# Morgan Lewis

## article

# communicate

## Open Source Software Guidelines
## for Financial Institutions
*Wall Street Lawyer*

# Open Source Software Guidelines for Financial Institutions

**by Rahul Kapoor and Matthew Miller***

## Introduction

The use of free and open source software ("FOSS") by technology-driven software companies has expanded dramatically in the last few years. Acceptance of FOSS—essentially software that users are allowed to run, study, modify, and redistribute without paying a licensing fee—by many reputable companies[1] has encouraged entities in other fields to consider incorporating FOSS into their business models. This article describes why FOSS is becoming increasingly important, gives a brief introduction to some of the most common open source licenses, and summarizes federal guidelines regarding the use of FOSS by financial institutions.

## FOSS Guidelines

The Federal Financial Institutions Examination Council ("FFIEC") was established pursuant to the Financial Institutions Regulatory and Interest Rate Control Act of 1978. Functioning as a formal interagency body, the FFIEC was empowered to establish uniform principles, standards, and report forms for the examination and supervision of financial institutions by the Board of Governors of the Federal Reserve System, the Federal Deposit Insurance Corporation, the National Credit Union Administration, the Office of the Comptroller of the Currency, and the Office of Thrift Supervision. On October 21, 2004, the FFIEC published guidelines[2] for examiners, financial institutions, and technology service providers on the acquisition and use of FOSS. These guidelines very likely will result in greater use of FOSS by financial institutions.

In its guidelines, the FFIEC states that, while the use of FOSS does not pose risks fundamentally different from the risks presented by use of proprietary software or software developed in-house, a company acquiring and using FOSS must implement unique risk management procedures. We will discuss the guidelines in more detail after we cover the necessary background information.

## Importance and Impact of FOSS

In a recent survey of 420 business-technology professionals, two-thirds stated they already use open source software, and an additional 16% said they plan to use open source software in 2005.[3] The migration from

**\* Rahul Kapoor (rkapoor@morganlewis.com) is a Partner at the law firm of Morgan, Lewis & Bockius LLP. He is a patent attorney with significant industry experience in the area of software integration; commercial and technology transactions, including software licensing and development outsourcing; patent prosecution; and intellectual property litigation. Mr. Kapoor teaches a class on Intellectual Property Strategy at Boalt Hall School of Law, University of California–Berkeley. Matthew Miller (mmiller@morganlewis.com), an Associate, is also a patent attorney. Mr. Miller's practice focuses on commercial and technology transactions with particular emphasis on software licensing and related professional services agreements.**

proprietary software to open source software is driven in large part by the desire to cut software acquisition costs.[4]

Unlike proprietary software, FOSS can be loaded on computers without a license fee. Moreover, installation of additional copies of FOSS does not prompt an increase in licensing fees for the user. For example, by using Linux as the operating system on its servers, a company does not incur additional licensing fees when it adds more servers to match its operational needs. Each new server can be installed with the existing version of Linux the company already is using.

. . . . . . . . . . . . . . . . . . . . . . . . . . . . .

### [A] company acquiring and using FOSS must implement unique risk management procedures.

. . . . . . . . . . . . . . . . . . . . . . . . . . . . .

While cost cutting is a significant motivation to begin using FOSS, institutions cannot risk using an inferior product merely to protect their bottom line. The financial services industry, where institutions process and catalog an extraordinary number of transactions on a daily basis, is particularly sensitive to this issue. The success of FOSS, both within and without the financial services industry, indicates open source software is not merely a cheap alternative to proprietary software, but one that also offers technical advantages.

