2nd Annual
MIDDLE EAST
Legal Perspectives for U.S. Investors
April 22 • Harvard Club • NY

Law.com's In-House Counsel

Select '**Print**' in your browser menu to print this document.

Page printed from: http://www.inhousecounsel.com

Back to Article

Caution: Open Source Is Far From Free

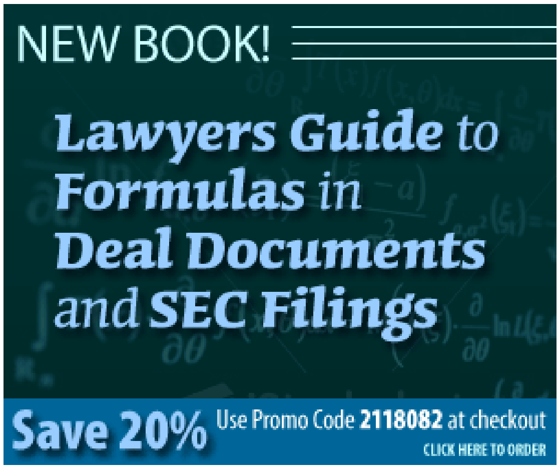
William H. Venema
The National Law Journal
October 21, 2008

In his highly acclaimed best seller, *The World is Flat*, Thomas Friedman hails open-sourcing as one of 10 "flatteners" of the world. Nevertheless, many enterprises have included open-source code in their proprietary software without fully understanding the risks.

Open-source software is software for which the source code is freely and publicly available. Although open-source software is also called "free software" (see www.fsf.org and www.opensource.org), it is far from free. There are specific licensing agreements that are applicable to such code, which specify the responsibilities of the user of the code.

Determining what restrictions open-source licenses impose on the use of particular open-source code is no easy task. As of March, the Open Source Initiative had approved 68 different open-source licenses. To complicate matters further, each of the 68 licenses is different from the others. In addition to the licenses approved by OSI, there are hundreds of other versions of software licenses, from the idiosyncratic licenses created by particular developers to variations on the licenses approved by OSI. Finally, if developers combine open-source code from several sources, along with their proprietary code, then they create a complex mix of intellectual property that is virtually impossible to decipher and may include conflicts that prevent its distribution at all. For example, software components licensed under the widely used Gnu General Public License may not be distributed with software components licensed under the Mozilla Public License.

Despite the numerous open-source licenses that are in existence, there are some that dominate the field. OpenLogic, an organization that provides enterprises with a certified library of open-source software, reported on the licenses that are used by the top 20 packages in its library: 20 percent used GPL or Gnu Library General Public License; 75 percent used Apache (Apache Software Foundation); and another 20 percent used Common Public License, Perl, Eclipse and BSD (a software license that was developed by the University of California, Berkeley). These percentages total to more than 100 percent because several of the packages involve more than one license.


NEW BOOK!
Lawyers Guide to Formulas in Deal Documents and SEC Filings
Save 20%
Use Promo Code 2118082 at checkout
CLICK HERE TO ORDER

BECOMING KNOWLEDGEABLE ABOUT THE DANGERS WITHIN

Many executives whose enterprises routinely use open-source software fail to understand that any violation of the applicable open-source license is a violation of copyright law, which can give rise to liability for significant damages. Incorporating open-source code into proprietary software is especially problematical because a common goal of many of the open-source licenses is to preserve "openness" by requiring that any modified software be redistributed on the same terms as the open-source software and at no charge. Thus, if a programmer includes, in the proprietary software of an enterprise, any open-source code that requires the modified software to be redistributed at no charge, then any later distribution of the proprietary software must be free. Obviously, the commercial value of that software is thereby virtually eliminated.

The adverse effects of failing to comply with the applicable open-source licenses can happen without warning because many of them occur automatically. Several of the most frequently used open-source licenses are "conditional licenses," which means they do not include standard notice-of-breach and cure provisions. Therefore, if a user fails to comply with the terms of the license, the license terminates automatically. Because the open-source code is protected by copyright, the license is the only source of rights to use the software. If the license terminates, then any continued use of the software constitutes copyright infringement. Public companies, in particular, cannot ignore such infringement and hope they avoid detection because the Sarbanes-Oxley Act and Financial Accounting and Standards Board Statement 142 require them to value their software and assess their litigation risks.

