A Primer on Open Source Licensing Legal Issues: Copyright, Copyleft and Copyfuture

Dennis M. Kennedy*

Open Source[1] software has recently captured the public attention both because of the attractiveness and growing market share of programs developed under the Open Source model and because of its unique approach to software licensing and community-based programming. The description of Open Source software as "free" and the free price of some of the software undoubtedly attracted other attention. The Open Source movement reflects the intent of its founders to turn traditional notions of copyright, software licensing, distribution, development and even ownership on their heads, even to the point of creating the term "copyleft" to describe the alternative approach to these issues.[2] Open Source software plays a significant role in the infrastructure of the Internet and Open Source programs such as Linux, Apache, and BIND are commonly used tools in the Internet and business systems.[3] Companies such as Red Hat and Caldera have captured the public's attention by leaping to stratospheric valuations

* Dennis M. Kennedy (dkennedy@thompsoncoburn.com) is an attorney who writes and speaks frequently on legal, technology and Internet topics. He practices in the Intellectual Property and Information Technology Department at Thompson Coburn LLP in St. Louis. Dennis has collected many of his articles at www.denniskennedy.com. He received the 2001 "TechnoLawyer of the Year" award (www.technolawyer.com) for his role in promoting the use of technology in the practice of law and shared a 2001 Burton Award for Legal Achievement for an article he co-authored.

[1] Open Source, as used in this Article, refers generically to an approach to software development with unique licensing arrangements and a community-based method of programming. See definition of "open source," *at* http://whatis.techtarget.com/WhatIs_Definition_Page/0,4152,212709,00.html (last visited March 15, 2001). See OPEN SOURCES: VOICES FROM THE OPEN SOURCE REVOLUTION (Chris DiBona et al., eds., 1999) (hereinafter OPEN SOURCES), for an excellent general discussion of "open source." See also Donald K. Rosenberg, *Copyleft and the Religious Wars of the 21st Century*, *at* http://www.stromian.com/copyleft.htm (last visited March 15, 2001) (hereinafter Rosenberg, *Copyleft and the Religious Wars*), for a fascinating retelling of the history and lore of "open source."

[2] The term "copyleft" is discussed in more detail at note 65 *infra*. See definition of "copyleft," *at* http://whatis.techtarget.com/WhatIs_Definition_Page/0,4152,211840,00.html (last visited March 15, 2001). *See also* Free Software Foundation, *What is Copyleft?*, *at* http://www.gnu.org/copyleft/copyleft.html (last visited March 15, 2001) (hereinafter *What is Copyleft?*); Rosenberg, *Copyleft and the Religious Wars*, *supra* note 1; Robert W. Gomulkiewicz, *How Copyleft Uses License Rights to Succeed in the Open Source Software Revolution and the Implications for Article 2B*, 36 HOUS. L. REV. 179 (1999).

[3] Linux is a well-known and widely used Open Source operating system. Apache is Open Source web server software used on over half of the web servers on the Internet. BIND is an important infrastructure component of the Internet that translates URLs, such as www.domainname.com, into IP addresses.

after initial public offerings, even though they have later returned to earth.[4] The most well known of Open Source programs, Linux, is seen by some devotees as a realistic competitor to Microsoft Windows.

I. Open Source Fundamentals.

Fundamentally, the ready availability of source code to Open Sources programs and the right to modify and make improvements in that source code are what distinguish Open Source software from standard commercial software. What is source code? In simplest terms, software can be distributed in one of two code formats. The first format is executable or object code, which is, in essence, the instruction sequence for the computer processor. As a practical matter, object code is not human-readable and most software we use is provided to us in object code for us to run on our computers.[5] Source code, on the other hand, consists of the programming statements created by a programmer, commonly through the use of a text editor, which are readable by humans and which must be compiled before the programming statements can be run on a computer processor.[6] The source code may include comments and notes as well as programming, all in human readable form. Object code can also be thought of as source code after it has been compiled. Most license agreements for commercial software prevent the licensee/user from having access to the source code.

A fundamental tenet of Open Source software is that the licensee/user must get both the access to source code and, more important, the right to make changes to the source code to correct defects and bugs, customize programs or add features as the licensee/user deems appropriate. In the case of Open Source software, the licensee/user gets both object code and source code.

The Open Source system of software development is community-based. Because users have access to and the right to modify source code, Open Source programs evolve by means of the changes, suggestions and coding of potentially thousands of contributing programmers. Users who modify source code in ways that will improve the

---

[4] Red Hat, Inc. and VA Linux are two examples of Linux distribution and services companies that soared on the opening day of their initial public offerings. The price of VA Linux shares soared 733% on the first day of trading to $250 per share, the largest first day IPO increase in history. At the same time, Red Hat was up more than 1,800% over its IPO price in a matter of months. *See* Joanna Glasner, *VA Linux Sets IPO Record*, *at* http://www.wired.com/news/business/0,1367,33009,00.html (December 9, 1999). A year after its IPO, VA Linux stock had dropped to $12 per share.

[5] See explanation of "object code" in definition of "source code," *at* http://whatis.techtarget.com/WhatIs_Definition_Page/0,4152,213030,00.html (last visited March 15, 2001).

[6] *See id.*, for definition of "source code."

program contribute those programming changes, fixes and new features back to those responsible for the program's development. As a result, the software is improved and produced through the efforts of a large group of volunteer programmers, each of whom has come up with ways to fix or improve the program.[7] The efforts of these volunteers are managed by a person or small group of people who are responsible for the Open Source program. In the case of Linux, perhaps the most well-known Open Source example, this person is Linus Torvalds and a small group of trusted assistants.[8] These responsible parties incorporate selected contributions of volunteers and, from time to time, release new "official" versions of the software.

In addition, Open Source programs have evolved from the notion of "free software" conceived of by Richard Stallman in the early 1980s.[9] Stallman's useful aphorism that the "free" in free software should be thought of as "free as in speech, not free as in beer"[10] is a memorable and useful way of thinking about Open Source software. Many Open Source programs are in fact also free in the sense that they are generally available, at least by downloading over the Internet, without charge, except for copying costs. Companies in the Open Source sector generate revenues by methods other than selling software at a retail price, such as providing consulting services.[11]

It is important to focus on the "freedom" aspect of Open Source software. Most commercial software licenses prescribe limitations on how the software may be used. These limitations may include restricting the use of the software to certain machines, a specified numbers of users, internal business purposes and the like. Some of the Open Source licenses use license agreements in the opposite manner to minimize or eliminate restrictions on users that might prevent the free use of the software, source code and, in certain instances, derivative works based on the

---

[7] *See generally* Eric Raymond, *The Cathedral and the Bazaar* and other essays, *in* RAYMOND, THE CATHEDRAL AND THE BAZAAR (O'Reilly & Associates 1999).

[8] *See* Linus Torvalds, *The Linux Edge*, *in* OPEN SOURCES, *supra* note 1, at 101; Andrew Leonard, *Chapter One: Boot Time, Part 2: Starting Points*, http://www.salon.com/tech/fsp/2000/03/06/chapter_one_part_2/print.html (March 6, 2001) (hereinafter Leonard, *Boot Time*).

[9] *See* Richard Stallman, *The GNU Operating System and the Free Software Movement, in* OPEN SOURCES, *supra* note 1, at 53.

[10] Gomulkiewicz, *supra* note 2 (quoting Richard Stallman); *see also* Chris DiBona et al., *Prologue* to OPEN SOURCES, *supra* note 1, at 3.

[11] *See, e.g.*, Bob Young, *Giving It Away*, *in* OPEN SOURCES, *supra* note 1, at 113; Frank Hecker, *Setting Up Shop: The Business of Open Source Software*, *at* http://www.hecker.org/writings/setting-up-shop.html (last modified June 20, 2000). For example, a company might provide the software for free, then have customers pay for customization and consulting services.

software.[12] As a result, an Open Source license reverses traditional licensing concepts by using the license to give the licensee more freedom rather than more restrictions. For example, while a standard commercial license might prohibit any access to source code for any purpose, some Open Source licenses not only require that source code be distributed with the program (or made easily available) but specify that the same terms also apply to source code for any derivative works.[13]

The unique philosophy of Open Source plus the recent explosive growth in availability and usage of Open Source software have focused interest on how Open Source licensing works and its implications. As Open Source has come into the spotlight, a number of points have stirred special interest. The licenses on which the Open Source approach is based have never been conclusively tested in court.[14] The interest and involvement of commercial software entities with conflicting agendas has stressed and tested the loose, community model of development that has served the Open Source movement so well. The notion of copyleft that forms the basis of Richard Stallman's view of Open Source and its assumptions about existing copyright law and its implications have been criticized from inside and outside the Open Source community.

For these and other reasons, it is important to take a closer look at the Open Source licenses from a legal perspective. This article will, therefore, discuss the Open Source history and the role of the Open Source Definition, describe the general categories of Open Source licenses, survey generally some of the legal issues raised in the Open Source approach and with the Open Source licenses, and draw some tentative conclusions about the likely impact of Open Source on traditional copyright and licensing law as it becomes a more significant component of the Internet and computer systems, as well as a part of our way of thinking about intellectual property and licensing in a rapidly changing world.

---

[12] *See, e.g., What is Copyleft?, supra* note 2.

