

Open Source – Friend or Foe?

Software is often derived from a number of sources. Over the last decade, the use of something called “open source” has gained popularity. However, without a clear understanding of what open source is and the various rights and restrictions associated with its usage, it is possible to unintentionally take on additional risk and in some cases give up valuable intellectual property rights in software developed with open source.

Consider Robert, the owner of a local inventory-consulting firm, who enters your office seeking counsel. Robert is very excited about an innovative new automated inventory ordering software program his firm has just finished developing. He is anxious to beta test the program with several of his clients, but thought it best to consult you first to make sure he has covered his bases.

Robert explains how his firm, through tightly managing his development budget and savvy cost-saving initiatives, was able to (a) take advantage of his firm’s existing software as a foundation for the new software; (b) obtain several of the new software’s components through shrewd negotiations with third-party licenses; (c) retain the cheap labor of college students for basic coding; and (d) pick up several “free” applications off the Internet that were easily integrated into the new software.

Although Robert is quite impressed with his success thus far, your number-one job is to let him know he must put the brakes on his project or potentially risk losing all rights to his new software and jeopardize his investment thus far.

Delicately, you tell Robert how wise he was to come see you, and that, in addition to the typical due diligence necessary to evaluate all developers’ contributions to the new software (which is beyond the scope of this article) and the potential risks they present to the software’s life cycle, the “free” applications found on

Free software, or more commonly referred to as “open source” software, may come without an actual licensing or distribution price tag, but it can be far from free.

the Internet and integrated into his firm’s new software will require specific attention.

You explain to Robert that the term “free” is only meant to relate to actual price to use the downloaded software. Free software, or more commonly referred to as “open source” software, may come without an actual licensing or distribution price tag, but it can be far from free. The misapplication of open source software can inadvertently place the new software in the public domain and expose Robert to claims from the open source authors for violations of the underlying license terms. At this juncture, you deem it a valuable education to share with Robert a bit of the history, benefits, and challenges commercial businesses encounter when integrating open source software into their products.

Robert asks, “What exactly is open source?” You explain that open source licensing refers to making the source



Kevin Gard
Contributing Author

Mr. Gard is assistant general counsel of Sun Microsystems, Inc., and practices technology licensing and intellectual property law in Itasca, Illinois.



Jen Salyers
Contributing Author

Ms. Salyers is a partner in the firm of Bon-nabeau, Salyers, Stites, Doe, Andresen in Minneapolis and concentrates her practice in the representation of corporations acquiring information technology products and services.

code for a software program openly available at no charge for the public to work with it. Typically, a community of developers works with the source code, creating enhancements and extensions and improvements, which are in turn freely shared back with the community and further enhanced, extended, and improved.¹ “Open” means open access to

source code and peer-review by an open community of developers. Robert may think this is describing his firm's use, but there's more. You explain that around 1989, the free or open software movement began as a reaction to proprietary (closed source) software that was available for a fee and only in object code (machine readable version) form. As a means of nurturing the technical community's desire to "see how it works" and "make it better," the open source movement began. About the same time, the Open Source Initiative (OSI)² was organized to promote a pragmatic approach to commercial use of open source software. The OSI has established criteria for a software license to qualify as an open source license based upon conditions that ensure the software intended to be governed by the OSI open source license maintains the original principles of open use by the community.³

Robert interjects that he's not so sure about this "open access to all" concept, but he does know that his development

time has been greatly reduced by utilizing open source code, that, from what he can tell, several respected developers contributed to the open source's development giving him confidence of its integrity and adherence to industry standards, and that the bottom line is the huge cost saving of not having to reinvent the wheel.⁴ All of Robert's asserted benefits of utilizing open source code in a commercial product are accurate; however, he must also acknowledge the challenges of using open source.

