LEGAL ISSUES IN OPEN SOURCE AND FREE SOFTWARE DISTRIBUTION[1]

RAYMOND T. NIMMER

---

[1] This materials have been adapted from Chapter 11 in Raymond T. Nimmer, The Law of Computer Technology (1997, 2005 Supp.).

I. Distinguishing Open, Free and Proprietary Software

Free software and open source (FSOS) software reflects a philosophy about what is the optimal manner for the distribution and development of software.[2] Although it has multiple facets, the philosophy emphasizes making the source code of the software available to transferees of the software and providing the software to others with limited or no restrictions on its use, and with a right to modify and redistribute the software.[3] This philosophy is implemented through licenses. It thus has a uniquely law-related core dimension.

Rather than substantively precise themes, however, FSOS concepts state a fluid philosophy about software development. There are several different ways to look at this philosophy. One is from the perspective of the core adherents – the founders or participants in the central core that zealously follow and promote the philosophy. It is here that the self-definition of FSOS philosophy is most vivid and aggressive. Many discussions of FSOS concentrate on this aspect of FSOS as if it were the only useful way of discussing FSOS, its philosophy and its adherents. But there are other perspectives. One emerges by focusing on the periphery of the so-called "community." As FSOS ideas became a broader part of the software development culture and achieved some market attraction, FSOS attracted a broader and differently motivated membership or following than that with which it began. Many involved in FSOS products today generally support the principles, but do not hold the missionary zeal in detailed adherence that the core group manifests. Additionally, many software purveyors are attracted to FSOS by the marketing opportunities it offers rather than by the core philosophy. If one imagined FSOS as a circle, non-zealot participants and aspirants who seek the label as a matter of convenience or market definition, would form a grey area at the edges of the circle and often extending beyond it, at least as the circle were defined by the core participants.

This is an increasingly heterogeneous community.

A FSOS Compared to "Proprietary" Software

---

[2] See David McGowan, Legal Implications of Open-Source Software, 2001 U. Ill. L. Rev. 241, 268, 274 (2001); Greg R. Vetter, *The Collaborative Integrity of Open Source Software*, 2004 Utah L. Rev. 563.

[3] There is an intentional distinction here between unrestricted use and the "right" to distribute in original or modified form. While the former is typically unrestricted under licenses that conform to FSOS dogma, the latter is often subject to significant restrictions grounded in an effort to advance or protect FSOS philosophy as applied to the software.

What then distinguishes between proprietary and FSOS software?

The answer is simple. The distinction was created by the FSOS community to indicate that those within the community and how they behave are different from the identity and behavior of "*the others*."[4] If you distribute software consistent with FSOS community norms, from the perspective of the community, yours is FSOS software. If not, from that same perspective, yours is proprietary software. Indeed, the leading "free software" organization defines "proprietary" as any software distributed under terms that do not meet the standards of FSOS.[5] The idea that FSOS software is one thing and that "proprietary software" is another resonates within the open software community. But for third-party understanding of FSOS, the definition is not always useful and it is unstable.

B. FSOS compared to public domain software

Free and open source software (FSOS) is not equivalent to public domain software. While FSOS software is not public domain, some FSOS technologists assume that are contributing to the public domain by releasing software under an FSOS license. That is not true. Public domain software is literally "free."[6] It consists of software for which no intellectual property restrictions exist. This might occur because the software code is insufficiently expressive to qualify for copyright protection (even though it may be important in the technology), and too obvious and well known to qualify for either patent or trade secret protection. Alternatively, the software may enter the public domain because the author/inventor of the code releases and abandons all rights to it. A work in the public domain is available for any and all to freely use in any way that they like. They can copy, resell, modify and otherwise deal with it as part of the freely available background that provides grist for new work.[7] Included in this is that others can incorporate public domain material into their own, proprietary works and assert property rights with respect to the

---

[4] http://www.gnu.org/philosophy/free-software-for-freedom.html.

[5] See http://www.gnu.org/philosophy/philosophy.html#TerminologyandDefinitions.

[6] But this does not necessarily mean that it is a desirable goal. See, e.g., Anupam Chander & Madhavi Sunder, The Romance of the Public Domain, 92 Cal. L. Rev. 1331 (2004). Compare

[7] See, e.g., Anupam Chander & Madhavi Sunder, The Romance of the Public Domain, 92 Cal. L. Rev. 1331 (2004); Jessica Litman, The Public Domain, 39 Emory L.J. 965 (1990); Pamela Samulson, Mapping the Digital Public Domain: Threats and Opportunities, 66 Law and Contemp. Pblms. 147 (2003).

work as a whole, keeping that work secret or redistributing it in any way and with any restrictions that they choose.