[13] *Id.*; *see also* Mike Perry, *Open Source Licenses*, http://fscked.org/writings/OpenSource.html (last visited March 28, 2001).

[14] ROSENBERG, OPEN SOURCE: THE UNAUTHORIZED WHITE PAPERS 92 (M&T Books 2000) (hereinafter ROSENBERG, OPEN SOURCE).

II. The Open Source Stories.

There are two primary threads in history of the Open Source movement. The first story centers largely around Richard Stallman, the Free Software Foundation, and the efforts that led to the GNU General Public License ("GPL") and the software released under the GPL. The second thread centers on the University of California at Berkeley and its role in the development of Unix operating system and the efforts that led to the BSD Unix family of programs and the BSD family of licenses. Both of those threads trace their roots back into the 1970s and 1980s.[15] A more recent development that played a key role in the history of Open Source occurred in 1998 when Netscape released its browser software under an Open Source license through an organization known as Mozilla.org.[16] In connection with the Mozilla project, Bruce Perens developed and published the Open Source Definition and the Open Source Initiative was established.[17] Although not without controversy, the Open Source Definition and the accompanying effort to use the term "Open Source" rather than "free" has helped to make Open Source licensing more organized and more accessible to commercial entities interested in Open Source development.

A. Richard Stallman, the Free Software Foundation and the General Public License.

The Free Software Foundation thread of the Open Source story traces its roots back to Richard Stallman at the Massachusetts Institute of Technology and some of the pioneering years in the development of computer technology and programming as we now know it.[18] Stallman became frustrated when he found that certain software licenses drastically restricted rights of the user and prevented programmers from fixing, adapting or developing software as they wished or prohibited access to source code as software vendors become more concerned with intellectual property rights. He felt that these licenses cut against the ways that software was being developed and used by the hackers and programmers who were taking software to new levels. The programming culture of the time created an environment where programmers freely worked on programs with each other, contributed fixes for the

---

[15] *Id.* at 3-13. *See also* OPEN SOURCES, *supra* note 1 (providing an excellent history of the roots and development of the Open Source movement); Andrew Leonard, *How Big Blue Fell for Linux*, *at* http://www.salon.com/tech/fsp (Sept. 12, 2000) (same).

[16] The Mozilla Organization, *at* http://www.mozilla.org (last modified March 15, 2000); Jim Hamerly et al., *Freeing the Source: The Story of Mozilla*, *in* OPEN SOURCES, *supra* note 1, at 197.

[17] See *The Open Source Definition*, *in* OPEN SOURCES, *supra* note 1, at 171, for an explanation of the open source definition and its history.

[18] *See* Stallman, *supra* note 9, at 53.

general public good, and saw development in a community context where people were free to take advantage of the innovations and improvements that others created, while still giving attribution and acknowledgment for the efforts of individual programmers. Stallman's vision and philosophy was that software should be free (as in speech, not as in beer) and that the programming that he and others did for the good of the community large, not for the benefit of individual commercial entities. Stallman and others believed that proprietary, commercial development of software would lead to a number of problems relating to security, loss of innovation, incompatibilities and the like, in part because it reduced the number of skilled, independent programmers who could analyze and correct source code.

After thinking about these issues in great detail, Stallman, with the assistance of law professors, published the General Public License, commonly known as the "GPL."[19] The GPL as been referred to as part manifesto and part license, because in it Stallman spells out the underlying philosophy of "free software" and, to an extent, codifies his views about software.[20] The essence of the GPL is revealed in its preamble: "when we speak of free software, we are referring to freedom, not price. Our general public licenses are designed to make sure you have the freedom to distribute copies of free software (and charge for the service if you wish), that you receive source code or can get it if you want it, that you can change the software or pieces of it in new free programs; and that you know that you can do these things."[21] Here are the three essential components of free software: the right to distribute, the right to get source code, and the right to modify. The preamble refers to the restrictions in the license that protect these rights and also says that these protections "translate to certain responsibilities."[22] To illustrate the effect of the GPL, "if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights you have. You must make sure that they, too, receive and can get the source code. And you must show them these

---

[19] Free Software Foundation, Inc. ("FSF"), *GNU General Public License*, *at* http://www.gnu.org/copyleft/gpl.html (last visited March 15, 2001).

[20] *See* FSF, *The GNU Manifesto*, *at* http://www.fsf.org/gnu/manifesto.html (last visited March 15, 2001), for Stallman's philosophical underpinnings for free software.

[21] FSF, *GNU General Public License*, *Preamble*, *at* http://www.gnu.org/copyleft/gpl.html (last visited March 15, 2001) (hereinafter *GNU License*, *Preamble*).

[22] *Id.*

terms so they know their rights."[23] The GPL emphasizes that the terms be clearly known, in reaction to the restrictions often buried in the "fine print" of traditional commercial software licenses.

Stallman also founded the Free Software Foundation in 1985 and began efforts to develop free software under the GPL license.[24] Stallman was a well-known and respected programmer and these development efforts resulted in useful and important software, including GCC, a widely used set of programming libraries for the "C" computer language, and Emacs, a widely-used text processor. The GPL has been modified somewhat over time and is currently in version 2.0. A separate license known as the Library General Public License or Lesser General Public License was implemented to address specific issues not covered under the GPL related to commercial and proprietary programs and their interaction with GPL programs.[25] Both the free software philosophy and the useful software played important roles in the popularization of free software and the GPL. The GPL model gradually began to spread among programmers who were interested in both the software and the philosophy underlying this approach. Stallman has also been a visible and outspoken advocate of the GPL and free software and his personality and presence has played an important role in its development.

In perhaps the most significant development in the spread of popularity of the GPL, Linus Torvalds, then a Finnish computer science student, in 1981 began the development of the operating system known as Linux.[26] Linux was released under the GPL, although it is important to note that Torvalds interprets some aspects of the GPL differently from Stallman.[27] As Linux grew in popularity, becoming commonly used in the infrastructure of the Internet and popularly known as a network operating system more stable than Microsoft Windows NT (at a much lower cost), awareness of the GPL its implications for the development of Linux and for programs associated with Linux grew in the developer and user communities.

---

[23] *Id.*

[24] *See* Stallman, *supra* note 9, at 60-61.

[25] *GNU License*, *supra* note 19. See Stallman, *supra* note 9, at 63, for Stallman's rationale for and description of the LGPL.

[26] *See* Torvalds, *supra* note 8, at 101; Leonard, *Boot Time*, *supra* note 8; Glyn Moody, *The Greatest OS that (N)ever Was*, *at* http://www.wired.com/wired/5.08/linux.html (last visited March 27, 2001).

[27] In fact, Torvalds added a special preamble to the GPL with the Linux kernel that clarifies that the copyright does not cover user programs that use kernel services by normal systems. This use, according to Torvalds' preamble, does not fall under the definition of "derived work" for purposes of the GPL. Linus Trovalds, *Preamble to the GPL*, *in* OPEN SOURCES, *supra* note 1, app. B at 263. Torvalds discusses these issues in Torvalds, *supra* note 8, at 108-09.

B. Berkeley, Bill Joy and BSD.

In the late 1970s and early 1980s, computer programmers at the University of California at Berkeley were developing and improving a version of the operating system known as Unix, originally developed and licensed by AT&T.[28] Bill Joy, a highly regarded programmer who later was a co-founder of Sun Microsystems, played a large role in the efforts improve Unix. These efforts led to software that later became known as the Berkeley Software Distribution ("BSD") of Unix, and a number of variants, including FreeBSD and OpenBSD. The BSD Unix was based on a version of Unix originated at AT&T, a situation that would lead to later issues and litigation.

The ideas behind BSD Unix were similar in many ways to those of the Free Software Foundation and the parallels in the approaches are intriguing. Source code was made readily available. Programmers could make derivative works as they saw appropriate to fix bugs, make improvements and fine tune the program. The general notion was that these modifications and improvements would be contributed back to the common good of the underlying software, although there was not a requirement to so return the new or improved code. Both the Berkeley and GPL approaches were community-based in nature. Also, as did Stallman, Berkeley generally charged only relatively small fee for copying the program in source code on to a medium usable by the licensee. The fee was designed to cover costs.[29] As the program become more popular, Joy crafted a short and simple version of a license I will refer to as the BSD License that allowed licensees to work with source code and make derivative works.[30]

The BSD License, while similar in a number of important ways to the GPL, did not require that the derivative works also be subject to the same terms as the initial BSD License. Therefore, the GPL notion of copyleft is not present in the BSD License. This distinction between the two licenses is an important one. The BSD has served as a model for a number of other Open Source licenses.[31]

---

[28] There are a number of detailed histories of the events at Berkeley involving Unix. *See* Andrew Leonard, *BSD Unix: Power to the People, From the Code*, *at* http://www.salon.com/tech/fsp/2000/05/16/chapter_2_part_one/print.html (last visited March 29, 2001) (hereinafter Leonard, *BSD Unix*); Kirk McKusick, *Twenty Years of Berkeley Unix, in* OPEN SOURCES, *supra* note 1, at 31. Interestingly, Donald Rosenberg simply refers to this period as the "computer world's equivalent of the Thirty Years War." Rosenberg, *Copyleft and the Religious Wars*, *supra* note 1.

[29] *See* Leonard, *BSD Unix*, *supra* note 28.

