructions – for their electronic devices, and on this point, GPLv2 and GPLv3 do differ. Under GPLv2, all that is required for distribution of covered software is the concurrent distribution (or concurrent offer to



distribute) the corresponding source code. For years prior to the release of GPLv3, many device manufacturers were able to comply with this provision and still maintain control over their devices' firmware following distribution simply by requiring any modified firmware to carry a unique digital signature needed for installation. Without the signature, any modified versions of the accompanying source code would be essentially worthless. The most famous (or infamous, depending on your politics) corporate fan of this practice was TiVo Inc., the manufacturer of the popular TiVo digital video recording device, most of the firmware for which was based on OSS.

The FSF considered this to be a problem, and it addressed it in GPLv3 with the inclusion of what some have referred to as the "anti-tivoization" provision. This provision requires a device⁵ manufacturer to distribute both the source code and the installation signatures for any GPLv3-licensed software incorporated in a covered device. Therefore, device manufacturers now must be careful to consider which GPL versions apply to any OSS to be used in their devices and, if GPLv3 is in the mix, whether consumers' ability to modify the device firmware will be consistent with their business plans.

Open-Source Licensing Risks and Enforceability

There is no doubt that copyright infringement, at least in the U.S., can carry substantial penalties, making compliance with all software licensing terms a business necessity. Proven infringement can support a permanent injunction, potentially crippling some developers' whole business models. Courts also may award actual damages and profits resulting from infringement or, at the copyright-holder's election, steep statutory damages of up to \$30,000 per work for non-willful infringement or up to \$150,000 per work for willful infringement.

However, for some time, there has been a debate over whether legal questions related to software licensed under open-licensing terms are all just academic. Some have argued, among other things, that the GPL or other OSS licenses fail to satisfy the conditions many jurisdictions place on so-called "clickwrap" licenses, that they are too vague to support enforcement, that they lack consideration, or that many downstream developers lack sufficient ownership rights to give them the ability to enforce the license terms against alleged infringers.⁶ As mentioned above, many of the arguments raised against the GPL have yet to be tested in court. However, some courts both abroad and, now, in the U.S. have considered the matter and have found OSS licenses to be enforceable, at least in the circumstances with which the courts were presented.

A recent opinion issued by the U.S. Court of Appeals for the Federal Circuit likely will serve as very strong ammunition for OSS advocates in future disputes. In *Jacobsen v. Katzer* (decided August 13, 2008), the court made it clear that when OSS licenses place conditions on the use and

copying of software, the holder of the copyright on the software may pursue a claim for copyright infringement. The court specifically rejected the notion that the software's distribution free-of-charge affected the analysis. It concluded that, if a license is limited in scope and the licensee acts outside the scope of the license, the copyright holder has a claim for infringement along with a claim for breach of contract. The terms of the license at issue – the Perl Artistic License – set conditions on the use and copying of the software, which it found the defendant had breached. According to the court, that breach constituted copyright infringement.⁷

Other recent disputes also point to the future viability of the GPL and other OSS licenses. In September 2006, a German civil court found that D-Link Germany GmbH violated the copyright in the Linux kernel and other GPL-licensed software when it distributed that software as firmware without including the accompanying source code.⁸ In addition, several lawsuits filed in the U.S. District Court for the Southern District of New York have resulted in out-of-court settlements and undisclosed payments made to the developers of BusyBox, an OSS program popular as firmware among a variety of device manufacturers.⁹

Open-Source Software Asset Management

The unique legal considerations inherent in the deployment of OSS present a correspondingly unique set of considerations for software asset management (SAM) professionals. With proprietary software, most of the SAM risks for businesses center on overdeployment – the use or installation of more instances of a product than are allowed under the licensing rights acquired by the business. However, with proprietary solutions there is also often minimal risk associated with the publisher's or vendor's right to distribute the software to the licensee business. With OSS, that dynamic frequently is reversed. Overdeployment still may be a concern, especially where the OSS component of a licensed solution is combined with proprietary elements governed by more restrictive licensing terms. However, overdeployment often is less of a concern than are the publisher's or vendor's prior rights and any exposure that the licensee business may be assuming in placing enterprise-level reliance on a product that may be nothing more than a claim of infringement waiting to be asserted by the lawful copyright or patent owner. Businesses deploying OSS therefore must be prepared also to deploy an appropriate level of flexibility in their SAM policies and procedures to mitigate a more dynamic set of licensing risks.