It is very important for Robert to weigh the potential benefits against the potential risks and obligations relative to integrating open source software into his firm's new software. A few risks include:

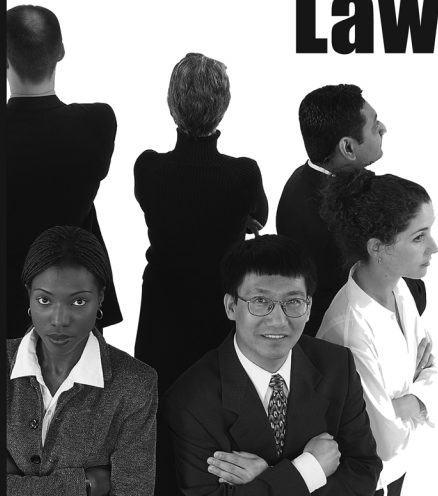
No intellectual property protection. Typically open source software is offered without intellectual property indemnity or any warranty language of any kind. Accordingly, if ever Robert receives a claim alleging the open

source code infringes upon another's intellectual property, his firm is left holding the bag. In addition, several open source licenses (e.g., MPL or IBM-CPL) revoke the imbedded open source license granted in the event the user/distributor (like Robert's firm) brings patent infringement action against the grantor. Therefore, it is imperative for Robert to attempt to perform his due diligence in evaluating the infringement potential. Other newer open source licenses (CDDL) actually do provide patent protection.

Encumbrances. In some cases, the open source licenses may require the modified or linked code to be cross-licensed or contributed back to the community. The General Public License (GPL) is an open source license that does just this. The GPL requires the user to cause any work that it distributes or publishes, that in whole or in part contains or is derived from the GPL licensed program, or



Hot Topics in Employment Law Litigation



TOPICS TO BE COVERED:

- Cutting-edge issues in sexual and racial harassment cases.
- Handling common law and statutory whistle-blower claims in a changing legal landscape.
- National-origin discrimination and harassment in a post-9/11 world.
- Wage and hour claims in the shadow of new regulations.
- Employment claims under federal law versus state law: A comparative analysis of legal strategy and litigation tactics.

FEATURING:

- **Justin Cummins** (course chair), Miller-O'Brian, P.L.L.P.
- **Kristy Albrecht**, Dorsey & Whitney, L.L.P.
- **Clayton Halunen**, Halunen & Associates
- **Erica Jacobson**, Minnesota Attorney General's Office
- **John Klassen**, John A. Klassen, P.A.
- **Paul Lukas**, Nichols, Kaster & Anderson, P.L.L.P.
- **Leslie Lienemann**, Culberth & Lienemann, L.L.P.
- **Holly Robbins**, Faegre & Benson, L.L.P.
- **Joseph Schmitt**, Halleland, Lewis, Nilan & Johnson, P. A.
- **Joseph Sokolowski**, Fredrickson & Byron, P.A.
- **Amy Taber**, Rider Bennett, L.L.P.
- **Laurie Vasichek**, EEOC

Thursday, January 25

8:45 a.m. to 4:30 p.m. at the HCBA

6.25 CLE credits applied for.

Register Today: Call 612-752-6600 • Sign up online at www.hcba.org • Use The Docket form at the back of this magazine.

HCBA member \$160 Non-HCBA member \$175 CLE Card holders: No charge*

any part thereof, to be licensed as a whole at no charge to all comers under the same terms as the original GPL license. The goal of the GPL license is to prevent anyone from taking the original open source or any derivative thereof and making it private.⁵ This is known as a “copyleft” provision.

Little may be known about the author or contributors. Did they originate the code, or was it wrongly lifted from others' code?

Violations of the GPL can include substantial monetary damages and injunctions that could stop shipment or mandate the release of code. While the copyright to the modifications to the GPL will be held by the party creating the modifications, once distributed, the normal ownership rights are degraded significantly by the requirements of the GPL. For example, if distributed, the party must also provide the source code to the software and allow for copying, modification, and redistribution, all without compensation to the author. Strategically, if a developer can separate identifiable sections of his or her work that are not derived from the open source program, and they can be reasonably considered independent and separate works in themselves, the GPL license terms would not apply to that code. Therefore, on the same distributed disk, there may be portions of the new software proprietarily protected by Robert's end user license, and other portions rightfully governed by the GPL license.

