

An Innovation in Contract Drafting

Open Source

BY CHRISTOPHER J. FALKOWSKI

“Open source” licensing is an innovation in the contractual framework for licensing software. Open source licensing can also be described as a business model or technology development methodology. Some proponents even consider it part of a broader social movement.¹

What is open source licensing?

Open source licenses make “source” code available to the public. Source code consists of computer instructions in a format that can be understood and updated by human programmers. Source code is compiled by a computer to create “object” code, which is the code format actually “run” on the computer to achieve the functionality of the computer program. Access to source code is necessary to properly modify or even understand a computer program.

In an open source framework, source code is available or open to all potentially interested users. In exchange for access to source code, a user accepts contractual conditions relating to the use and distribution of source code. Most open source licenses specifically provide that use of the computer program constitutes acceptance of the terms in the license.²

By providing source code access to any user willing to agree to the terms of the open source license, collaborative development activities can be facilitated between large numbers of disbursed individuals and organiza-

tions. Parties without knowledge of each other can nonetheless collaborate, allowing each programmer to benefit from the work of many other programmers.

There are many different types of open source licenses. In evaluating a particular open source license, it is important to focus on the specific terms of the applicable license. The range of licensee obligations can vary widely. Specific examples of open source licenses can be found on numerous websites, such as www.opensource.org or www.gnu.org.

Some open source licenses impose few restrictions on the ability of users to behave in a proprietary or “closed” source manner. For example, the primary provision in the Berkley Software Distribution (BSD) license is a liability limitation protecting source code contributors. The terms of the BSD license provide that the code is licensed on an “AS IS” basis.³ The BSD license does not prevent the user from charging license fees to other users.

Most open source licenses prohibit users from incorporating open source code elements into a software product that is then licensed to third parties on a proprietary basis. One common type of open source license is the General Public License (GPL). Linux users are subject to the GPL license. Section 2(b) of the GPL⁴ provides that “[y]ou must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof,

to be licensed as a whole at no charge to all third parties under the terms of this License.”

The GPL framework is often referred to as a “copyleft” license because the copyrights of the program creator and subsequent contributors are used to preclude the ability of subsequent users from asserting copyrights against users even further downstream. Under the “copyleft” concept, each user who redistributes the software must pass along to subsequent users the same freedom to copy and change the software enjoyed by the distributing user. The “copyleft” framework creates lengthy chains of contractual relationships in which each new contractual relationship is bound by the terms earlier in the chain. The original creator of a “copyleft” computer program defines the license terms. All subsequent contributors and users are bound to those initial terms. This is true even though the relative value of the original code contribution will diminish over time in relationship to the value of subsequent improvements and derivative works.

The Lesser General Public License (LGPL)⁵ is a derivation of GPL in which “less” is done to “protect the user’s freedom.” In contrast to the GPL, the LGPL “permits more lax criteria for linking other code with the library” so that an open source component can link with proprietary components.

The BSD, GPL, and LGPL licenses are classic examples of open source licenses. There are different versions of these licenses,

and new types of licenses are being created. The Open Source Institute provides a list of “approved” open source licenses at <http://www.opensource.org/licenses/>.

Is open source software free?

The phrase “free software” in the context of open source relates to the freedom of users to access and improve source code, not the prices charged by users.⁶ In a BSD license, license fees can be charged. Most other frameworks such as GPL specifically provide a “no charge” clause with respect to license fees.⁷ Other types of charges are specifically permitted. Section 1 of the GPL at <http://www.gnu.org/licenses/gpl.txt> provides that “[y]ou may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.” Furthermore, open source licenses do not prohibit the selling of services related to open source software.

What are the advantages of open source licensing?

Many of the advantages can be best summarized as a derivative of the adage that “two heads are better than one.”

PEER REVIEW

By facilitating collaborative efforts and access to source code, software can be subjected to peer review. In a proprietary model, the inability to access source code and modify the proprietary products of another hinder the peer review process. Open source software development allows the entire world to serve as bug identifiers and bug fixers.

COST SAVINGS

Open source software is typically less expensive than proprietary software because there is no license fee. A corresponding benefit of decreased costs is a larger user base, which further reduces per-unit costs for software. Many governmental entities around the world are committed to open source software development to facilitate inexpensive distribution of software. Although open source software is not free because users must pay for related services, competition between service providers helps keep prices down.

STANDARDIZATION AND ENHANCED INTEROPERABILITY

Open source software development can encourage the development of common standards that enhance software interoperability. The benefits of common standards and enhanced interoperability are significant. Per-unit costs can be reduced while quality is improved. The creation of larger and more competitive markets can further support the demand for new products and services.

What are the risks and challenges raised by open source software?

The advantages of open source software are intrinsically related to certain risks.