An increasing number of businesses are beginning to offer innovative SAM solutions to OSS licensees. Some, such as OpenLogic,¹⁰ offer comprehensive SAM solutions on a subscription basis – in return for a monthly fee, the provider gives the subscriber access to a library of certified, enterprise-appropriate OSS products, which the provider regularly updates with new releases and/or software patches. These providers also offer OSS network discovery tools and



indemnification against the legal risks that may be associated with deployment of OSS in their libraries of certified products. In addition, insurance providers increasingly are offering coverage for the deployment of OSS, both as productivity tools (operating systems, databases, office suites, etc.) as well as integrated components of the insured business' larger software solutions. The variety of products and services being offered to OSS licensees appears to be increasing in proportion to the ever-expanding number of new enterprise-level OSS solutions now under development.

Conclusion

This paper is only intended to be a brief review of some of the more notable characteristics and pitfalls related to OSS licensing. Any developer contemplating the use of open-source code in his or her end product is well advised to carefully consider whether the low cost and other benefits of open source software are sufficient to overcome any disadvantages that such use may entail. For some businesses, those costs may be negligible or non-existent. Others, though, may want to retain a degree of control over their products that is inconsistent with applicable license terms. Where there is any doubt regarding how to proceed, always consult with an attorney knowledgeable in copyright, licensing and open-source issues.

¹Available at: <http://www.fsf.org/licenses/licenses/gpl.html>.

²See "Our Core Work" at: <http://www.fsf.org/about/what-is-free-software>.

³On June 29, 2007, the FSF released version 3 of the GPL (GPLv3). However, many developers prefer and continue to use version 2 (GPLv2). The majority of open-source software available today is licensed under one, or a combination, or a derivation of the two GPL versions. There are "competitors" to the GPL, however, a notable example of which is the Artistic License published by the Perl Foundation, available at: <http://www.opensource.org/licenses/artistic-license-2.0.php>

⁴See "What is the difference between an 'aggregate' and other kinds of 'modified versions'?" at: <http://www.fsf.org/licenses/licenses/gpl-faq.html#MereAggregation>.

⁵The anti-tivoization provision of GPLv3 applies specifically to any "User Product," which it defines as "any tangible personal property which is normally used for personal, family, or household purposes" or "anything designed or sold for incorporation into a dwelling." Therefore, voting machines, medical appliances, and other devices where access to the firmware could cause safety or security concerns do not fall within its scope.

⁶See generally, Wacha, Jason B., Taking the Case: Is the GPL Enforceable?, available at: <http://www.open-bar.org/docs/GPL-enforceability.pdf>.

⁷The text of the Federal Circuit's *Katzer* opinion is available at: <http://www.cafc.uscourts.gov/opinions/08-1001.pdf>

⁸See "gpl-violations.org project prevails in court case on GPL violation by D-Link," at: http://gpl-violations.org/news/20060922-dlink-judgement_frankfurt.html

⁹Among others, see cases filed against Monsoon Multimedia Inc. (Case 07-CV-8205), Xterasys (Case 07-CV-10456), High-Gain Antennas (Case 07-CV-10455), Bell Microproducts, Inc. (Case 08-CV-5270), Super Micro Computer, Inc. (Case 08-CV-5269), and Extreme Networks, Inc. (08-CV-6426).

¹⁰<http://www.openlogic.com/>

Robert J. Scott
Managing Partner
Scott & Scott, LLP