Restrictive or ambiguous license terms. Some open source licenses are poorly worded, so the terms are restrictive or ambiguous, thus raising serious interpretation issues. This is common when a developer tries to create his or her own open source license crudely based upon an existing approved OSI license and does not rely on existing OSI-certified open

source licenses or OSI certification of the newly created version.

Support costs. Although the open source code may seem to be “free” initially because there is no license fee or royalty attached to use, there may be downstream support costs that must be considered. Is Robert's firm capable of addressing the support needs of customers if issues arise? Are support resources available via the original open source developers? This could be either a hidden revenue generator for Robert or a major problem depending upon the availability of support resources for the open source.

Unclear ownership. The origin and ownership of the open source code may be unclear. Little may be known about the author or contributors. Did they originate the code, or was it wrongly lifted from others' code? Given the lack of intellectual property indemnification, without a clear chain of authorship, Robert's firm runs the risk of undisclosed authors later appearing and threatening to halt distribution and seeking damages for wrongful use.

Cumbersome attribution requirements. The open source license may include cumbersome notice and attribution requirements that must be followed and will require Robert's firm to clearly spell out in either header files or introductory screens of his firm's new software the origins and attributes of the open source code. This gives credit where credit is due, but can be cumbersome and distracting when presented in a commercial product.

Impact on current or future business objectives or strategy. The impact on Robert's firm's current and future business strategies must be considered. Robert needs to make sure that a short-term solution to development schedule pressures does

not become a long-term headache for his firm. Robert's use of open source may also raise concerns for customers considering the use of Robert's product. While Robert's firm's use of open source may not be surprising to a licensee, savvy customers will expect Robert's firm to stand behind the open source in the same way it stands behind the rest of the product, with the same warranties, indemnification from third-party intellectual property claims, and maintenance and support. Furthermore, if Robert's firm's product is one that can be integrated with any of its customers' software, this could also raise concerns for Robert's customers. Those customers will want to know whether the use of the open source in Robert's firm's product with the customer's software could jeopardize any intellectual property rights the customer has in its software. If so, Robert may find that customers are reluctant to purchase his firm's product.

Any of these issues coming to fruition

...watchdog groups and courts are becoming more active in enforcing open source licenses in the commercial arena.

could result in negative publicity and lawsuits for Robert's firm. In addition, watchdog groups and courts are becoming more active in enforcing open source licenses in the commercial arena. In April 2005, a Munich court ruled that a British software company must stop selling and distributing products that incorporate open source software unless those products comply with the terms of the incorporated open source license (GNU General Public License).⁶

Common OSI-approved open source licenses include the BSD, GPL, LGPL, Apache, MIT, MPL, and CDDL. OSI maintains a list of open source licenses that conform to the open source definition, have been through public scrutiny, and have received its approval. These


open source licenses run the gamut regarding their integration friendliness in commercial products, but to their credit do give the reader familiarity and comfort relative to the consistency of their identical content—good or bad. Before committing to using any license as part of the new software, each component's license should be carefully reviewed. However, many open source licenses are not OSI-approved, may be derivatives of certified open source licenses, or may be simply constructed piecemeal by the author/developer, thus requiring a scrutinizing eye by the reviewer. Furthermore, use of non-OSI-approved licenses in commercial products may lead to additional concerns from customers who are unfamiliar with the license terms. Customers considering purchasing such products need to understand if incorporating the product into their environment could impact the intellectual property rights in anything integrated. For a stand-alone commercial product this is less of a concern.

So, how does Robert protect his investment, but not let go of the benefits his new software has received from the integration of the open source code? To limit the risk, Robert should adopt a best-practices approach comprised of appropriate management controls regarding use of open source; education of employees regarding the unique aspects of open source; and enforcement compliance with these controls and policies.⁷ Such best practices should include:

- **Due diligence.** Closely review all open source licenses utilized to identify the contents and requirements of each license, their authors' identities, and the likelihood of exposure to liability and license requirements.
- **Bifurcation of code.** If necessary, and if technically feasible, proprietary code not derived from open source should be separated from the actual open source itself. This limits the applicability of the open source licenses to the proprietary code (and avoids GPL copyright issues).