In contrast, FSOS software is a creature of its licenses and the restrictions attached to them. These restrictions are grounded in intellectual property rights and contractual terms. As one frequently used free software license comments: "To protect your rights, we need to make restrictions…."[8] The fact that some of the restrictions are intended to preserve what FSOS participants believe to be important does not make them less restrictive - they employ intellectual property rights to restrict the conduct of third parties with respect to the licensed software. The motive does not change the legal effect. Truly free public domain software entails no such control.

C. FSOS community and philosophy

There are two broad groups within the FSOS community. One, the "open source" group, was developed several years after the FSOS movement began and was intended to better accommodate commercial interests. The other, which was the first group, is the "free software" movement. The "free software" movement is the more aggressively normative in character. While the two groups are different and often approve (or disapprove) different licenses as consistent with their philosophy, they are not always in opposition. There is significant cross-fertilization and common ground.

What the two groups have in common is a focus on supporting "community" development of software by emphasizing the availability of source code and an absence of license restrictions on modification. While this may be seen as an issue of "freedom", it also entails a belief that it is a more effective method of developing quality software than is a proprietary approach in which one entity controls all rights to improve or adapt the software. Eric Raymond explains this in *The Magic Cauldron*:

> [T]he discussion and advocacy of open-source development … should not be construed as a case that closed-source development is intrinsically wrong, nor as a brief against intellectual-property rights in software, nor as an altruistic appeal to 'share'. While these arguments are still beloved of a vocal minority in the open-source development community, experience . . . has made it clear that they are unnecessary.

[8] Free Software Foundation, GNU General Public License (June 1991), at http://www.fsf.org/copyleft/gpl.html [hereinafter GPL].

> An entirely sufficient case for open-source development rests on its engineering and economic outcomes—better quality, higher reliability, lower costs, and increased choice.[9]

1. Free Software

The leading "free software" site (GNU.org) contains the following "free software" definition:

> Free software is a matter of the users' freedom to run, copy, distribute, study, change and improve the software. More precisely, it refers to four kinds of freedom, for the users of the software:
>
> - The freedom to run the program, for any purpose (freedom 0).
> - The freedom to study how the program works, and adapt it to your needs (freedom 1). Access to the source code is a precondition for this.
> - The freedom to redistribute copies so you can help your neighbor (freedom 2).
> - The freedom to improve the program, and release your improvements to the public, so that the whole community benefits (freedom 3). Access to the source code is a precondition for this.
>
> A program is free software if users have all of these freedoms. … In order for these freedoms to be real, they must be irrevocable as long as you do nothing wrong; if the developer of the software has the power to revoke the license, without your doing anything to give cause, the software is not free.[10]

These premises define the core of a "free software" license. In addition, however, the distinguishing feature of "free software" is its preference for license provisions that are described as "copyleft" in effect and that are

---

[9] www.tuxedo.org/~esr/writings/magic-cauldron/magic-cauldron § 2 [herinafter "Magic Cauldron"].

[10] See http://www.gnu.org/philosophy/free-sw.html "In order for these freedoms to be real, they must be irrevocable as long as you do nothing wrong; if the developer of the software has the power to revoke the license, without your doing anything to give cause, the software is not free"

present in the primary free software license - the GNU General Public License (GPL). The copyleft concept is described in FSF literature as follows:

> Copylefted software is free software whose distribution terms do not let redistributors add any additional restrictions when they redistribute or modify the software. This means that every copy of the software, even if it has been modified, must be free software. In the GNU Project, we copyleft almost all the software we write, because our goal is to give *every* user the freedoms implied by the term ``free software.'' … Copyleft is a general concept; to actually copyleft a program, you need to use a specific set of distribution terms. There are many possible ways to write copyleft distribution terms, so in principle there can be many copyleft free software licenses. However, in actual practice nearly all copylefted software uses the [GNU GPL].[11]

The last sentence of this comment is not correct. Nevertheless, this copyleft attribute of a free software license places restrictions on the redistribution of the software. This is allegedly justified by its goal - to prevent a subsequent party from placing restrictions on the redistribution of the software that are inconsistent with the FSF principles. The general counsel of one FSOS-based company describes licenses that contain these terms as "protective" licenses, while other observers use the description of "viral" or "reciprocal." We can defer coming to grips with the