[30] Bill Joy has said that he modified a University of Toronto license agreement to create the first BSD license. *The Joy of Linux*, LINUX MAGAZINE, November 1999

[31] For example, the MIT X, XFree86, and XOpen licenses, as well as the FreeBSD, OpenBSD and NetBSD variants of the license. *See* ROSENBERG, OPEN SOURCE, *supra* note 14, at 99.

In the early 1990s, AT&T reassessed the value of its intellectual property associated with Unix and determined that Unix was a corporate asset that should produce more revenue than it was, and, in a controversial move, began to require Unix licenses at much higher fees than it had done in the past.[32] AT&T's approach was controversial both in that AT&T wanted to charge higher, and in some cases unaffordable, license fees to programmers who had in fact improved Unix and kept it viable over the years and that it raised concerns that AT&T had some claim over source code modifications that had been freely provided by the BSD Unix community. At the same time, issues arose with respect to BSD Unix because, although massively changed from the original Unix, it still contained small modules of the original AT&T Unix code of AT&T.[33] Litigation arose with respect to these issues, but the suit was settled on confidential terms without any definitive court ruling on the license itself or other relevant issues.[34] As part of that process, the BSD community replaced the remaining AT&T Unix code with original code and eliminated this issue.[35]

C. Netscape, Mozilla and Opening to Commercial Developers.

In 1998, Netscape Corporation made a strategic decision to release its browser program, Netscape Navigator, then the most popular Internet browser, as an Open Source project.[36] This decision was due in part to increased competition from Microsoft's browser, Internet Explorer, as a result of activities by Microsoft that were later the subject of an important antitrust case against Microsoft.[37] Netscape decision-makers were also intrigued by the Open Source style of development as it was explained in the highly influential essay by Eric Raymond called *The Cathedral and the Bazaar*.[38] In *The Cathedral and the Bazaar*, Raymond argued for the "bazaar" approach to software development, essentially the Open Source model as demonstrated by Raymond's Open Source experience

---

[32] The licensing fee for AT&T's Unix increased from $99 to, in some cases, over $250,000 over the course of several years. *See* Leonard, *BSD Unix*, *supra* note 28; Young, *supra* note 11, at 121.

[33] McKusick, *supra* note 28, at 44-45.

[34] *See id.*; Leonard, *BSD Unix*, *supra* note 28.

[35] *See* McKusick, *supra* note 28, at 44-45; Leonard, *BSD Unix*, *supra* note 28.

[36] See Hamerly et al., *supra* note 16, at 197, for a good discussion of the Mozilla project.

[37] U.S. v. Microsoft, 530 U.S. 1301 (2000).

[38] RAYMOND, THE CATHEDRAL AND THE BAZAAR (O'Reilly & Associates 1999).

in charge of a software project. The "bazaar" was a community-based model with many strengths not necessarily visible to the casual observer. The bazaar notion suggests a marketplace of ideas and an exchange of information. Raymond contrasted the "bazaar" approach with the tradition commercial model, a single source, highly managed, or "cathedral", approach. In simplest terms, the "cathedral" approach can be read as a commentary on Microsoft and its development of Windows. The contrasting notions of freedom and control that characterize the relationship of traditional commercial software development and the Open Source movement are brilliantly captured by the cathedral and bazaar metaphors.

Adopting an Open Source model of browser development was appealing to Netscape for both competitive business reasons and as a way to raise the level of its software by taking advantage of the positive benefits (reduction of bugs, et al.) that Raymond argued were benefits of the Open Source model. The move to Open Source also fit Netscape's culture and reputation of taking innovative approaches, including its early business strategy of making the Netscape browser widely available by distribution over the Internet, in essence giving it away in order to create revenues from other specialized products and services.[39]

Netscape's intention to release its browser as an Open Source browser resulted in two key developments in the history of Open Source. First, the discussions about the licensing issues involved a number of Open Source 'heavyweights" and led directly to the publication of the Open Source Definition discussed below. Second, as a commercial developer, Netscape was very aware of the significant issues raised under the existing Open Source licenses in the case of a conversion of a commercial product to Open Source, especially for commercial projects where the underlying code incorporated software licensed from a large number of developers. In order to release the Netscape browser under an Open Source license like the GPL, each of the underlying licensors of software used in the browser program would have to agree to release their incorporated code under the Open Source license ultimately chosen by Netscape. The project of getting those agreements was daunting and unlikely to be accomplished with the assurance of either completeness or timeliness. As a result, like many commercial developers who have investigated the idea of releasing commercial software products under an Open Source license, Netscape decided that the best approach would be to write its own Open Source license. That new license would attempt to address Netscape's specific issues while keeping within the general requirements of the Open Source

[39] See CUSAMANO & YAFFIE, COMPETING ON INTERNET TIME (Free Press 1998), for a history of Netscape.

types of licenses and making some "improvements" to the language and coverage of the license based on recommendations of lawyers.[40]

Netscape's initial efforts in this regard are instructive in a number of ways. First, they illustrate the tendency of commercial developers moving into Open Source to want to rewrite standard Open Source licenses to "improve them" in order to fill in gaps, deal with specific issues or add additional protections. Second, Netscape's original effort showed that Open Source is truly a community-based project and that the community will vocally react to and pressure anyone who wants to change the Open Source licenses. Open Source programmers can also withhold their efforts on projects if the license terms for those projects violate the spirit of the Open Source movement. In the case of Netscape, Netscape wanted a new license to cover both the existing licensed software within its program and to allow it to retain certain rights to release proprietary commercial versions of software developed as part of the proposed Netscape Open Source project. There is little question that Netscape's cause was helped because it enlisted the assistance of the Open Source community and, more important, it listened to the comments of the community and worked with it.

As a result of reaching a compromise form of license that was acceptable to the Open Source community, Netscape developed what is now known as the Mozilla Public License.[41] This license now serves as an important model of an Open Source license in situations where commercial software or commercial development is involved.

In connection with the Netscape efforts, Bruce Perens, who had been involved in the Open Source Linux community and was already working on an effort to create some standards for Open Source and to organize and define which licenses were Open Source and which were not, was asked to consult with Netscape on its efforts.[42] Perens ultimately converted his definition project into the Open Source Definition.[43] The Open Source Definition was designed to be a comprehensive statement of Open Source principles, a definition, and a way of tracking which

---

[40] *See generally* Hamerly et al., *supra* note 16, at 197, 200. Linus Torvalds and Eric Raymond were among the Open Source "heavyweights" consulted.

[41] Mozilla Organization, *Netscape Public License: Version 1.0*, *at* http://www.mozilla.org/MPL/NPL-1.0.htmlations (last visited March 29, 2001); ROSENBERG, OPEN SOURCE, *supra* note 14, at 298.

[42] Perens, *supra* note 17, at 171.

[43] *Id.* See The Open Source Initiative, *Open Source Definition: Version 1.8, at* http://www.opensource.org/docs/definition_plain.html (last visited March 27, 2001), for the "canonical" version of the Open Source definition. In connection with the Open Source Definition, the Open Source Initiative was established as a "caretaker" organization for the definition and for other initiatives and purposes.

licenses complied with the Open Source Definition. Setting standards and designating which licenses were Open Source would also benefit developers in commercial organizations and help them choose among the conforming licenses and act in ways that would be acceptable to the Open Source programming community. The Open Source Initiative was also established as an entity to maintain the Open Source Definition, to promote Open Source and to help ease the way for continuing involvement of commercial entities in Open Source projects. A lesser, but still important, goal of the efforts surrounding the Open Source Definition was an effort to move away from the notion of "free" software associated with Stallman.[44] The terminology of "free software" had proven difficult for commercial entities to understand and appreciate. "Open Source" was considered more neutral and less controversial than "free software."[45]

D. The Open Source Definition.

The Open Source Definition emphasizes the three key elements of free distribution, readily available source code and the right to make derivative works. The Open Source Definition contains nine separate provisions and then provides a list of licenses that conform to the Open Source Definition. An understanding of the provisions of the Open Source Definition is essential to any discussion of Open Source and this section of this Article will set out the nine elements of the Open Source Definition.

1. Free Distribution. "The license may 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 may not require royalty or other fee for such sale."[46]

2. Source Code. "The program must include source code, and must allow distribution in source code as well as compiled form. Where some form of a product is not distributed with source code, there must be a well-

[44] ROSENBERG, OPEN SOURCE, *supra* note 14, at 19-25.

[45] *Id.*

[46] This section means that the license may make copies, sell or give away the software and not have to pay any one for the privilege. *Open Source Definition: Version 1.0 Comments*, *reprinted in* Perens, *supra* note 17, at 176-80 (hereinafter *OSD Comments*). *See also* Gomulkiewicz, *supra* note 8, at 179 (discussing each section of the Open Source Definition in detail).

publicized means of downloading the source code, without charge, via the Internet. The source code must be the preferred form in which a programmer would modify the program."[47]

3. Derived Works. "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."[48]

4. Integrity of the Author's Source Code. "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."[49]

5. No Discrimination Against Persons or Groups. "The license must not discriminate against any person or group of persons."[50]

6. No Discrimination Against Fields of Endeavor. "The license must not restrict anyone from making use of the program in a specific field of endeavor."[51]

7. Distribution of License. "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."[52]

8. License Must Not Be Specific to a Product. "The rights attached to the program must not depend on the program's being part of a particular software distribution."[53]

---

[47] The *OSD Comments* say that the intent is "for source code to be distributed with the initial work, and all derived works." *ODS Comments*, *supra* note 46, at 177.