The adverse effects of failing to manage the use of open-source code can extend beyond copyright infringement damages. For example, when IBM acquired Think Dynamics a few years ago, an examination of its software code found 80 to 100 examples of open-source code that the company had failed to account for properly. As a result, IBM reduced the purchase price for the company from $67 million to $46 million. Discovering unaccounted-for open-source code in an enterprise's software during the due diligence process could also upset a potential joint venture, an investment in the enterprise or other strategic transaction.

The dangers inherent in open-source code are not only legal, however. The technical aspects of such code can also harbor risks for the enterprise. Although many open-source aficionados would disagree, many software experts claim that open-source code, in which the blueprints for the applications are made public, is more dangerous than proprietary source code, in which they are secret, because hackers can use the source code to find and exploit flaws. And it is no secret that most enterprises today are very concerned about the security of their information systems.

Accordingly, management practices must not only guard against potential violations of the copyright law, which can lead to liability for damages, and the risk that a strategic transaction might be derailed during due diligence, but must also consider the technical security risks posed by open-source code.

It is clear that enterprises should carefully manage the use of open-source code within their organizations. But if one is tempted to deal with the risks posed by open-source code by adopting a policy prohibiting its use, forget it. Attempting to do so would be futile and counterproductive. Enterprises are using open-source code at an accelerating rate. OpenLogic reports that in 2006, enterprises on average used 75 different open-source packages and that the number grew to 94 in 2007. See "OpenLogic Recaps 2007 Open Source Trends" (Jan. 22, 2008).

One of the causes of this growth is the widespread adoption of the "software assembly" model, whereby developers create a product by using existing code from a variety of internal and external sources, thereby making it difficult to determine the ownership of the product. Another cause is the use of offshore programmers, who like to use open-source code and mix it with the proprietary code they create for many U.S. corporations. Although both developments are welcomed in many quarters because they permit software to be developed rapidly, thereby reducing costs, such savings must be weighed against the risks associated with the uncontrolled use of open-source software.

The use of open-source software brings with it risks that enterprises cannot ignore. Accordingly, a good policy must balance the desire to use open-source code against the risks that it poses.

Regulating the use of open-source code could take a variety of forms. An open-source policy could range from a simple one for a company with relatively little need to use such software to an elaborate structure that meets the needs of a public company. Typically, such policies are created and implemented by a committee that includes technical, business development and legal representatives.

Such committees should have the responsibility and authority to govern all use of open-source code within the enterprise, including both internal use and the incorporation of open-source code into commercial products. Although such committees are usually tasked with reviewing and deciding requests to use open-source code, they can reduce the number of requests that they must review by categorizing certain items of open-source code.

The committee could prepare a list of open-source code that, because of the licenses to which it is subject, may be used at any time and some code that may be used for certain purposes (i.e., an "approved" list). On the flip side, the committee could also prepare a list of open- source code that, because of the licenses to which it is subject, may not be used at any time or for any purpose (i.e., a blacklist). Open-source code that is on neither list would be subject to a review process to determine whether the enterprise should use the code at all and, if so, under what circumstances.

CONSIDERING HOW CODE WILL BE USED IN THE ENTERPRISE

But preparing an approved list and a blacklist is only a start. The committee should also consider not only the provisions of the license that governs the use of the open-source code, but also how the code is proposed to be used within the enterprise. Consequently, the committee must thoroughly understand the needs of the enterprise. Some uses of open-source code will not put the enterprise at risk, while other uses, such as incorporating it into commercially distributed software, could pose serious problems. The committee should also address the use of open-source code by outside contractors, including how the enterprise will monitor such use, as well as the contractors' compliance with the policies of the enterprise. Periodically, the committee should oversee an audit of some or all of the enterprise's software to determine what open-source code is present and whether the committee's management of the use of open-source code has been effective.

Clearly, open-source code holds great promise. It can help enterprises run their organizations more efficiently and bring their products to market more quickly and at lower cost. With millions of contributors around the world, many open-source packages offer products that are vastly superior to commercially available products. Nevertheless, such code poses risks. In order to avoid the risks described above, enterprise executives must understand the provisions of the licenses that govern the open-source code being used within their organizations and develop effective policies for managing such use.

William H. Venema is a member in the Houston office of Epstein Becker & Green whose practice focuses on information technology contracts, outsourcing, licensing, joint ventures, M&A and private equity. He is the author of "The Strategic Guide to Selling Your Software Company: Essential Advice from a Veteran Deal Warrior" (Windermere 2006).