[11] http://www.gnu.org/licenses/licenses.html#WhatIsCopyleft. The FSF describes the issue as follows: "*Copyleft* is a general method for making a program free software and requiring all modified and extended versions of the program to be free software as well. The simplest way to make a program free is to put it in the public domain, uncopyrighted. This allows people to share the program and their improvements, if they are so minded. But it also allows uncooperative people to convert the program into proprietary software. They can make changes, many or few, and distribute the result as a proprietary product. People who receive the program in that modified form do not have the freedom that the original author gave them; the middleman has stripped it away. In the GNU project, our aim is to give *all* users the freedom to redistribute and change GNU software. If middlemen could strip off the freedom, we might have many users, but those users would not have freedom. So instead of putting GNU software in the public domain, we ``copyleft'' it. Copyleft says that anyone who redistributes the software, with or without changes, must pass along the freedom to further copy and change it. Copyleft guarantees that every user has freedom."

language, but the concept is simple: in order to maintain the "free" nature of the software, this type of provision forces inclusion of the new developments within the terms of the free license.

A distribution format in which terms in earlier licenses dictate terms in subsequent licenses is not unique to FSOS. A common commercial software format also involves a three-party relationship in which the first licensee (a software distributor or OEM) is allowed to copy and transfer software, but conditions are placed on the terms of its retransfer.[12] While FSOS adherents might argue that the intent behind the two systems make them different, that intent is legally irrelevant. Both formats involve a licensor controlling the software distribution chain through licenses that limit a licensee's right to transfer the software to others.

Copyleft or pass-through provisions create a compatibility risk (e.g., a licensee faces two or more licenses that mandate different and incompatible actions). In effect, in such cases it is impossible to comply with both licenses. FSF recognizes this risk, suggesting that, in addition to "free" and "non-free" licensed software, there is a category of software that, while it might meet the definition of "free", is incompatible with the GPL.

> Two different copyleft licenses are usually ``incompatible'', which means it is illegal to merge the code using one license with the code using the other license; therefore, it is good for the community if people use a single copyleft license [the GPL].

This is a plea for uniformity organized around the GPL provisions. That plea has not been answered. Although GPL is the most influential "free" software license, the number of approved and purported open source or free software licenses has multiplied.

2. Open Source Initiative

The second camp is the "open source" community.

Its primary organizational representative is the Open Source Initiative (OSI), which was born in 1998. A group came together with the specific purpose of making open source more palatable to the business community. Stallman and FSF chose not to join the resulting organization

---

[12] See Microsoft Corp. v. Harmony Computers & Elecs., Inc., 846 F. Supp. 208 (EDNY 1994);
Microsoft Corp. v. Grey Computer, 910 F. Supp. 1077 (D. Md. 1995).

- Open Source Initiative (OSI). OSI currently operates as a clearinghouse and as administrator of the certification mark, "OSI Certified"."[13] A list of approved licenses is set out at the OSI website; it currently numbers around fifty different licenses.[14]

As the name implies, the open source initiative focuses on making source code open and available for persons to whom the software is distributed. The principles within this group go further, however. But, in general, "open source" themes emanating from OSI are less normative or socio-engineering in content and more accepting of commercial aspects of intellectual property. OSI defines the issues in the following terms:

> The basic idea behind open source is very simple: When programmers can read, redistribute, and modify the source code for a piece of software, the software evolves. People improve it, people adapt it, people fix bugs. And this can happen at a speed that, if one is used to the slow pace of conventional software development, seems astonishing. We in the open source community have learned that this rapid evolutionary process produces better software than the traditional closed model, in which only a very few programmers can see the source and everybody else must blindly use an opaque block of bits. Open Source Initiative exists to make this case to the commercial world.[15]

This description of FSOS philosophy focuses on the belief that, by making source code available and allowing modifications of the program, the developer may enlist the community of programmers and more rapidly and effectively move the program through an evolutionary process to higher quality. That being said, when one looks at the framework documents, OSI standards are not limited to source code availability and do resemble the core requirements for "free software." OSI's framework document is the so-called Open Source Definition. The Open Source

---

[13] The website comments as to approved licenses: "To identify your software distribution as OSI Certified, you must attach one of the following three notices, unmodified, to the software, as described below. The full notice is: This software is OSI Certified Open Source Software. OSI Certified is a certification mark of the Open Source Initiative. The shorter notice is**:** OSI Certified Open Source Software." Importantly, what is being certified is the license and not the software itself.

[14] See http://opensource.org/licenses/osl-2.0.php.

[15] http://www.opensource.org/.