[48] The OSD Comments say that the intent is "for any modification of any sort to be allowed." The licensee must be allowed to distribute modified works under the same license terms as the initial work, but there is no requirement to do so. There are specific references in the OSD Comments to this section about the different treatments of this licensing issue by the GPL and the BSD licenses. Both licenses are considered "conformant" to the Open Source Definition. *Id.*

[49] The OSD Comments indicate that this section was designed to provide ways to protect authors who were worried that poorly-done modifications to their work by third parties would be perceived as their own work. The "patch" issue has raised a good deal of debate. The issue of appropriate attribution has always been an important issue in Open Source, since reputation take the place of remuneration for Open Source programmers. *See id.* at 178.

[50] The OSD Comments note that no matter how laudable the intent of the restriction, it will still not be permitted. *See id.* at 179.

[51] For example, there can be no restrictions against business use or use in genetic research or an abortion clinic. *See id.*

[52] The OSD Comments say simply: "The license must be automatic, no signature required." *ODS Comments*, *supra* note 46, at 179. The OSD Comments also note that there is some question about the validity of this type of contract. *Id.*

9. License Must Not Contaminate Other Software. "The license must not place restrictions on other software that is distributed along with the licensed software."[54]

It is important to note that the GPL, the BSD license and the Mozilla Public License are all considered "conformant" to the Open Source Definition.[55] With the release of the Open Source Definition and the general approach of using the term "Open Source" rather than "free" to describe the software, the acceptance of the Open Source style of development and licensing appears to be increasing. Stallman, however, still favors the approach of "free" software, emphasizing the notion of freedom, and the use of that term, but, as will be discussed later, some people in the Open Source movement have some problems with the GPL's approach to freedom and copyleft is not a required element of the Open Source Definition.[56]

E. Public Domain, Shareware and Freeware.

People often confuse Open Source software and public domain software and it is important to differentiate the terms.[57] While some software is released freely into the public domain and the author has no copyright rights, Open Source software most definitely is not. Open Source software is governed by a license and the owners of the copyright in the software continue to own the copyright and assert their rights thereto. An important way to think about Open Source software is that people are receiving a software license but a license that gives them more rights than they have become accustomed to expect under commercial software licenses.[58] A software author who chooses

---

[53] The OSD Comments clarify that a program may not be restricted to one of the several Linux distributions (e.g., Red Hat, Debian, etc.). *Id.*

[54] The OSD Comments emphasize that the concern is with aggregations (e.g., several programs placed on the same CD-ROM) rather that derivations, which are covered by Section 4. *Id.* at 180.

[55] *See* The Open Source Initiative, *Licenses, at* http://www.opensource.org/licenses (last visited March 27, 2001) (hereinafter *Licenses*), for a complete list of licenses certified as conformant to the Open Source Definition, along with hyperlinks to the licenses. *See also* FSF, *Various Licenses and Comments About Them, at* http://www.gnu.org/philosophy/license-list.html (last visited March 27, 2001) (hereinafter *Various Licenses*), for a useful list of licenses.

[56] *See* Stallman, *supra* note 9, at 69-70.

[57] *See* Perens, *supra* note 17, at 180-81, for a general discussion of public domain software. Note that simply releasing software into the public domain would not have achieved the goal of keeping software free that Stallman sought with copyleft. *See also* Gomulkiewicz, *supra* note 2, for a discussion of why licensing is preferred over placing software into the public domain.

[58] Perens, *supra* note 17, at 171.

to release his or her software into the public domain surrenders the copyright. Other people can then use the author's work as they see fit, including modifying it, removing the author's name, treating it as their own work, or even removing a particular version from the public domain by asserting copyright ownership.

Open Source software is also different from "shareware" or "freeware" software.[59] In each of those cases, the developer offers a standard license, with special pricing terms or at no price, but does not give access to source code or the right to make derivative works. While freeware or shareware often has the connotation of being "free", it is free as in beer, not free as in speech and its license would typically have many of the restrictions found in commercial licenses.

III. A Taxonomy of Licenses.

While a standard commercial proprietary software license is focused on protecting the copyright interests of the owner by restricting the uses of the software, it can be argued that the Open Source licenses place minimal restrictions on the use of software in order to "free" the software. The Open Source licenses all have in common a requirement that source code be made available and that users of the software have the right to make derivative works. The licenses all disclaim warranties and many make an effort to limit liabilities. For purposes of this article, I have divided the Open Source licenses into four distinct categories. These four categories are the GPL or "free" family of licenses, the BSD set of licenses, including the MIT X licenses, the Mozilla Public License, and, finally, other non-GPL and "commercial" licenses. The categories can be distinguished from each other in a number of ways, but the most important distinction is in the way that the license addresses the issue of permitting derivative works to be later made proprietary.

A. The GPL Licenses.

Richard Stallman's GNU General Public License has been described as a manifesto as well as a license.[60] It contains strong language about the freedom of software and the copyleft requirement that all derivative works also remain free. As such, the GPL and its family of licenses have been controversial for both Open Source and

---

[59] *See Shareware: A WhatIs Definition, at* http://whatis.techtarget.com/WhatIs_Definition_Page/0,4152,212977,00.html (last modified Nov. 30, 1999), for a definition of "shareware." *See also Freeware: A WhatIs Definition*, at http://whatis.techtarget.com/WhatIs_Definition_Page/0,4152,212159,00.html (last modified Oct. 18, 1999), for a definition of "freeware." In each case, the key distinguishing factor is the approach to pricing, not the method of licensing.

[60] Perens, *supra* note 17, at 181.

commercial developers, especially those who want to make their derivative works private or subject to a standard commercial license. Since the GPL is perhaps the "classic" example of an Open Source license, this section of this article looks at the provisions of the GPL in some detail.

In its preamble, the GPL talks about software freedom in the following way: "The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software – to make sure the software is free for all of its users."[61] The intriguing method for doing so is to place restrictions on the use of the licensed software that are different from traditional restrictions in standard commercial licenses. As the preamble says, "to protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender these rights." The preamble also emphasizes that software under the GPL has no warranty and that any patents must be licensed for everyone's free use or not licensed at all. The GPL covers specifically the rights to copy, distribute and modify the software.

In Section 1 (in some versions of the GPL, Section 1 is numbered as Section 2), the GPL allows licensees to copy and distribute verbatim copies of source code as it is received provided that licensees "conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this license and to the absence of any warranty; and give any other recipients of the program a copy of the license along with the program." A fee can be charged for the physical act of transferring a copy or for warranty protection.[62]

Section 2 permits the licensee to modify a copy of the program or any portion of it and copy and distribute the licensee's modifications under the terms of Section 1 so long as each modified file carries a prominent notice

---

[61] In simplest terms, the general public licenses are:

> [D]esigned to make sure that you have the freedom to distribute copies of free software (and charge for the service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

GNU License, *Preamble*, *supra* note 21.

[62] *Id.* The GPL sets out instructions for applying the GPL to a work by placing the appropriate notices. In all events, the recipient/license must get a copy (or access to a copy) of the license. At least one commentator has raised questions about how the notice requirements can be met if software is embedded on a chip and other implications of embedded software for the GPL. ROSENBERG, OPEN SOURCE, *supra* note 14, at 143. Note that the GPL starts out with Section 0, but other print versions of the GPL start with Section 1. Take care to correlate the comments in this section of this article with appropriate provisions of the GPL, regardless of the numbering convention employed.

saying that the licensee changed the file and any date of the change. Each work that is distributed or published that in whole or in part contains or is derived from the program or any part thereof must be licensed as a whole at no charge to third parties under the terms of the GPL. In the case of certain types of interactive programs, notices must be run when the program starts. While these requirements apply to the modified work as a whole, identifiable sections of work that are not derived from the free software are not covered by the license when distributed as separate works. In addition, mere aggregation of another work not based on a GPL program with the GPL program on a "volume of a storage or distribution medium" does not bring the other work under the scope of the license. Simply placing a GPL program on a CD-ROM with non-GPL programs does not have any effect on the GPL program or the non-GPL programs.

Section 3 requires that object code or executable code also be distributed with source code or the source code be made readily available in a number of permitted ways, including accompanying the object code on a "medium customarily used for software interchange," providing a written offer, valid for at least three years to provide the source code, and, in certain cases, accompanying it with the offer to provide source code originally provided with the program. This requirement emphasizes the key element that source code of Open Source programs must be made available so that the licensee can modify, fix or improve them.

Section 4 automatically terminates rights under the license if a user attempts to copy, modify, sublicense or distribute a program except as expressly provided under the GPL. Importantly, however, downstream parties who have received rights under the GPL will not have their licenses terminated as long as they stay in full compliance.

In Section 5, the GPL is set up as what is popularly known as a "shrinkwrap" or "click-wrap" type of license.[63] No signature is required and the simple act of modifying or distributing the program, or any work based on the program, indicates acceptance of the license and all terms and conditions of the GPL. Downstream distribution also gives the recipient an automatic license from the original licensor.