Technology giants such as IBM, Hewlett-Packard, and Intel are contributing technology, marketing muscle, and thousands of professional programmers to Linux.[5] IBM alone has 600 programmers dedicated to Linux, up from two in 1999.[6] FOSS supporters expect that such a group effort, coordinated across multiple technology leaders, will result in a technically superior product in comparison to proprietary operating systems. In fact, Linux already is making its way into Motorola cell phones, Mitsubishi robots, eBay servers, and NASA supercomputers that run space-shuttle simulations.[7]

Google clearly demonstrates the potential benefits of using servers with low cost FOSS operating system software to analyze, store, and serve up enormous amounts of data. Google's network system is estimated to have grown to more than 100,000 Linux servers in data centers across the globe and is capable of serving up nearly 200 million searches on any given day.[8] The system is easily scaled by adding additional Linux servers at a fraction of the cost of adding servers with proprietary operating systems.

Finally, some entities make the switch to FOSS to avoid becoming too dependant on any particular proprietary software. For example, Axciom Corporation replaced the Unix operating system on its computer servers with Linux and the company is considering doing the same with its desktop computers.[9]

. . . . . . . . . . . . . . . . . . . . . . . . . . . . .

### [O]pen source software is not merely a cheap alternative to proprietary software, but one that also offers technical advantages.

. . . . . . . . . . . . . . . . . . . . . . . . . . . . .

The use of open source software has increased dramatically in recent years as institutions grow more accustomed to the benefits and risks of using FOSS. Market researcher IDC has forecast that sales of servers using Linux will grow faster than the overall market at least through 2008, and sales of Linux servers could account for 25% of server unit shipments by that time (compared to 15% in 2003).[10] In addition to Linux, prominent FOSS programs include Sendmail, Apache, and Firefox. The Apache Web server is estimated to be in use at more than 40 million Web sites.[11] Apache currently runs on nearly 69% of public Web sites, as opposed to roughly 21% for Windows servers.[12] Sendmail, an open source e-mail routing program, is used to route 60% of Internet e-mail traffic.[13] The Firefox Web browser, only recently released in November 2004, has forced Microsoft Internet Explorer's market share below 90% for the first time this decade.[14]

Although FOSS programs are generally available without any license fees, many companies have been built around the FOSS industry. According to a study conducted in 1999, the fees for a software solution are approximately 30% license and 70% implementation,[15] and there is little to suggest these percentages have changed in recent years. Some companies (Red Hat, Inc., for example) capitalize on this ratio and obtain their revenues by offering consulting services, such as warranty and implementation services,

relating to FOSS. According to Red Hat's public filings, the company offers a customized version of the Linux operating system without any license fee. Then, Red Hat charges for the services and warranties they offer to support implementations of that system. This business model targets a number of the risks enumerated by FFIEC, including lack of systems integration, support, code integrity, documentation, and warranties.

Several financial institutions, including Italian bank Banca Popolare di Milano, are already making successful use of FOSS.[16] Banca Popolare recently faced a dilemma with its legacy banking applications totaling approximately 90 million lines of code written in Cobol. The bank was having significant difficulty maintaining these applications on IBM's aging OS/2 Presentation Manager operating system, but was unwilling to throw away the Cobol code. Banca Popolare found a solution by using a proprietary integration tool from a third party vendor that allowed the bank to connect its legacy banking applications to IBM's WebSphere product with the system running in a Linux partition on the bank's mainframe computer. Banca Popolare successfully used Linux, licensed for free, as a commodity platform on which to build its solution. This strategy created significant value by preventing the bank's legacy applications from becoming obsolete, while at the same time increasing performance and customer satisfaction due to, among other things, faster transactions in branch offices.

Deutsche Börse Group, the German Stock Exchange, offers another example of the successful implementation of FOSS.[17] Deutsche Börse Group's Internet platform includes a combination of proprietary software, such as Documentum WCMS and an Oracle database, with FOSS software, such as the Jetspeed application portal, Apache Web server, and the Lucene search engine. This combined solution runs on an open source Jboss application with an integrated Tomcat servlet engine. In addition, Deutsche Börse Group used FOSS software as the basis for a system that controls the timeliness and quality of regular publications, such as annual financial statements and quarterly reports, that exchange-listed companies must deliver.