Definition states ten criteria related to the distribution terms of a license for it to qualify as “open source”:[16]

> 1. Free Redistribution
> 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 a royalty or other fee for such sale.
> 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-publicized means of obtaining the source code for no more than a reasonable reproduction cost -- preferably, downloading via the Internet without charge. …
> 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.
> 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. The license must explicitly permit distribution of software built from modified source code. The license may require derived works to carry a different name or version number from the original software.
> 5. No Discrimination Against Persons or Groups.
> The license must not discriminate against any person or group of persons.
> 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. For example, it may not restrict the program from being used in a business, or from being used for genetic research.
> 7. Distribution of License.

[16] http://www.opensource.org/osd.html.

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.

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. If the program is extracted from that distribution and used or distributed within the terms of the program's license, all parties to whom the program is redistributed should have the same rights as those that are granted in conjunction with the original software distribution.

9. License Must Not Contaminate Other Software.

The license must not place restrictions on other software that is distributed along with the licensed software. For example, the license must not insist that all other programs distributed on the same medium must be open-source software.

10. License Must Be Technology-Neutral

No provision of the license may be predicated on any individual technology or style of interface.[17]

II. License terms: diversity, restrictions and core values

In slightly over two decades, there has been a proliferation of licenses apparently consistent with FSOS values and a greater number that use FSOS approaches, but do not fully conform to FSOS doctrine. As FSOS becomes more commercially viable, this diversity will expand because more and more players, not necessarily wedded to the core philosophies, will bring products to market.

There are ongoing debates about whether particular license terms comply with the core philosophy. Entrepreneurs and companies frequently

[17] The rationale for this rule is stated as: "This provision is aimed specifically at licenses which require an explicit gesture of assent in order to establish a contract between licensor and licensee. Provisions mandating so-called "click-wrap" may conflict with important methods of software distribution such as FTP download, CD-ROM anthologies, and web mirroring; such provisions may also hinder code re-use. Conformant licenses must allow for the possibility that (a) redistribution of the software will take place over non-Web channels that do not support click-wrapping of the download, and that (b) the covered code (or re-used portions of covered code) may run in a non-GUI environment that cannot support popup dialogues."

test the limits of the "pure" FSOS.[18] The result is that some licenses are approved as FSOS compliant by one group, but not by another.

To understand FSOS as a whole, then, it is essential to acknowledge this diversity. While the diversity of license formats is not different from that found in non-FSOS ("proprietary") licensing, it has a more serious bite here because, as we have seen, the license terms used reflect and, even, define the philosophy involved. As with many non-FSOS licensing frameworks, the terms of FSOS licenses are typically contained in non-negotiable standard forms.

This diversity does not mean that there is no countervailing push. Instead, there are several influences that reduce the diversity. One comes from the statements of FOS philosophy that set out guidelines for "acceptable" licenses. Equally important are the several template licenses whose role in FSOS gives them influence as reference points for newer licenses. The most significant are the GPL (and LGPL)[19] and the so-called BSD (Berkeley) license.[20]

If one arrayed all FSOS licenses with all non-FSOS ("proprietary") licenses in terms of their substantive terms, the two would not split neatly into delineated, separate categories as to restrictions on licensees, allocation of risk, or other variables commonly considered to be important in evaluating software and software licenses. Instead, FSOS and the other licenses overlap. Some FSOS licenses are more restrictive than some "proprietary" licenses. For example, the General Public License (GPL) contains numerous provisions that attempt to control distribution of new developments made by the licensee.[21] In contrast, however, many "proprietary" licenses contain lesser restrictions on redistribution or modification of the licensed software than does the GPL. FSOS licenses are consistently less "restrictive" than equivalent other licenses ("proprietary") only if one redefines the word "restrictive" in a manner like

---

[18] What this means, of course, is that FSOS systems are attractive marketing environments. Whereas many participants in FSOS are philosophically linked to its tenets, as the visibility and impact of the systems increases, many others will be attracted to the marketing opportunities and seek to bring with them at least some of the battery of proprietary terms that occur elsewhere in the software world.

[19] Free Software Foundation, GNU Project, GNU General Public License, version 2, 1991, *at* http://www.gnu.org/licenses/gpl.html (last visited December 28, 2004) [hereinafter GPL].

[20] http://opensource.org/licenses/osl-2.0.php.

[21] See §§ 11:29 - 11:34.

the redefinition of “free” in FSOS culture, and concludes that a limitation is not a restriction if it implements FSOS philosophy.