Sections 7 and 8 deal with patent issues, placing an obligation to license a patent for the benefit of all or not to distribute the code subject to the patent. Section 9 allows the Free Software Foundation to revise or create new versions of the GPL. Section 11 states that there is no warranty for any program "to the extent permitted by applicable law." The program is provided "as is." Section 12 provides that the copyright holder or any other party

[63] *See, e.g.*, Hanson & Covello, *Click-Wrap Licenses: The Pros and Cons*, NAT'L L.J., September 20, 1999, at B8, for a general discussion of cases on the shrinkwrap issue.

who modifies or redistributes the program in accordance with the license will not be liable for damages, including "any general, special, incidental or consequential damages arising out of the use or inability to use the program . . . even if such holder or other party has been advised of the possibility of such damages." The GPL also supplies detailed instructions on how to apply the GPL to a work.

In many respects, the provisions of the GPL reflect standard license agreements, at least in terms of format and the types of matters covered. While the advice of law professors is reflected throughout, the GPL varies from standard commercial licenses in several important ways, primarily in the types of standard provisions it leaves out. Verbatim distribution of copies of the license itself is permitted, but no one other than the Free Software Foundation is allowed to change the terms of the GPL. Not surprisingly, the GPL is considered "conformant" with the Open Source Definition.[64]

Perhaps the most important distinguishing aspect of the GPL is that it does not allow licensees to take modifications of GPL programs private or make them proprietary. Modifications must be distributed under the same terms contained in the GPL, including the copyleft provisions. While in many respects the terms of the GPL are relatively easy to understand, that is not necessarily the case with Section 2(b), which has been the subject of confusion and concern. Section 2(b) says that "you 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." (emphasis added). Interestingly, Section 0 of the GPL creates a defined term called "work based on the program" that means either the program itself or any derivative work under copyright law. More specifically, a work based on the program is, according to Section 0, "a work containing the program or a portion of it, either verbatim or with modifications and/or translated into another language." Note that Section 2(b) does not use the defined term "work based on a program." Instead, it talks about "works that in whole or part contain or are derived from a program or any part thereof." Therefore, an ambiguity arises as to whether "contains or derived from" means something different than a derivative work under copyright law.

The interpretation of this Section is significant because of the effect of Section 2(b) and the implications of copyleft. This effect is one of the most important issues involving the GPL. More specifically, Section 2(b) is at the heart of what is sometimes referred to as the "taint" or "viral" nature of the GPL, but it also gets to the essence of the

[64] *Open Source Definition: Version 1.0, reprinted in* Perens, *supra* note 17, at 180 (hereinafter *OSD*).

concept of "copyleft."[65] Since modifications of software licensed under the GPL must also be released under the GPL, there has been a great deal of concern that incorporating any code written under the GPL into another program will require that the second program in which the code is incorporated be licensed under the GPL. This concern has made some commercial developers wary of using GPL software, let alone modifying the code. In addition, there are questions whether applications that interface with or run on GPL software may be tainted and fall under the GPL.

These concerns led to the development of the Library General Public License or Lesser General Public License (LGPL).[66] The LGPL explicitly allows proprietary software to be used in connection with GPL software through the use of programming libraries without requiring that the proprietary software also be subject to the provisions of the GPL. The LGPL eased the acceptance of Open Source in the commercial environment by providing a comfort zone for owners of proprietary applications and has served to some degree as a model for the commercial Open Source licenses discussed later. Interestingly, the LGPL was developed by the Free Software Foundation as an accommodation to developers who were concerned about this issue, but the Free Software Foundation does not especially encourage the use of the LGPL.[67]

The GPL is not an exclusive license. The copyright owner who places the program under the GPL has the right to violate his or her own license and to license the original program under a commercial or other type of license. This right does not apply to works of others later incorporated into the program.

In summary then, the GPL tracks the general provisions of the Open Source Definition. The important feature of the GPL is that it does not allow for GPL software to be mixed with non-GPL software without making the non-GPL software also subject to the terms of the GPL. The GPL does not allow for modifications to be taken private. Under the GPL, source code must be released and, most important, derivative works must also be released under the terms of the GPL, which has led some to refer to the "viral" aspect of the GPL, since it can "infect" other

---

[65] *See* ROSENBERG, OPEN SOURCE, *supra* note 14, at 93-94, for a discussion on the "viral" effect of the GPL; Hamerly et al., *supra* note 16, at 201.

[66] *GNU License, Preamble*, *supra* note 21.

[67] In fact, it is recommended for "special circumstances only." *Various Licenses*, *supra* note 55.

software with the "freedom" of the GPL.[68] If a licensee distributes modifications of a program, source code must be made available and there is to be no fee other than copying and similar charges.

Examples of software used released under the GPL include GCC, Emacs, the Linux kernel, and various Linux distributions.[69]

B. The BSD Licenses.

I use the term "BSD Licenses" for a category of licenses that grow out of or are similar to the BSD License discussed above. The BSD Licenses are generally considered the least restrictive of the Open Source licenses.[70] Under the BSD Licenses, distribution of source code is permitted, but not mandated for derivative works. As a result, programs under the BSD Licenses can be combined with proprietary software. Some code under the BSD Licenses can now be found in common commercial software such as Windows NT and the Macintosh operating system, OS X.[71] The family of BSD Licenses includes the MIT X licenses, which are very similar to the BSD licenses both in concept and in language.[72] Sometimes the BSD Licenses have been referred to generically as either the MIT X Licenses, the X Licenses or the MIT Licenses.

The BSD Licenses are remarkably simple and do not look very much like standard commercial software licenses drafted by lawyers.[73] They allow a for redistribution and use of source code and object code with or without modification so long as the redistribution of source code retain required copyright and other notices and the disclaimer of warranties and limitation of liability clauses. In previous versions, the original BSD License had certain attribution requirements, including mandatory attribution and naming of contributors in advertising of

---

[68] *See* ROSENBERG, OPEN SOURCE, *supra* note 14, at 93-94, for a discussion on the "viral effect of the GPL; Hamerly et al., *supra* note 16, at 201. For a practical discussion of avoiding the "viral" issues, *see also Hecker*, *supra* note 11.

[69] GCC is the "GNU Compiler Collection," an important set of C programming libraries. Emacs is a popular and well-regard text editor. Linux is a well-known and widely used Open Source operating system.

[70] Perens, *supra* note 17, at 183; Hecker, *supra* note 11; Perry, *supra* note 13; Eric S. Raymond, *Chapter 2: On Not Reinventing the Wheel*, *at* http://www.tuxedo.org/~esr/writings/taoup/chapter2.html (last visited March 29, 2001).

[71] ROSENBERG, OPEN SOURCE, *supra* note 14, at 99.

[72] Perry, *supra* note 13; Perens, *supra* note 17, at 183.

[73] Bill Joy has said that he modified a University of Toronto license agreement to create the first BSD license. *The Joy of Linux*, *supra* note 30.

software using the code that some people saw as onerous.[74] Recent revisions have eliminated the most burdensome aspects of these requirements, including the problematic advertising clause.[75] However, licensees must pay close attention to BSD Licenses to ensure compliance with their terms, including the requirement for appropriate copyright and permission notices, and watch out for code under older versions of the BSD Licenses. The BSD Licenses have been used in the operating systems called FreeBSD, OpenBSD and the like, and for certain variations in the X family of programs. The Apache server software that runs the majority of Internet servers is released under the Apache License, which has terms similar to the BSD Licenses.[76]

Ironically, the BSD Licenses are considered by many to be more "free" than the GPL because, unlike the GPL, they permit developers to release derivative works under whatever license they prefer, including licenses without the same terms as the BSD license applicable to the original code. In other words, the BSD Licenses do not contain copyleft terms. This approach has made the BSD type of licenses more attractive to commercial developers. The BSD Licenses permit licensees to do nearly anything they wish with the source code, subject to the specific requirement of the license. In part this approach is because software under the X and BSD licenses originally was funded by monetary grants from the U.S. government.[77] The BSD Licenses illustrate an important aspect of the Open Source definition, namely that copyleft is not a requirement under the Open Source Definition. Copyleft restrictions are usually only found in the GPL types of licenses.

C. The Mozilla Public License and Related Commercial Licenses.

In connection with Netscape's release of its browser under an Open Source license in 1998, much effort was put into developing the appropriate Open Source license for the Netscape browser. As discussed earlier in this article, there were difficulties raised because the browser source code included code licensed under a variety of commercial licenses and permissions and relicensing of underlying code would have been required for the release of the code under the GPL in particular.[78] Netscape, not surprisingly, also had some concerns about its ability to use

---

[74] See discussion of the classic BSD License in Raymond, *supra* note 70.

[75] See discussion of "the modified BSD License" at *Various Licenses*, *supra* note 55.

[76] Note that some consider the Apache license as a relative of the BSD License, *OSD*, *supra* note 64, at 183, while others seem to place it in another category. ROSENBERG, OPEN SOURCE, *supra* note 14, at 183.

[77] Perens, *supra* note 17, at 183.