It should be noted that Deutsche Börse Group's experience with FOSS was not always easy. At times the implementation team needed to expend additional time and energy to find the necessary documentation to correct a problem, or had to enlist the services of outside vendors. However, the costs associated with such efforts were not great enough to offset the savings over the license and maintenance fees that would have been associated with the use of proprietary software. In addition, Deutsche Börse Group has been able to continuously improve the portal platform so that it is available to users 99.5 percent of the time.[18]

One final example of the impact of FOSS on a financial institution is Credit Suisse First Boston's migration to Linux to run one of its busiest trading applications when its RISC-based Unix infrastructure became outdated. The result has been an increase in revenue of $20 million from the company's trading desk because CSFB was able to leverage increased trade processing into a higher volume of business.[19]

## Open Source Software Licenses

The FOSS movement is not controlled by any group or entity. Consequently, there are numerous open source licenses (and licenses that purport to be open source) governing the use of FOSS. Each of these licenses has its own particular legal rights, obligations, and restrictions. The Open Source Initiative ("OSI"),[20] a non-profit organization focused on promoting the use and development of open source software, offers a certification mark program to indicate whether a given open source software license complies with OSI's open source definition. However, that definition, while widely accepted in the software industry, has created some confusion due to its vague language and often contradictory use of certain words and phrases.

One commentator has developed a more useful set of "Open Source Principles" to describe the key issues in open source software licenses. It is these Open Source Principles that make the use of FOSS desirable, while at the same time creating the need for the risk management procedures described in the FFIEC guidelines. The Open Source Principles are:[21]

1. Licensees may use open source software for any purpose whatsoever. An open source license may not restrict a licensee's use of the software. OSI's use of the phrase "free to use" in the open source definition is intended to mean free to use without any conditions that would restrict use. Restrictions such as "for research and noncommercial uses only" are not permitted.

2. Licensees can make copies of open source software and distribute them without payment of royalties to the licensor. This principle does not prohibit the licensor from charging for initial distribution of the software, but licensors cannot charge additional fees for further distribution of the software by a licensee. That means an interested party can obtain FOSS for free as a redistribution from a licensee. Thus, this principle pushes the cost to obtain such software from the copyright owner to the marginal cost of making the copy and effecting the distribution. If the copyright owner wants to charge a higher fee to provide the software to a user, the user is free to obtain the software for free from a licensee.

3. Licensees can modify the open source software and create derivative works thereof, and may distribute such modifications without payment of any royalties to the licensor. A licensor cannot charge a royalty for the right of licensees to create derivative works of the licensed software or impose other restrictions on the nature of the derivative works.

4. Licensees must get free access to and use of the source code of open source software. The ability to create derivative works is dependent on the licensee receiving the source code. The licensor is not required to actively distribute the source code to everyone, but must make it available for free to anyone who asks.

5. Licensees may combine open source software with other software. This includes combination into a single program, and combination on a storage media or hard drive.

Examples of widely used open source software licenses include the GNU General Public License, GNU Lesser General Public License, BSD Software License, Apache Software License, and Sun Community Source License.[22]

### GNU General Public License

The GNU General Public License ("GPL")[23] governs the use of Linux, among numerous other programs. The GPL has three main purposes. The first, and by far the most important, is to keep the software free, in the sense that it can be distributed and modified without additional permission of the licensor. This imposes a mirror-image restriction on the licensee, whereby the licensee must distribute any derivative works subject to the same limitations and restrictions as the original licensed software. The second is to ensure the licensee is aware that software under the GPL is distributed "as-is" and without any warranty. The third is to ensure the licensed software is free of any patents, or, to the extent any patent applies to the licensed software, it is licensed along with the code.[24]

Typically, the GPL is described as a viral license because its provisions reach through to any derivative work of the licensed open source software. The GPL also is known as a "copyleft" license—a term used to make the point that free software has a purpose opposite to the common proprietary uses of copyright law. A copyleft license involves a two step process: the creation of a copyrighted work, and the distribution of the copyrighted work under a license that mandates that any future software incorporating or modifying the original copyrighted work must be distributed freely and openly.