THE CHAIN-OF-TITLE “TRAIN WRECK” SCENARIO

Open source frameworks such as GPL can link thousands of users together in complex chains of contractual relationships. Intellectual property disputes relating to a single user or single copy of source code can easily spread upstream and downstream to other users and contributors. Chain-of-title problems occur in other areas of law such as

real estate transactions, but the tangible nature of those transactions limits the dispute to a finite number of parties. With respect to open source software, a product specifically designed to be copied for free and distributed widely, a chain-of-title dispute can negatively impact the entire user community for the software.

A prominent real world example of a potential train wreck scenario is the current litigation over Linux. In 2002, SCO filed a lawsuit against IBM, contending IBM misappropriated SCO’s property in contributing UNIX code components to the Linux operating system. See *The SCO Group, Inc v IBM*, Case Number 2:03cv0294 (D Utah 2002).⁸ SCO asserts that it, and not IBM, holds title to a substantial body of code used to create Linux. In response to the IBM lawsuit, Novell issued a press release⁹ asserting that Novell retained title to the copyrights asserted by SCO. Novell was then sued for defamation of title. See *The SCO Group, Inc v Novell*, Case Number 040900936 (3rd Judicial District of Utah 2004).¹⁰ Both cases are currently pending.¹¹

If SCO prevails against IBM and Novell, every user of the increasingly popular Linux operating system may be liable to SCO for copyright infringement if they are not customers of SCO, and for breach of contract if they are customers of SCO. Such an outcome appears unlikely, but it would be a devastating blow to Linux users and proponents of open source software licensing. SCO has already filed lawsuits against mere end users such as AutoZone and DaimlerChrysler.¹²

FORFEITURE THROUGH “CONTAMINATION”

The commercial value of proprietary code components may be forfeited if they are inadvertently combined with open source components. Under the GPL, a combination of open source and proprietary elements is subjected to the terms of the open source license, if the combined code is licensed or otherwise distributed to a third party.

Section 2(b) of GPL provides that: “[y]ou must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge

Fast Facts

In an open source framework, source code is available or open to all potentially interested users.

By providing source code access to any user willing to agree to the terms of the open source license, collaborative development activities can be facilitated between large numbers of disbursed individuals and organizations.

With respect to open source software, a product specifically designed to be copied for free and distributed widely, a chain-of-title dispute can negatively impact the entire user community for the software.

to all third parties under the terms of this License.” Under the terms of Section 2(b), an accidental commingling of proprietary and open source components (or its derivatives) results in the inability of a licensor to distribute the code combination in a proprietary manner. While this may not be a significant concern for end users, it could have a dramatic impact on software vendors. Even the inclusion of a relatively minor or insignificant open source code component could “contaminate” a large system of proprietary applications. The negative effect of an accidental forfeiture is only worsened by the likelihood that the accidental commingling would only be identified significantly after the distribution of the contaminated code, when it would be too late to avoid forfeiture.

To avoid the risk of forfeiture, proprietary licensors may require “no contamination” warranties from licensees providing code components to proprietary licensors. The risk of contamination may force users to either remain fully in the proprietary software world, or to fully embrace the open source model. Another option for end users is to simply refrain from distributing or sublicensing code components.

The forfeiture issue was raised defensively in *Computer Associates Intl v Quest Software Inc*, Case Number 1:02cv04721 (ND Ill Aug 3, 2004). In response to a claim of copyright infringement by Computer Associates (CA), Quest argued that CA’s code was derived from GPL code, and was thus subject to the GPL license. CA prevailed on a factual determination that the proprietary code was not a derivative work of the open source code components.

INTELLECTUAL PROPERTY LIABILITY

In a proprietary transaction, it is common for the software vendor to provide some type of indemnity or warranty against the infringement of third party intellectual property rights. In an open source transaction, vendors are typically reluctant to make such commitments because the vendor cannot vouch for all of the different code contributions. The Linux indemnification plans offered by Novell and HP highlight this reluctance.¹³ Thus, so long as the proprietary licensor has the resources to support the war-

IT decision makers need to be cognizant of their particular cost-benefit tradeoffs in order to navigate a highly dynamic environment lacking clear and binding judicial precedent.

ranty or indemnity, an open source licensee will often risk greater intellectual property liability than a proprietary licensee.

The enhanced risk is particularly acute with respect to patents. The “copyleft” approach is grounded in copyright law, not patent law. Copyright protection is limited to the actual expression in a computer program authored by a programmer. A patent can cover multiple embodiments or variations of software before they are actually created. It is easier to accidentally infringe a patent than a copyright. Furthermore, issuance of software patents did not come into prominence until the mid-1990s.¹⁴ Some open source licenses fail to even mention the word “patent.” Thus, it is possible that under some licenses, users can benefit from the sharing of code components without giving up their right to assert patent claims against those contributors. Such an outcome goes against the spirit of the open source transaction, but it may not violate the terms of those licenses.