[78] Hamerly, et al., *supra* note 16, at 201.

source code developed under the license for its own proprietary products and to place software products developed under the Mozilla Open Source project under its own proprietary licenses in order to produce revenue for Netscape. Netscape consulted with a variety of leading figures in the Open Source movement in order to develop a license and also received commentary from the Open Source community.[79]

As a result, Netscape ultimately produced two licenses. The first license was the Netscape Public License which related specifically to the issues involved by the underlying third party licensed code in the Netscape browser and other concerns specifically related to the conversion of existing code.[80] More important, Netscape developed the Mozilla Public License.[81] The Mozilla Public License is in many ways a model for Open Source licensing for commercial software entities. Because Netscape solicited and received a variety of comments on the license, including input from lawyers, the Mozilla Public License looks and reads more like a standard commercial software license prepared by lawyers than do the GPL or BSD Licenses. For example, much effort was placed into defining terms like "Covered Software" to distinguish exactly what software and code was subject to the license and what was not.[82] The Mozilla Public License can be seen as an effort was made to combine the best features of the BSD Licenses and the GPL, despite the fact that different Open Source advocates can disagree on what are the best aspects of the two licenses.

Under the Mozilla Public License, commercially licensing of derivative works is permitted. Changes to covered program source code must be made freely available to anyone. Importantly, the Mozilla Public License does not contain the copyleft provisions of the GPL. Additions, as opposed to modifications, of covered source code

---

[79] *Id.* at 200.

[80] *Id.* at 201. See also discussion of Mozilla Public License in Hecker, *supra* note 11.

[81] Mozilla Organization, *Netscape Public License: Version 1.0*, *at* http://www.mozilla.org/MPL/NPL-1.0.htmlations (last visited March 29, 2001).

[82] Netscape, *Netscape Communicator Open Source Code White Paper*, *at* http://home.netscape.com/browsers/future/whitepaper.html (last visited March 29, 2001); Hecker, *supra* note 11 (discussing Mozilla Public license); Perens, *supra* note 17, at 184; Schallop, *The IPR Paradox: Leveraging Intellectual Property Rights to Encourage Interoperability in the Networking Computing Age*, 28 AIPLA Q.J. 195, 246 (2000). Note that Netscape does encourage the return of new code to Netscape. The Mozilla discussions pointed out the tensions caused by Netscape wanting to make new features contributed to the browser proprietary as opposed to wanting to do the same with bug fixes. The Open Source community balked on this proposed treatment of new features. *See* Hamerly et al., *supra* note 16, at 202.

that form a "larger work" may be licensed differently and published or not even published at all. In this sense, the Mozilla Public License is more like the BSD Licenses than the GPL.[83]

The Mozilla Public License is especially important because it serves as a model for the future releases of commercial software into Open Source. There has been a tendency for commercial developers who want to start with the Mozilla Public License and modify it to meet their specific situations. Unfortunately, however, the Mozilla Public License also was designed with some specific issues involving Netscape in mind. While those in the Open Source community would prefer that Open Source software be done either under the GPL or the BSD Licenses, the momentum toward "commercial" Open Source licenses along the lines of the Mozilla Public License is building.

D. Other Open Source Licenses.

There are a number of Open Source licenses that do not quite fit either of the above categories. IBM, Sun Microsystems and other large commercial software companies have developed variations of Open Source licenses.[84] Other examples of Open Source licenses that are important or have been around for a long time include the Artistic license for Perl programming and the Aladdin public license.[85] What many of these other licenses have in common is one or more subtle variations, although some can be important variations, from the standard GPL or BSD Licenses, or, in the case of the larger commercial entities, the Mozilla Public License. These variations can include adding provisions to a license that add to or vary the provisions of the standard GPL style of license, such as adding a choice of law provision or other contractual provision that may be found in standard commercial agreements to "improve" the GPL or to cover gaps that are not covered in the GPL. In some cases, these alterations are not significant enough to take the license outside the Open Source Definition. Changes of this type, however, can result in a license not being considered compatible with the GPL and raise a great deal of controversy in the Open Source

---

[83] *See* Rex Brooks, *Open Source Licenses Overview*, *at* http://www.vrml.org/TaskGroups/vrml-ipr/open_source_overview.html (last modified Oct. 27, 1998) (discussing Mozilla Public License); ROSENBERG, OPEN SOURCE, *supra* note 14, at 102.

[84] ROSENBERG, OPEN SOURCE, *supra* note 14, at 103-05; See *Various Licenses*, *supra* note 55, for a useful list of licenses with summaries and commentaries; Schallop, *supra* note 82, at 248, for a discussion of Sun Microsystem's licenses; Perry, *supra* note 13.

[85] See *Various Licenses*, *supra* note 55, for a useful list of licenses with summaries and commentaries. See also *Licenses*, *supra* note 55, for a complete list of licenses certified as conformant to the Open Source Definition, along with hyperlinks to the licenses.

community.[86] It will always be important to consult with the current version of the Open Source Definition to see which licenses fall within the definition. Licensors are generally well advised to resist the urge to create new licenses.

E. Choosing an Open Source License.

One of the key aspects of the development of the Open Source movement has been the limitation of the number of licenses and the drive to keep these licenses standard. As a general matter, anyone licensing in the Open Source software is going to be pressured by the Open Source community to release software source code under an already existing license.[87] As a practical matter, there is encouragement to use either the GPL, the BSD Licenses, or, in some cases, the LGPL. In other specialized cases, such as the Apache software or Perl, certain licenses and approaches to licensing have been used for a long period of time and it is better to use the licenses that are commonly used in that area of software programming, in part due to the level of comfort with the licenses in the programmer community.[88]

The main consideration people face when considering using different Open Source licenses is whether derivative works are subject to the same license or, in other words, whether derivative works can be made proprietary. Depending on the copyright owner's philosophy regarding that question, a choice of the licenses can usually be made fairly easily. Those comfortable with the copyleft provisions of the GPL will choose the GPL. Otherwise, the BSD Licenses will be preferred.

The choice of an Open Source license becomes much more difficult when there are special issues and special concerns, as was the case with the Netscape. Licensors will probably argue that there are special

---

[86] Some of these alterations, however, will take certain license outside the GPL and these programs would not be considered free software or copyleft software. See *Various Licenses*, *supra* note 55, for a web site, maintained by the Free Software Foundation, that compares the various Open Source licenses in terms of their compliance with the GPL.

[87] *See* Raymond, *supra* note 70; Andrew Leonard, *License to Be Good*, *at* http://salon.com/tech/col/leon/2000/09/22/licenses/print.html (last visited March 29, 2001); Perens, *supra* note 17, at 175; Slashdot, *Interview: Bruce Perens Answers Open Source License Questions*, *at* http://slashdot.org/interviews/99/07/30/2220240.shtml (last visited March 29, 2001).

[88] There are a number of good resources for assistance in comparing and choosing an Open Source license. *See* Perens, *supra* note 17, at 185; Donald K. Rosenberg, *Evaluation of Public Software Licenses*, *at* http://www.stromian.com/Public_Licenses.html (last visited March 29, 2001) (hereinafter Rosenberg, *Evaluation of Public Software*); Hecker, *supra* note 11; Perry, *supra* note 13; Brian Behlendorf, *Open Source as Business Strategy*, in OPEN SOURCES, *supra* note 1, at 149; Brooks, *supra* note 83; ROSENBERG, OPEN SOURCE, *supra* note 14, at 87 ff.; Eric Kidd, A History of "*Open Source,*" *at* http://discuss.userland.com/msgReader$19844#19889 (Aug. 19, 2000).

circumstances in every case involving a commercial program in which a conversion to Open Source is being considered. In general, anyone wanting to release software under an Open Source license must spend a significant period of time becoming very familiar with the licenses, learning their strengths and weaknesses for the particular situation and determining whether release under an existing Open Source license is acceptable or whether to start the process, which should be done in coordination with the Open Source community, of developing a new license that fits the Open Source Definition and is also attractive to programmers. An important aspect of the Open Source movement and Open Source licenses is that since a developer is relying on a community of developers to work on an Open Source project, a developer will want to provide them with incentive to work by giving them an understandable and attractive license to work under. Obviously, Open Source developers are going to be more comfortable working on a standard known license than trying to figure out the nuances of a customized license agreement for a particular project. As a practical matter, only large software players such as IBM and others will have the clout to create new types of licenses.

F. Multiple Licensing.

It is important to note that the copyright owner releasing software under an Open Source license is not granting an exclusive license. It is perfectly permissible for a copyright owner to release source code under an Open Source license and to release a proprietary version of the software under standard commercial licenses in order to make money.[89] Software can also be released under multiple Open Source licenses. In some cases involving the Artistic license, it is fairly common for the release of software under both the Artistic license and the GPL.[90] The difficulty with this approach will center on the ability to use source code that is developed by the Open Source community as part of the proprietary version of the program. This issue is especially thorny, and probably impossible, under the GPL.

IV. Sampling the Legal Issues with Open Source Licenses.

The Open Sources licenses raise a number of important legal issues. There are questions about how the notion of copyleft interrelates with copyright laws and the fair use doctrine. There are important issues involving the individual ownership of source code within a community-based development project and the implications for

[89] *See* Perens, *supra* note 17, at 185.

[90] *Id.*

enforcing the licenses and dealing with potential copyright infringement. There are also important issues about the validity and enforceability of the Open Source licenses and particular terms of the licenses. There have also been questions about interpreting the licenses and the vulnerability of the licenses to software patents. Finally, there is a question about who is in charge of an Open Source program and what happens when the leading figures associated with the Open Source movement are no longer involved in or disappear from the scene.

It is important to note that no cases have been decided that directly interpret any of the Open Source licenses or the particular issues arising under specific licenses.[91] The Free Software Foundation since its inception has not filed any lawsuits relating to the GPL.[92] The issues that arose initially with respect to the BSD Unix licenses were settled out of court.[93] In other words, there are only questions with regard to the legal issues involving Open Sources licenses, not any definitive answers.