Incorporation of any software licensed under the GPL into a company's proprietary software could result in the company's entire proprietary software (including its source code) being licensed free of charge to all third parties under the terms of the GPL. This requirement applies to any software considered to be a derivative work of the original GPL-licensed software. If there are identifiable sections of the combined software code that are not derived from the software licensed under the GPL and can reasonably be considered separate works, the GPL will not apply to those sections.

While the GPL is most likely enforceable against licensees, the extent of its viral reach is unclear. Our courts have not yet directly addressed the enforceability and interpretation of the GPL.

### GNU Lesser General Public License

The GNU Lesser General Public License ("LGPL")[25] governs the use of open source software libraries and is generally thought to be less restrictive than the GPL. Under the LGPL, a library is defined as a collection of software functions and/or data prepared so as to be conveniently linked with application programs to form executables. (Put more simply, a library is a collection of reusable routines a programmer can call when writing a new program to avoid having to write new code for that routine.) If a company uses a library licensed under the LGPL, but does not comingle such library with the company's proprietary software prior to distribution, the company is not required to distribute the company's proprietary software free of charge under the LGPL.

### BSD Software License

The BSD Software License[26] was created to govern the distribution of a version of Unix created at the University of California, Berkeley, known as the "Berkeley Software Distribution." Today, the BSD-style license is applied to a wide variety of open source software.

Under the BSD Software License, redistribution of either the source code or object code forms of the licensed software, modified or not, requires retention of the applicable copyright notice and of the conditions and disclaimer contained in the license. If the foregoing requirements are met, incorporation of software licensed under the BSD Software License does not require modified versions of the licensed software or programs incorporating such software to be licensed free of charge to third parties. In other words, the BSD Software License permits a licensee to incorporate the open source software into the licensee's proprietary software and maintain the resulting program as proprietary.

### Apache Software License

The Apache Software License,[27] which governs the use of the Apache Web server and Jakarta Java™ applications (among other pro-grams), permits the redistribution of both the source code and object code forms of the licensed software, modified or not, provided the redistributed software retains the applicable copyright notice and the conditions and disclaimer contained in the license, and the end user documentation includes the applicable acknowledgement. If the foregoing requirements are met, someone incorporating software licensed under the Apache Software License need not license the modified versions of the licensed software or programs incorporating that software free of charge to third parties. As with the BSD Software License, the Apache Software License permits a licensee to incorporate the open source software into the licensee's proprietary software and maintain the resulting program as proprietary.

### Sun Community Source License

The Sun Community Source License ("SCSL")[28] is a hybrid license with both proprietary and open source aspects. The SCSL does not comply with OSI's open source definition; nor does it adhere to the Open Source Principles discussed above. Thus, it is not a true FOSS license.

The SCSL governs Sun's distribution of Java™ software development kits ("SDK") to developers. Under the SCSL, a licensee is permitted to maintain certain modifications of the SDK, such as performance enhancements or platform adaptations that are compatible with the licensed technology, as proprietary. However, all error corrections[29] (including the associated source code) and reformatted specifications must be provided to Sun, which may incorporate such corrections and reformatted specifications into future distributions of the SDK. Licensees are not required to share their modified code, other than error corrections, with the community, but are encouraged to do so.

One goal of the SCSL is to ensure compatibility of implementations of Java™ so programs written for the Java™ platform will function in the same way on any operating system running Java™. Thus, all modified versions of any software licensed under the SCSL must be certified for compatibility by Sun before they can be deployed by the licensee, either within its own organization or to third parties.

## FFIEC Guidelines

As noted above, on October 21, 2004, the FFIEC published guidelines on the use of free and open source software. The purpose of the guidelines is to "raise awareness within the financial services industry of risks and risk management practices applicable to the use of" FOSS.[30] FFIEC defines three types of risk associated with the use of FOSS: strategic risks, operational risks, and legal risks.