The overlap is accentuated if one compares comparable licenses. Most FSOS licenses cover both end use and distribution. The relevant comparison with the non-FSOS (“proprietary”) world is not a comparison of FSOS with an “end user” license. Given the goals of an end user license, it is predictable that such licenses will not provide the same rights to redistribute as do the melded FSOS licenses. The right to copy and distribute copies is often not relevant to an end user and the licensee would often elect to not negotiate or pay for such rights. On the other hand, most non-FSOS end user licenses contain more warranty protection for an end user than do a typical FSOS license.[22] While warranty issues might be affected by the provision of source code and a right to modify the software, for many end users, neither of these is relevant in fact where the purchaser lacks the capacity or the desire to make its own changes in the software. The comparison between FSOS and non-FSOS licenses thus looks different depending on whether one is considering end use, distribution, or development contexts.

The overlap is even more fully crystallized if one examines not only licenses that the core group defines as fully compliant with FSOS philosophy, but also licenses that come close even though their terms may not conform fully to core doctrine. Especially when we consider “nearly” FSOS licenses, there is a great deal of overlap among FSOS and other (“proprietary”) licenses in the terms used. Table 11.1 suggests some of the issues.

Table 11.1
Comparison of Selected Licenses

Licenses ↓	Terms					
	Cpyright Retained	**Distribution right granted**	**Warranty of quality**	**Warranty or indemnity re non infringe**	**Patent suit terminates license?**	**Source code disclosed**
Pub Domain	No	Yes	No	No	No	Not necessarily

[22] See §§ 6:26 - 6:30.

GPL[23]	Yes	Yes, subject to GPL terms only	No	No	No	Yes
BSD[24]	Yes	Yes, subject to notice and disclaimer	No	No	No	Not necessarily
OSL[25]	Yes	Yes, subject to OSL terms	No	No	Yes Under stated conditions	Yes
Microsoft: end user	Yes	Yes, as complete transfer of rights	Comply with doc; replace or repair	???	No	No
Microsoft: Distribute License	Yes	Yes, subj to terms	Comply docs; replace	Yes	No	Varies based on deal

Table 11.1 does not imply that all non-FSOS ("proprietary") licenses are consistent with FSOS or that FSOS is not distinguishable from non-FSOS licenses. Neither position would be true. But there is overlap of significance.

To say that overlap exists is not to suggest that there are no coherent differences between FSOS software licenses and others. FSOS philosophy has a strong doctrinal base. This base leads to a number of characteristics that identify a "free" or "open" software license, even if they do not always distinguish such licenses from all "proprietary" licenses. FSOS licenses tend to have several provisions that include:

- Source code made available to the licensee
- Licensee allowed to modify the software
- Licensee allowed to redistribute the software
- Licensee allowed to use the software as desired[26]

[23] GPL http://www.gnu.org/licenses/licenses.html#GPL.

[24] Berkeley Software Distribution License (BSD), http://opensource.org/licenses/osl-2.0.php (last visited on December 28, 2004) [hereinafter BSD license].

[25] Open Software License, http://opensource.org/licenses/osl-2.0.php (last visited on December 28, 2004) [hereinafter OSL].

[26] See http://www.gnu.org/philosophy/free-sw.html [visited 12/28/2004].

The intent is to support a process of community or shared development and evolution of software. This is expected to enable achievement of higher quality software more quickly and to establish in a form of free speech community oriented to the software environment. It also tends to drive down the commercial value of software as measured by the price attainable and, thus, to drive up the relative value of associated hardware and services.

IV. Mapping the Legal Context

The reality of the free software and open source (FSOS) communities is that there is a proliferation of licenses whose terms are not always consistent, and that at the periphery of "approved" licenses lie a larger number of FSOS-like licenses. The GPL is the most-used license, but numerous FSOS licenses exist, some of which are certified by the Open Source Initiative or approved by FSF, but others are not. Also, many FSOS licenses have changed over time.

Diversity is an important characteristic of FSOS. To place the issues in proper context, however, we need to recognize that they differ in significance depending on the point of contact that a company or software developer has with FSOS. To simplify, there are three different settings, the issues with respect to which differ markedly. The three categories are:

> **Initial Developers.** The issues for initial developers tend to focus on under what terms the software developer desires to release its own product. In some cases, the release will be commercial, while in others it may not be. It is with respect to such releases that questions about OSI or other approval of the license terms are most significant. The issue will be to what extent can the developer achieve whatever goals it has while approaching or coming within an approved FSOS framework. In some cases, of course, a developer desires to make an FSOS distribution because it believes in and desires to support the tenets of the community, while in others the main motivation may commercial - software in the FSOS framework appeals to a significant market. In some cases, developers whose goals cannot be shaped to meet the FSOS approval standards will simply opt to go without them. In other case, the developer will alter its goals. In still others, a developer might use a dual licensing approach, releasing software under parallel FSOS and "proprietary"

license systems, anticipating that the different frameworks will appeal to different markets.