An important new development that may have a major effect on the Open Source Licenses is the Uniform Computer Information Transactions Act ("UCITA").[94] UCITA is a controversial new uniform law related to "computer information" that was promulgated after years of effort with the expectation that it will be adopted in all fifty states in the United States.[95] Will Open Sources licenses be covered under UCITA if UCITA is adopted by the state whose law is applicable to a particular Open Source license? The definition of "computer information" in UCITA is certainly broad enough to bring the Open Source licenses under UCITA.[96] If UCITA does cover the Open Source licenses and Open Source software, there are important implications. UCITA is designed as a gap filler

---

[91] *See* Raymond, *supra* note 70 (discussing when you need a lawyer).

[92] *Id.*

[93] McKusick, *supra* note 28, at 44-45

[94] The text of the final version of UCITA and Official Comments may be found at *UCITA Online*, http://www.ucitaonline.com/ucita.html (last modified February 8, 2001). Note that there is a great deal of controversy about UCITA and that all claims about what UCITA says should be checked against the actual text of UCITA. Far too many discussions of UCITA contain unhelpful hyperbole and distortions.

[95] For an extensive discussion of UCITA and its practical implications, see Fendell & Kennedy, *UCITA is Coming!!! Part 1: Practical Analysis for Licensee's Counsel*, COMPUTER LAW., July 2000, at 3 (hereinafter Fendell & Kennedy, *Part 1*); Fendell & Kennedy, *UCITA is Coming!!! Part 2: Practical Analysis for Licensors' Counsel*, COMPUTER LAW., August 2000, at 3 (hereinafter Fendell & Kennedy, *Part 2*).

[96] Uniform Computer Information Transactions Act, Nat'l Conf. Of Commissioners on Uniform State Laws, § 102(a)(10), *available at* http://www.law.upenn.edu/bll/ulc/ucita/ucitaFinal00.htm (September 29, 2000) (hereinafter UCITA).

statute that provides default provisions for software licenses in the event there are gaps or silences.[97] As a result, if UCITA applies to an Open Source license, UCITA may imply terms that are contrary to the intent of the authors of a particular Open Source license. In particular, UCITA might imply terms relating to duration of the license, warranties and other restrictions.[98] On the other hand, UCITA takes a strong approach to and supports the validity of what are known as shrinkwrap licenses, those licenses which are provided without the opportunity to negotiate terms and which are not signed.[99] The Open Source licenses are a classic model of shrink wrap license and UCITA would give support to the notion that this type of license is enforceable and that the Open Source licenses themselves are enforceable. At the time this article is written, UCITA has only been adopted in Maryland and Virginia but the development of UCITA must be watched carefully for its implications for the Open Source licenses, even though Maryland has amended UCITA to create an exemption from the implied warranty of merchantability that should cover most Open Source software.[100]

In the balance of this section, I will make a general survey of some of the legal issues arising with respect to the Open Source licenses.

A. Ownership.

In Open Source development, the general principle is that the author of the code remains the copyright owner and that he or she simply applies the Open Source license to his or her code.[101] To a limited extent, authors using the GPL have assigned their copyright ownership to the Free Software Foundation so that the Free Software Foundation holds all copyright interests in a program and centralizing ownership in a single entity.[102] While there have been some efforts to centralize the ownership of copyrights in project managers, most Open Source programs, however, will ultimately involve code with many copyright owners.

---

[97] *See* Fendell & Kennedy, *Part 1*, *supra* note 95, at 4.

[98] *Id.* at 4-12.

[99] Fendell & Kennedy, *Part 2*, *supra* note 95, at 15-16.

[100] *Id.* at 3. Developments on UCITA, such as the Maryland amendment can be tracked at http://www.ucitaonline.com/whatsnu.html (last visited May 31, 2001).

[101] *GNU License*, *Preamble*, *supra* note 21.

[102] *See* Raymond, *supra* note 70 (discussing copyright status).

Because Open Source software is developed under a community model, there will be questions that arise as to ownership of particular aspects of the source code. While the general picture of Open Source development is one of programmers working on their own in their homes late into the night creating code for specific programs, there is a growing amount of professional development in Open Source and, in either case, there will be questions that arise as to whether some of the software was developed as part of a scope of employment and, therefore, subject to the work made for hire doctrine under copyright law.[103] If the work made for hire doctrine applies, the employer, rather than the individual programmer, would be the owner of the code and the one with authority to grant the license. In addition, a programmer may be subject to employment or development agreements that would not give them the right to contribute source code as part of Open Source development or to apply the Open Source license to the code. As a result, sorting out ownership issues in the event of a dispute could be quite difficult and there will always exist some possibility that some code in an Open Source project may have been licensed by people without the authority to grant the license.

Another aspect of ownership involves "project ownership." Development of the Linux program is a classic example of Open Source development. Linus Torvalds and a group of trusted lieutenants are responsible for developing the official Linux kernel program and determining what code is used in the final version and what is not.[104] Historically, Open Source development has been based on a loose, voluntary and consensus approach to the software projects. It is certainly conceivable that there may be disputes over issues such as software development decisions, version control, release dates, programming standards and other aspects of control of the development of a given program. There have been incidents already under the BSD licenses where programs have "forked," meaning that because of a dispute over what is the official version, alternative programs have been developed.[105] In these cases, there will be questions not only of ownership of the project but who would have the right to enforce the copyright in an infringement action.

B. Enforcement of Copyright and Licenses.

---

[103] *See generally* GigaLaw.Com, *What is a Work Made for Hire?*, *at* http://www.gigalaw.com/articles/loc-2000-02-p1.html (last visited March 29, 2001). *See also* Young, *supra* note 11, at 121-22 (disputing the stereotypes about Open Source programmers and referring to professional software development).

[104] *See* Torvalds, *supra* note 8, at 101; Leonard, *Boot Time*, *supra* note 8; Moody, *supra* note 26.

[105] Rosenberg, *Evaluation of Public Software*, *supra* note 88.

As mentioned in the previous section, any Open Source project may include the work of a large number of copyright owners, each of whom has applied an Open Source license to his or her work. In the case of an infringement or a violation of a particular license, it is unclear who will have the right to enforce the copyright and the license terms. Must all copyright owners be found and joined in the action or will an individual copyright owner be able to bring the action on behalf of all the other owners? The answer to that question is not yet clear. The Open Source licenses that require any distribution to include the names of authors and the dates of revisions will be helpful, but a large, practical questions remain about who may enforce the licenses.[106]

The approach of the Free Software Foundation is important in this respect because by getting authors to agree to assign copyright ownership to the Free Software Foundation, the Free Software Foundation can act itself to enforce licenses and take actions in the event of infringement without the need to track down all copyright owners. The approach of the Free Software Foundation, however, is unusual in the Open Source movement.

C. Commercial Exploitation.

As the Open Source model of development becomes increasingly attractive to commercial software developers, a large number of questions will arise with respect to releasing commercial software as Open Source software. For example, what are the rights of licensees who have licensed the software under a commercial license when the licensor later releases that software as Open Source software? What rights does a commercial developer have to use software contributed by the community as part of the Open Source project that is based on the company's original proprietary software? Will commercial development inevitably lead to a multiplication of the number of Open Source licenses, each of which will have significant variations? Will commercial developers and their lawyers be content to limiting their choices to one of the existing Open Source licenses? Will commercial developers looking for ways to obtain revenue from Open Source products take liberties with the licenses to create new revenues? With the absence of any court decisions to give us any definitive answers, consideration of Open Source will invariably raise many questions and require participants in Open Source efforts to make assessments of likely results and risks.

---

[106] A good discussion of Open Source copyright issues may be found at Potter, *Opening Up to Open Source*, 6 RICH. J.L. & TECH. 24 (2000). The Software Licensing Committee of the American Bar Association's Intellectual Property Section has issued a report called *An Overview of "Open Source" Software Licenses*, *at* http://www.abanet.org/intelprop/opensource.html (last visited March 29, 2001). This report covers a number of copyright ownership issues and notes the potential difficulties in auditing the code base of an Open Source project because of the multiple means of introduction of infringing code.

D. Interpretation Issues.

The Open Source licenses were not written by software licensing lawyers. As a result, they are often quite simple and straightforward, but may not cover certain issues and they may create some ambiguities when later reviewed by lawyers or courts. Even developers trying to comply in every detail with the Open Source licenses will have questions of interpretation about exactly what can and cannot be done under certain licenses.

As suggested earlier in this section, UCITA may play an important role in the interpretation of Open Source licenses, if it is applicable, because it will supply default provisions and other gap fillers in places where the Open Source licenses are silent or ambiguous.[107]

One of the most unique aspects of the Open Source licenses is that they have been consistently interpreted over time by the community, often with the vocal efforts of some of the leading members of the community, such as Richard Stallman.[108] Historically, the sometimes heated exchange of ideas and debate in the community has also been to hammer out interpretation questions and to require modifications to licenses that attempted to vary the terms of standard Open Source licenses or "improve" them. It is clear that Richard Stallman's interpretations, as a founder of the Open Source movement, have played a key role in interpretation of the GPL and there will be some uncertainties as, over time, he and other long-time members of the movement recede from the scene.