### Strategic risks

The first strategic risk identified by FFIEC relates to the ability of a licensee to customize FOSS. FOSS licenses provide source code for the acquired software to the general public. Thus, licensees can modify the software to suit their particular needs. In contrast, proprietary software is provided to licensees in object code form and cannot be modified by the licensee. Financial institutions must test the customized software to ensure performance and maintenance of confidentiality, integrity, and availability of systems and data.

> ### [I]nstitutions using FOSS should be careful to ensure the open source software meets their needs for compatibility and interoperability.

Second, interoperability of FOSS programs with a company's proprietary software may not be formally certified. Consequently, institutions using FOSS should be careful to ensure the open source software meets their needs for compatibility and interoperability. An institution may need to hire additional personnel to successfully integrate FOSS into its systems.

The third strategic risk relates to the maturity level of the software, particularly if the software will be used in mission critical applications. Mature software generally presents fewer risks than less mature software. Moreover, development of FOSS is fundamentally different from development of proprietary software; FOSS is developed in an open environment, where code is shared and modified by numerous unaffiliated parties. Therefore, institutions considering the use of FOSS should investigate the length of time the software has been actively used in non-

beta settings, the stability of the development community supporting the software, the availability of user guides and documentation, the number of distributors and value-added resellers supporting the software, and any security lapses in prior use of the software. Analysis of these factors will help institutions reduce the risks associated with FOSS. As a general rule, mature FOSS tends to have large and active development communities with a project leader determining which new or modified code is incorporated.

> ### FOSS is developed in an open environment, where code is shared and modified by numerous unaffiliated parties.

"Forking" is the fourth strategic risk associated with the use of FOSS. Forking occurs when the development community splits over the path of development for a given program and results in two or more different development paths for the same original software. Forks undermine the foundation of the open development process by splitting the user base, as well as the programmers who contribute to the project.[31] In the worst case scenario, a given fork is terminated or the development direction may be changed so significantly that the software no longer meets an institution's needs. To reduce the risk associated with forking, institutions should ensure adequate support is available for the current FOSS.

The fifth strategic risk deals with integration and support services for FOSS. Before opting to integrate FOSS software using its own employees, an institution should carefully consider employees' ability to identify, test, modify, install, upgrade, and support the software. Alternately, FOSS can be obtained from a distributor or systems integrator that ensures compatibility of the FOSS software with the institution's existing software and systems.

Finally, institutions must include both direct and indirect costs in their analysis when evaluating the total cost of ownership of FOSS. While the software itself is free, indirect costs of FOSS may be higher than those with comparable proprietary software.

## Operational risks

FOSS can be acquired from a relatively large number of uncontrolled and uncertified sources. Thus, confirming the integrity of the acquired code becomes critical. Institutions must implement procedures to ensure they obtain source code from a trustworthy entity and verify the integrity of the source code they incorporate into their existing systems.

Proprietary software is accompanied by extremely comprehensive documentation and user guides. Unfortunately, this is generally not the case with FOSS. To combat a potential deficiency in documentation, institutions should implement acquisition policies setting minimum documentation standards and establishing procedures for supplementing inadequate documentation.

. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

*Institutions must implement procedures to ensure they obtain source code from a trustworthy entity and verify the integrity of the source code they incorporate into their existing systems.*

. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

Unlike proprietary software, FOSS is not controlled by unique corporate entities. In fact, different "forked" versions of the same original FOSS code may be supported by various programmers. Any of these versions may be abandoned by the supporting programmers at any time. Therefore, each institution using FOSS must have contingency plans to protect against abandonment of a particular FOSS offering that the institution uses. To decrease these risks, institutions should ensure adequate support is available for the FOSS they plan to utilize. In addition, institutions should develop an exit strategy for replacing critical applications in the event their chosen FOSS is no longer supported.