Secondary Developers or Distributors. Within this category are companies that might use FSOS software in their own products or as their own product, perhaps after having added value. In addition to the question of whether the secondary developer or distributor desires to make its distribution of its product under an FSOS license, an issue concerns to what extent copyleft provisions in an FSOS license "contaminate" or restrict the companies licensing of the software it has created and whether different FSOS terms may create contradictory demands. In practice, companies often encounter this question inadvertently when their employees have without the company's clear authorization, incorporated FSOS code into the developer's new product. A secondary developer is also concerned, of course, about questions related to assurances of quality, non-infringement and other more traditional "legal" issues associated with the FSOS software.

End Users. This category includes companies or individuals that obtain the software for personal or internal use. These licensees are not likely to be concerned about the terms of redistribution since, by definition, they are not primarily contemplating distribution of the code. The end user may, or may not, be concerned about the availability of source code and the right to modify the software, depending in part on whether it has the technological capability to use such code. In contrast, the end user is more likely to be concerned about the extent to which the license provides assurances of quality and other ordinary legal issues. At least to some extent, then, the end user is not concerned about copyleft provisions. In practice, however, questions may arise about what a given FSOS license views as "distribution" and this may create issues for even a company that perceives itself primarily as an end user of FSOS. For example, is it a "distribution" to make the software available for use by third-party service providers by delivering a copy or by allowing remote access?

A. Is an FSOS license an enforceable contract?

One issue about free software and open source (FSOS) licenses concerns whether create an enforceable contract. In one sense, the issue hinges on general contract law under which courts routinely uphold the enforceability of shrinkwrap, ordinary standard form, and online agreements that are very much like the typical FSOS license.[27] However, some FSOS licenses are not structured to establish contractual assent, while others are uncertain in terms of whether a contract is the transactional goal or not. The community opinion is split. Some refer to FSOS licenses as non-contractual, restrictive copyright notices, relying on a view of a license as no more than a unilateral permission to use property, while others simply treat FSOS licenses as would any license agreement.[28]

What is at stake? Treatment as a contract (or as a noncontractual) release affects both the enforceability of the license and the potential remedies that might exist if a licensee fails to comply with the terms of the license.

Enforceability. If FSOS licenses are not contractual, the impact of an FSOS license in law rises or falls on the claim that a partial release of copyright (or other intellectual property rights) can be enforced by the release-recipient (licensee who uses the software), and by the person granting the release (licensor who desires to enforce restrictive terms). As we shall see, the claim to non-contractual enforceability is suspect in some contexts, with Supreme Court and other precedent firmly rejecting the enforceability of a non-contractual, restrictive notice in some settings.[29] Indeed, the enforceability of a software license without a contractual

---

[27] See, e.g., ProCD, Inc. v. Zeidenberg, 86 F.3d 1447 (7th Cir. 1996) (contract enforceable; limits use of database to consumer purposes only); Hill vs. Gateway 2000 Inc., 105 F.3d 1147 (7th Cir. 1997) (contract requiring arbitration enforceable based on use of computer without objecting to contract terms); M.A. Mortenson Co., Inc. v. Timberline Software Corp., 970 P.2d 803 (Wash. 2000) (license enforced when it followed purchase order; "Reasonable minds could not differ concerning a corporation's understanding that use of software is governed by licenses containing multiple terms.").

[28] See generally Greg R. Vetter, *The Collaborative Integrity of Open Source Software*, 2004 Utah L. Rev. 563 (cataloging issues from treating open source licenses as agreements).

[29] See *Bobbs-Merrill Co. v. Strauss et al.*, 210 U.S. 339 (1908) (restrictive resale notice not enforceable when not contractual); *Jazz Photo Corporation et al. v. Dynatec International et al.*, 264 F.3d 1094, 1106 (2001) (single use restriction not enforceable when not contractual).

commitment to it raises interesting questions with respect to "proprietary" licenses: could a software publisher enforce a non-contractual restrictive notice waiving copyright only so long as the uses of the software are non-commercial and do not involve modifications of the code? On the other hand, treating an FSOS license as a contract dependent on contractual rights for enforceability, requires consideration of each transaction to answer whether or not, in that transaction, there were adequate indicia of assent to the contract or, as stated in UCITA, whether the licensee manifested assent to the license after having had an opportunity to review its terms.[30] Because of this contextual aspect of contract-making (a contract is, after all, an agreement between two parties), asking non-contextual questions such as "is the GPL an enforceable contract?" is nonsensical. The question is, rather, whether as used in a particular transaction, did the FSOS license come to represent the terms of contract of the parties?