- **Notices.** Make sure all notices and attributions have been integrated as instructed within the applicable open source licenses.
- **Open source technical support.** Confirm a support mechanism is in place in the event any error arises relative to the open source components. This may be a revenue-generating opportunity. If a firm like Robert's is not able to handle the support service, maybe the original authors provide such a service.
- **EUL language clarifiers.** Integrate disclaimers in the end user license (EUL) relative to open source components relating to intellectual property indemnification and no warranties.
- **Inclusion of open source licenses.** Provide the related open source licenses in "readme.txt" files or introductory screens as directed by the licenses governing the open source code.
- **Service provider alternative.** Alternatively, consider open sourcing the new software entirely. Revise the business model to provide implementation services and code maintenance to end users as a means of generating revenue.
- **Reinventing the wheel.** As a means of avoiding the open source issue entirely, consider utilizing alternative third-party code components or creating them in-house. Yes, an additional expense, but such a course of action does minimize risk and provide peace of mind by not having to invest in an open source compliance program.

Any use of open source should be carefully evaluated. While the use of open source code can have significant cost- and time-savings benefits for the developer, its usage, if not properly managed, can also have a serious impact on the ownership and marketability of the resulting product, and may also impact the rights of the end user. Although this

article's primary focus is on developers' use of open source, end users of products containing open source should also consider what risks they may be assuming and if it is appropriate for them to assume those risks. 

¹ Christian H. Nadan, *Open Source Licensing: Virus or Virtue?* 10 Tex. Intell. Prop. L. J. 349, 352 (2002).

² www.opensource.org/docs/definition.php.

³ Open Source Initiative, The Open Source Definition at www.opensource.org/docs/definition.html (visited March 29, 2006). Definition: (a) The license shall not restrict any party from selling or giving away the software as a component of an aggregate software distribution containing programs from several different sources. The license shall not require a royalty or other fee for such sale; (b) The program must include source code, and must allow distribution in source code as well as compiled form; (c) The license must allow modifications and derived works, and must allow them to be distributed under the same terms as the license of the original software; (d) The license may restrict source-code from being distributed in modified form only if the license allows the distribution of "patch files" with the source code for the purpose of modifying the program at build time. The license must explicitly permit distribution of software built from modified source code. The license may require derived works to carry a different name or version number from the original software; (e) The license must not discriminate against persons or groups; (f) The license must not restrict anyone from making use of the program in a specific field of endeavor; (g) The rights attached to the program must apply to all to whom the program is redistributed without the need for execution of an additional license by those parties; (h) The rights attached to the program must not depend on the program's being part of a particular software distribution. If the program is extracted from that distribution and used or distributed within the terms of the program's license, all parties to whom the program is redistributed should have the same rights as those that are granted in conjunction with the original software distribution; (i) The license must not place restrictions on other software that is distributed along with the licensed software; and (j) No provision of the license may be predicated on any individual technology or style of interface.

⁴ "Open Source Initiative, Advocacy, the Open Source Case for Business," at www.opensource.org/advocacy/case_for_business.html (visited March 30, 2006). The foundation of the business case for open source is high reliability. Open source software is peer-reviewed software, proofed and tested by a whole community of developers. By leveraging a community of open source developers, a software program can be written more quickly and more cheaply—in effect one can outsource some of the work to the open source community, which does not charge for its services.

⁵ Laura A. Majerus, *Court Evaluates Meaning of "Derivative Work" in an Open Source License*, at <http://library.findlaw.com/2003/Jun/16/132811.html> (visited March 29, 2006).

⁶ *Welte v. Forinet UK Limited*, Landgericht Muenchen I, No. 21 0 7240/05, 4/12/05.

⁷ Open Source's Pandora's Box, at www.optimizemag.com/issue/027/law.htm (visited March 29, 2006).