The final operational risk set forth by FFIEC is the level of external support provided for the relevant FOSS. While external support is becoming more robust with the entrance of distributors, value-added resellers, and independent vendors, institutions should carefully evaluate such support and its related costs to determine the true benefits of FOSS. For example, distributors of Linux, including Red Hat, Inc. and Novell, Inc., include helpful user manuals, regular updates, and customer service with their Linux offering.[32]

## Legal risks

FOSS use is governed by one of more than fifty different licenses, with significant differences in the rights and restrictions contained in each. These licenses, some of which are discussed above,[33] are generally unclear in setting forth the rights and obligations of the parties executing the licenses. Thus, licensing of FOSS is fraught with risks, which institutions contemplating the use of FOSS need to understand.

. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

*Another legal risk faced by institutions using FOSS is the potential of being sued for patent or copyright infringement.*

. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

In general, FOSS licenses permit copying, distribution, and modification of the applicable software but do not contain any warranty or indemnification provisions. FFIEC guidelines recommend that institutions considering using FOSS seek competent legal counsel regarding the requirements and restrictions of the particular licenses governing FOSS. Legal counsel must have experience in the software arena and a strong technical background in order to fully understand the impact of each license on the institution's products and proposed integration plans.

Another legal risk faced by institutions using FOSS is the potential of being sued for patent or copyright infringement. Although this risk is also inherent with the use of proprietary software, it is far greater with FOSS. Again, unlike proprietary software, FOSS is developed in an open environment, which increases the possibility that proprietary code may inadvertently or intentionally be inserted into the FOSS at some point.

A well-publicized example of the potential legal risks associated with the use of open source software is the lawsuit by the SCO Group against IBM over IBM's contributions to Linux. SCO Group asserts that IBM placed proprietary Unix code into Linux in violation of a contract between the parties. Since bringing suit against IBM, SCO Group has commenced legal action against end users AutoZone and Daimler-Chrysler. To alleviate fears surrounding the use of Linux, Novell and Hewlett-Packard offered to indemnify users of Linux against certain infringement lawsuits. Red Hat has offered guaran-

tees that it will replace for free any code found to infringe SCO Group's intellectual property.

Institutions using FOSS can mitigate this type of risk by retaining highly qualified legal counsel for advice regarding the licensing of FOSS and the evaluation of any indemnities being offered by third parties. Such legal counsel must have a strong commercial licensing background and expertise in patent and copyright laws.

Finally, FOSS is typically licensed "as is" without any warranties or indemnities. Distributors and value-added resellers sometimes market FOSS with warranties and indemnities that cover the additional services, documentation, and support they provide. Institutions should evaluate the terms of these warranties and indemnities, as well as the entities' financial capacity to stand behind such provisions and provide a meaningful defense. If available, institutions also should consider third party insurance to protect against claims. Open Source Risk Management, a company created in the wake of the SCO Group's suit against IBM, has begun selling Linux users protection against copyright infringement claims.

## Conclusion

Once limited primarily to universities and independent programmers, use of FOSS by technology companies and other entities has increased steadily over the past few years. The FFIEC's recent guidelines should encourage financial institutions to incorporate FOSS into their businesses. Although FOSS does not pose risks that are fundamentally different from the risks presented by the use of proprietary software, financial institutions must be mindful of the FFIEC's guidance and implement adequate procedures to minimize the strategic, operational, and legal risks the use of FOSS presents. (W)

1   Dell, Hewlett-Packard, and IBM sell PCs and servers preloaded with Linux, an open source operating system similar in functionality to Unix.

2   "Risk Management of Free and Open Source Software," Federal Financial Institutions Examination Council (Oct. 21, 2004), *available at* <www.fdic.gov/news/news/financial/2004/FIL11404a.html>.

3   Helen D'Antoni, "Open-Source Software Use Joins The Mix," INFORMATIONWEEK (Nov. 1, 2004), *available at* <www.informationweek.com/story/showArticle.jhtml?articleID=51201599>.