Remedy. There are also questions of remedy that are tied up in the difference between characterizing the licenses as contractual or non-contractual in orientation. If enforceable, a non-contractual notice would be enforceable by the licensee to prevent infringement claims against it only based on arguments of estoppel or waiver (having caused reliance on the permission, the licensor cannot rescind it), and enforceable by the licensor only through its reserved property rights (the copyright owner never permitted the particular type of use). Neither theory is suited to enforcing affirmative obligations of the licensee as obligations in themselves. Thus, for example, if the license provides for "reciprocity" (e.g., the license terms must be used on any derivative work) and is non-contractual in nature, a licensee's failure to make this condition happen may take the licensee outside the protective umbrella of the license and at risk of an infringement suit, but it would not breach a contract and could not be remedied by other remedies, such as specific performance. Similarly, in a non-contractual framework, even if the notice is enforceable as a restriction, a licensee who does not disclose its own source code could not be sued for specific performance, but only for infringement if the non-disclosure was coupled with conduct that in fact infringed a copyright owner's property rights.

Unfortunately, on these issues, we face a picturesque, but troubling diversity in practice and in the standard form licenses themselves. FSOS licenses exist in two very different contexts. In the one, they are mere standard forms, made available for use by others, but until used, no more

[30] UCITA § 112 (2000 Official Tet).

than statements about what a license might entail. Indeed, most of the more-popular FSOS licenses are drafted to accommodate use by others. In this pure, unused form, the licenses are neither contracts nor releases, nor any other type of legally relevant document. In the second context, the standard form is used (without or without modification) in an actual transaction or release of software. It is only in this context that one can evaluate whether the license became part of a contract or not. To make that evaluation, however, one must also know about the context of the actual use. A standard form FSOS license is not in itself a contract, but it may or may not be used in a transaction in a way that creates a contract.

1. Licenses as non-contractual releases

The idea that an FSOS license (or any other software license) can be treated as an enforceable, non-contractual restrictive notice or conditional release, seems conceptually simple: the licensor holds property rights in the software and conditionally releases some or permits the licensee to use some of those rights as to copies transferred to, or subsequently made by, the transferee. No contractual obligations are required; this is a mere promise to not sue if the licensee uses the property in a particular way.

Perhaps the leading proponent for the view that FSOS licenses are non-contractual is Egan Moglen, General Counsel for FSF. Moglen has been quoted as follows:

> The word 'license' has, and has had for hundreds of years, a specific technical meaning in the law of property. A license is a unilateral permission to use someone else's property. The traditional example given in the first-year law school Property course is an invitation to come to dinner at my house. If, when you cross my threshold, I sue you for trespass, you plead my 'license,' that is, my unilateral permission to enter on and use my property. A contract, on the other hand, is an exchange of obligations, either of promises for promises or of promises of future performance for present performance or payment. The idea that 'licenses' to use patents or copyrights must be contracts is an artifact of twentieth-century practice, in which licensors offered an exchange of promises with users: 'We will give you a copy of our copyrighted work,' in essence, 'if you pay us and promise to enter into certain obligations concerning the work.' With respect to software, those obligations by users include

> promises not to decompile or reverse-engineer the software, and not to transfer the software.[31]

Putting aside the political characterization of proprietary licenses and how they differ from FSOS licenses, the core idea lies in the theory that property rights can be partially waived without requiring a contract to do so.

For FSOS advocates, the argument that an FSOS license is not a contract serves two practical functions. One accommodates the diverse and often-informal manner in how FSOS software is made available to others. While significant companies distribute FSOS software, many distributions occur between private parties, programmers who do not know or comply with the steps needed to establish a contract. If FSOS licenses require a contractual basis to establish limitations or obligations, then in many cases no contract is formed and no limitations established. The assertion that the "licenses" are not contractual in nature circumvents that problem.

A second function of the non-contractual characterization is that eliminates the risk (for the licensee) of being subject to contractual remedies, such as a claim for specific performance of a promise to release its own source code in any redistribution of the software in modified form. Breach of a "noncontractual" license can be enforced only by an infringement action. The General Counsel of Red Hat, a major FSOS company, describes the enforcement issue as follows:

> Keep in mind that, absent the rights granted under the GPL, there are rights under copyright and conditions that copyright imposes. A person obtaining GPL software has no rights under copyright law to make more than a single backup copy and to use that one copy. They have no right of redistribution. They have no right to make derivative works. Those are rights that can only be granted by the copyright owner, and they are granted by the GPL. If the GPL were held unenforceable, all of those rights would disappear.[32]

---

[31] Pamela Jones, "The GPL is a License, not a Contract, http://lwn.net/articles/61292. See also Eben Moglen, Quesitoning SCO: A Hard Look at Nebulous Claims, www.osdl.org/docs/osdl_eben_moglen_position_paper.pdf.