As Open Source licensing becomes an increasingly prevalent model, the loose, community, plain language approach to the Open Source licenses may prove inadequate as specific legal issues arise and lawyers and courts begin to take a closer look at the Open Source licenses. While the history and the opinions of the community will continue to matter, it is likely that the judicial interpretation of the Open Source licenses will become increasingly important. In addition, interpretation of copyright laws, and possible changes to copyright laws, may have significant consequences for the interpretation of the Open Source licenses. Given the years of tradition and interpretation in the community itself, it will be interesting to see the weight that courts give to the existing interpretations that are in common currency in the community.

While it has been helpful in the matter of interpretation to keep a limit on the number of Open Source licenses, there will be continuing pressure to increase the number of Open Source licenses and variations within

---

[107] *See* Fendell & Kennedy, *Part 1*, *supra* note 95, at 4.

[108] Raymond, *supra* note 70.

them, especially as commercial software products are converted to the Open Source model. Any increase in the number of Open Source licenses, or permissible variations, will continue to result in interpretation questions and the production of programs that involve a mix of potentially incompatible licenses.

E. Specific License Issues.

The simplified format of the Open Source licenses raise a number of particular interpretation issues. For example, the GPL and other licenses do not provide that the licenses are perpetual. Under UCITA, where the duration of a license is not specified, the term of the license will be considered to be a reasonable duration.[109] This omission may cause problems at a later point when a court determines that an Open Source license with respect to a program has expired because the "reasonable" duration is over. Other "gap filler" or default rules may raise similar types of issues.

Consider this example in the BSD Licenses. The BSD Licenses limit liability for all damages, including direct damages.[110] There is an argument that the limitation of liability for direct damages is so key to the essence of a contract that not allowing someone to obtain direct damages for breach raises a fundamental issue about contract formation. A court might, therefore, determine that no contract exists or that such a provision makes the contract unconscionable or unenforceable.[111] Unlike the Open Source licenses, many commercial software licenses contain a severability clause that would permit a court to throw out or modify only the offensive provision while enforcing the rest of the license. Such a clause is especially important under UCITA and a software licensing lawyer might be tempted to make an "improvement" such as this to the Open Source licenses that would not be allowed by the community.

The fact that the Open Source licenses are not signed and not negotiated may raise questions. These questions relate to the enforceability of the shrinkwrap category of agreements and include whether the licensee has manifested acceptance and consent to the terms, whether all terms are enforceable, or whether the contract is enforceable in its entirety. The answers to these questions can vary under state law. UCITA would probably help the Open Source movement because it should validate shrinkwrap agreements generally, but UCITA raises a

---

[109] UCITA § 308. *See* Fendell & Kennedy, *Part 1*, *supra* note 95, at 8.

[110] *Licenses*, *supra* note 55.

[111] For example, UCITA, echoing most common law, will not allow parties to contract in ways that violate fundamental public policies or result in unconscionable terms. *See* Fendell & Kennedy, *Part 2*, *supra* note 95, at 4.

number of other difficult issues for the Open Source licenses. For example, UCITA is quite specific about the ways that acceptance can be manifested in these contracts.[112] There would also be questions whether acceptance of the Open Source licenses would be appropriately manifested under the Open Source licenses.

As the Preamble to the GPL states, the disclaimer of warranties is an essential element of Open Source licensing.[113] Developers are not willing to contribute to programs if they might incur personal liability or if they are required to provide some form of warranty with respect to their source code, especially if they are contributing the code without compensation. The Open Sources licenses, therefore, provide that the software source code is supplied on an "as is" basis, with no warranties. These disclaimers are designed to avoid personal liability, but they also make sense because the licensee/user has access to the program's source code and the right to modify it to fix any problems in the code. Disclaimer of warranties also is a common practice when software is not priced in a way that will produce the revenues necessary to supply warranty support.

While disclaimers of warranties are common in software licenses, state law, including consumer protection laws, may specify certain requirements or limitations. UCITA also provides some new considerations with respect to warranties. For example, UCITA implements some new implied warranties with respect to computer information and, perhaps more important, sets forth certain specific requirements for the waiver of warranties.[114] In particular, the general statement that computer information is provided "as is" and without warranties does not waive UCITA's implied warranty that information will be delivered free of claims of infringement or misappropriation, since specific language is required to waive this warranty.[115] As a result, if UCITA covers a certain license, there will be a question as to whether the general disclaimer of warranties and a statement that the software is being provided on an "as is" basis in fact disclaim all warranties.[116] The potential applicability of consumer protection laws may also have an impact on warranties for Open Source programs.

---

[112] UCITA § 112. *See* Fendell & Kennedy, *Part 2*, *supra* note 95, at 14-15.

[113] *GNU License*, *Preamble*, *supra* note 21.

[114] UCITA § 406. *See* Fendell & Kennedy, Part 1, *supra* note 95, at 10-11.

[115] UCITA §§ 401(d),(e). *See* Fendell & Kennedy, *Part 1*, *supra* note 95, at 9.

[116] *See* Fendell & Kennedy, *Part 1*, *supra* note 95, at 10-11.

F. License Management.

There are two important issues raised by the Open Source license with respect to the management of licenses of source code used in a program. The first issue was discussed above in the discussion of the GPL. Will downstream licensees of programs be bound by the applicable Open Source license? In the case of the GPL, the copyleft provisions are designed to bind downstream licensees. Since no licensee is signing any of the Open Source license agreements, the question of whether downstream licensees are bound by the terms will be an important one in the case of a dispute. Clearly, the intent in the Open Source movement, especially with the GPL, is that downstream licensees are to be bound by the Open Source license. A related aspect of that question is whether licensees under a commercial license for a program that is later brought into Open Source will receive the Open Source license rights or be bound by the existing commercial license.

The second issue arises because simply keeping track of licenses may become important in Open Source programs if the number of Open Source licenses is allowed to proliferate. Anyone developing an Open Source program must be keenly aware of which licenses apply and the impact of those licenses. In particular, there will be the concern over Section 2(b) of the GPL and whether the mere presence of GPL code in a software program will result in the entire program being considered subject to all terms of the GPL. In general, care should be taken not to mix licenses in a program without a complete understanding of the consequences.

G. Patents.

Those in the Open Source movement have been very concerned about the possibly detrimental effect of software patents since the early days of development of the Open Source licenses. The scope and dimension of software patents has still yet to be resolved.[117] The GPL and some of the commercial Open Source licenses attempt to deal with the possible problem of software patents. The concern would be that a software patent might be claimed with respect to a portion or even all of a software program. The result would be that the Open Source software program would be considered infringing and that use of the program could be enjoined or that a license fee could be charged before the program could be used. Some of the Open Source licenses, such as the GPL, attempt to

[117] *See generally* Jason V. Morgan, *Chaining Open Source Software: The Case Against Software Patents*, *at* http://lpf.ai.mit.edu/Patents/chaining-oss.html (last modified June 11, 1999); Stallman, *supra* note 9, at 67-68.

handle the patent issue by having the licensee agree to any patent or agree not to apply for any software patent which would affect the license.[118]

V. A Few Conclusions.

The period from 1998 to 2001 in particular has drawn a lot of attention to Open Source software. Open source software plays a key role in the infrastructure of the Internet and an increasingly important role in servers and operating systems. It would be difficult to overstate the importance of Open Source software in the development of the Internet. There is increased attention and interest in the Open Source method as a way to develop more stable, less buggy and less expensive software. Reservations about the community style of approach to software are starting to be overcome. Increased publicity and funding for Open Source development has resulted in greater interest in Open Source programs. As a result, increased attention has also been placed on the foundation of the Open Source movement, namely the Open Source licenses. There is a concern, however, that the licenses, which were written in a perhaps a more informal era when things could be done without the involvement of lawyers, will no longer hold up as Open Source becomes increasingly used in commercial environments and the edges of the Open Source licenses are pushed. It is highly unlikely that the Open Source licenses will be able to last another 15 to 20 years without legal challenge as they have done to this point.

The Open Source licenses need to be looked at carefully, in light of both the legal issues that arise under the licenses and in ways that they may be reconciled with the communitarian approach to development. Developments in copyright law and, in particular, UCITA can have important, unintended consequences for the Open Source licenses.

We are likely to see further modifications in the basic Open Source licenses and further rethinking of the philosophies at the foundation of these licenses. In addition, it is fair to expect that most of the licenses will see some discussion and potential revision in order to shore up some of the legal issues that have arisen and may later be raised by UCITA or other new laws. There will be a continuing battle to keep the number of Open Source licenses limited and specific, and a growing pressure from each commercial developer releasing software into Open Source to create its own licenses. The role of the community in the development of this process will remain important, but the role of courts will likely play a larger role in the development of Open Source.

---

[118] *See, e.g.*, Section 8 of the GPL, http://www.gnu.org/copyleft/gpl.html (last visited April 2, 2001).

Perhaps the most important of the notions that arise out of Open Source, the notion of copyleft, using licenses to protect freedom and a public interest approach to intellectual property, will continue to play an important part in the development of intellectual property law. There will be a place at the table not just for the protection of intellectual property rights but for the protection of the rights of the community to use intellectual property developed for the good of the community. The development of the Open Source movement, the Open Source licenses, and the notions of copyleft and software freedom will play a very important role in the development of the Internet and our approach to intellectual property law in the future.