4   *See, e.g.,* Steve Hamm, "Linux, Inc.," BUSINESSWEEK (Jan. 31, 2005) ("Holcim Ltd., the Swiss cement giant, just switched from Unix to Linux for some of its accounting, manufacturing, and

human resource applications. The attraction: 50% savings on hardware and 20% on software."), *available at* <www.businessweek.com/magazine/content/05_05/b3918001_mz001.htm>.

5   *Id.*

6   *Id.*

7   *Id.*

8   Computer Sciences Corporation, "Leading Edge Forum Report 2004, Open Source: Open for Business," at 27, *available at* <www.csc.com/features/2004/48.shtml>.

9   "Linux, Inc.," *supra* note 4.

10  Stephen Shankland, "IDC: Linux server sales to hit $9.1 billion in 2008," (Dec. 6, 2004), *available at* <http://news.com.com/IDC+Linux+server+sales+to+hit+9.1+billion+in+2008/2100-1010_3-5479681.html>.

11  Netcraft, "February 2005 Web Server Survey Finds 40 Million Sites on Apache," *available at* <http://news.netcraft.com/archives/2005/02/01/february_2005_web_server_survey_finds_40_million_sites_on_apache.html>.

12  *Id.*

13  "Case Study: Sendmail, Inc." 2004, *available at* <www.reasoning.com/pdf/SendMail_Case_Study.pdf>.

14  Press release dated November 22, 2004, *available at* <www.onestat.com/html/aboutus_pressbox34.html>.

15  John Koenig, "Seven Open Source Business Strategies for Competitive Advantage," (Feb. 7, 2005) at 6, *available at* <www.riseforth.com/images/Seven%20Strategies%20-%20Koenig.pdf>.

16  Malcolm Wheatley, "Myths of Open Source," CIO MAGAZINE (Mar. 1, 2004), *available at* <www.cio.com/archive/030104/open.html?printversion=yes>.

17  "Open Source: Open for Business," *supra* note 8, at 24.

18  *Id.*

19  Phil Hochmuth, "Financial Services IT execs weigh in on Linux's strengths," NETWORK WORLD FUSION (Jan. 24, 2003), *available at* <www.nwfusion.com/news/2003/0124linuxfin.html>.

20  Information about this organization is available at <www.opensource.org>.

21  *See* Lawrence Rosen, OPEN SOURCE LICENSING, SOFTWARE FREEDOM AND INTELLECTUAL PROPERTY LAW, Prentice Hall PTR, 2005, at 8-9.

22  The summaries we offer are not intended to address the application of these licenses to any specific fact pattern. Rather, they are presented as a general introduction to some of the issues raised by FOSS licenses and only describe a portion of the requirements of each license. Institutions should retain qualified legal counsel to evaluate the specific circumstances surrounding their potential use of FOSS.

23  The text of this license is available at <www.gnu.org/licenses/gpl.html>.

24  Andrew M. St. Laurent, UNDERSTANDING OPEN SOURCE & FREE SOFTWARE LICENSING, O'Reilly, 2004, at 36-37.

25  The text of this license is available at <www.gnu.org/licenses/lgpl.html>.

26  The form of this license is available at <www.opensource.org/licenses/bsd-license.php>.

27  The text of this license is available at <www.apache.org/licenses/LICENSE-2.0>.

28  The text of this license is available at <www.jcp.org/aboutJava/communityprocess/SCSL3.0.rtf>.

29  Error corrections are defined very broadly in the SCSL.

30  "Risk Management of Free and Open Source Software," *supra* note 2.

31  *See* UNDERSTANDING OPEN SOURCE & FREE SOFTWARE LICENSING, *supra* note 24, at 172.

32  "Linux Inc.," *supra* note 4.

33  For a fuller explanation of common open source licenses and the rights and limitations therein, *see* OPEN SOURCE LICENSING, SOFTWARE FREEDOM AND INTELLECTUAL PROPERTY LAW, *supra* note 21.