The Public Patent Foundation recently identified 283 issued patents potentially infringed by Linux.¹⁵ New open source agreements directly address the issue of patent rights,¹⁶ but licensees are limited by the terms of the license. This challenge may be mitigated somewhat if other companies follow the example of IBM, which recently dedicated 500 patents to open source developers.¹⁷

ENFORCEMENT ISSUES

Many issues relating to enforcement of open source licenses need to be considered in any risk analysis.

Binding users to the terms

Users are bound to the terms of the open source license by viewing the terms within

the source code itself. What happens if the user does not read the license? What happens if the proper license terms are deleted, and the source code is then distributed to unsuspecting users downstream? A failure of even one user to be bound to the terms of the license can result in a different variation of the “train wreck” scenario discussed above.

These issues have not been tested in the specific context of open source licensing. The enforceability of open source licenses may depend on whether there must be actual notice of contract terms. For example, anyone working in the IT industry has reason to know that Linux is an open source product subject to some type of GPL license. Existing case law is not necessarily definitive on the question of notice.¹⁸

Standing to sue

The question of who has standing to sue a non-complying user bound to an open source license is potentially subject to dispute. Many open source licenses specifically remove the ability of one user to sue another. Section 6 of the GPL provides that “[y]ou are not responsible for enforcing compliance by third parties to this License.” It is not clear that a user has the right to sue another user for breach of an open source license, since there may not be privity of contract between the two parties. Moreover, only the owner or exclusive licensor of a copyright can bring suit for copyright infringement. Enforcement of open source licenses will present interesting standing issues in the future.

Conclusion

Open source licensing is an exciting contractual innovation that promises many

advantages to the IT community, and to the public at large. It also presents unique and largely untested risks. IT decision makers need to be cognizant of their particular cost-benefit tradeoffs in order to navigate a highly dynamic environment lacking clear and binding judicial precedent. ♦



Chris Falkowski is a partner in the intellectual property group at the law firm of Honigman Miller Schartz and Cohn LLP. Mr. Falkowski's practice focuses on technology-related legal issues, including software patents and licens-

ing transactions. He is a council member for the Computer Law Section and a former editor of the Michigan Computer Lawyer. Mr. Falkowski currently authors a blawg at www.ITblawg.com.

Footnotes

1. See http://www.bollier.org/pdf/Georgetown_remarks_%20Oct2002.pdf; http://www.asiaosc.org/article_46.html; <http://siliconvalley.internet.com/news/article.php/2246751>; <http://demandmedia.net/story/2004/8/18/163422/223>; <http://www.elearnspace.org/blog/archives/001087.html>; and http://news.com.com/Open+source+under+the+microscope/2008-7344_3-5133553.html.
2. For an example of such a provision, see Section 5 of the GPL license at <http://www.gnu.org/licenses/gpl.txt>.
3. <http://www.opensource.org/licenses/bsd-license.php>.
4. <http://www.gnu.org/licenses/gpl.txt>.
5. <http://www.gnu.org/licenses/lgpl.txt>.
6. <http://www.gnu.org/licenses/gpl.txt>.
7. See Section 2(b) of the GPL license at <http://www.gnu.org/licenses/gpl.txt>.
8. Formerly available at <http://www.sco.com/ibmlawsuit/amendedcomplaintjune16.html>.
9. <http://www.novell.com/news/press/archive/2003/06/pr03036.html>.
10. Formerly available at http://www.sco.com/novell/complaint_filing_jan_20_2004.pdf.
11. Recently filed pleadings, formerly available at <http://www.sco.com/ibmlawsuit/> and <http://www.sco.com/novell/>.
12. Formerly available at <http://www.sco.com/autozone/> and <http://www.sco.com/daimlerchrysler/>.
13. See <https://h30201.www3.hp.com/Default.asp> (to qualify for the indemnity, the end user can only use HP hardware and the end user is not permitted to access, much less modify, the source code); <http://www.novell.com/collateral/4613395/4613395.pdf> (minimum annual purchase requirement of \$50,000 and liability capped at the value of annual purchases).
14. Christopher J. Falkowski, "Software-Related Patents: Risk and Opportunities," *Michigan Lawyers Weekly*, October 11, 2004, at 4.
15. http://news.com.com/Group:+Linux+potentially+infringes+283+patents/2100-7344_3-5291403.html.
16. See Sections 7 and 8 of the GPL at <http://www.gnu.org/licenses/gpl.txt>.
17. <http://internetnews.com/dev-news/article.php/3457381>.
18. See *Ticketmaster Corp v Tickets.com, Inc*, 2003 US Dist LEXIS 6483 (CD Cal March 7, 2003) (opinion differentiates between contractual terms that required user scrolling and terms placed so that the defendant's executive admitted that "they could not be missed") and *Register.com, Inc v Verio, Inc*, 126 F Supp 2d 238 (SD NY 2000) (breach of contract found where defendant "does not argue that it was unaware" of contractual terms posted on a website's term of use).