[32] Mark Webbink, Licensing and Open Source, 1, 10, in Brian Fitzgerald & Graham Bassett, Legal Issues Relating to Free and Open source Software (2003)

This ignores the various ways in which rights are created; it ignores estoppel, waiver, implied-in-fact contract, and implied in law contracts for example. But the conceptual model does have value in understanding the FSOS argument.

Moglen's comments relate to the GPL (and LGPL), the most controversial FSOS license because the copyleft provisions threaten trade secrets of companies that might use GPL software in their own products. Moglen can speak with respect to the intent of FSF as the copyright owner with respect to its use of the GPL, although as we see later, the standard form language in GPL is not fully consistent with a non-contractual model. Moglen cannot speak for the intent of subsequent licensors or for companies that use the GPL for their own products unrelated to an FSF product. Because of the importance of GPL, we discuss its approach on this issue in greater detail below.

Other FSOS standard forms explicitly contemplate a contractual relationship. For example, the Open Software License, drafted by the general counsel of the OSI, states:

> by exercising any of the rights granted to You in Section 1 herein, You indicate Your acceptance of this License and all of its terms and conditions. This License shall terminate immediately and you may no longer exercise any of the rights granted to You by this License upon Your failure to honor the proviso in Section 1(c) herein.
>
> This License represents the complete agreement concerning the subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable.[33]

This language cannot control whether an individual transaction supports that there was assent to the license as a contract, but it clearly contemplates use as part of an agreement.

On the other hand, the language of at least some FSOS licenses does focus a non-contractual concept. For example, the BSD license, a frequently used license template, uses a copyright notice framework. The license reads as follows:

[33] OSL §§ 8, 13 (Version 2.1, 2004).

Copyright (c) <YEAR>, <OWNER>
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of the <ORGANIZATION> nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.[34]

This language does not propose a contractual relationship, but merely states conditions under which distribution and use can occur. Putting aside questions about what each of the conditions means, one can see the argument that all that is happening in this document is a conditional approval of another person's taking specific, designated steps with respect to the software in compliance with several limited conditions.

But there is more to be considered here than simply that the language in a standard form is not promissory in nature. In practice, many contractual documents use non-promissory language, with the contractual nature of the relationship coming from the context and the behavior of the parties. But more important, the language of a standard form divorced of

[34] Berkeley Software Distribution License (BSD), http://opensource.org/licenses/osl-2.0.php (last visited on December 28, 2004) [hereinafter BSD license].

information about how the parties use it tells very little about the effect of the form in a particular transaction between two parties. It is that use that determines whether a contract is sought and created. For example, with respect to the BSD license, consider a transaction in which the transferor offers to deliver a copy of the software for "$1,000, subject to the terms of the BSD license." The licensee accepts this offer. Under modern contract law, the license terms are clearly part of a contract between the parties.

In addition, while BSD is a very simple form, the purported conditions go beyond simply conditioning how the property can be used. They require affirmative conduct by the user and disclaim warranties and remedies that exist in law in various types of transaction. Unless the license has a contractual basis, the disclaimer and remedy limitation are likely to be ineffective. They are not conditions on use, but disclaimers of promissory obligations. Indeed, it is arguable that both Article 2 and UCITA require that a disclaimer of implied warranties must be part of an agreement to be effective, and certain a disclaimer of the promissory obligations found in express warranties must likewise relate to a contractual relationship.

2. Enforceability of noncontractual conditions

The theory underlying the claim that a waiver or notice is enforceable without a contract is that the owner of property can conditionally waive enforcement of its property rights without obtaining a contractual assent to the limiting terms. If I allow you to enter my living room, but not my bedroom, I do not need a contract to enforce the latter restriction, at least in theory.

The theory has superficial appeal, but a weakness is that there is little case law support for it in intellectual property law and significant case law that refuses to enforce restrictive aspect of such releases at least in some cases.[35]

---

[35] See *Bobbs-Merrill Co. v. Strauss et al.*, 210 U.S. 339 (1908) (restrictive resale notice not enforceable when not contractual); *Jazz Photo Corporation et al. v. Dynatec International et al.*, 264 F.3d 1094, 1106 (2001) (single use restriction not enforceable when not contractual); *Hewlett-Packard Co. v. Repeat-O-Type Stencil Mfg. Corp.,* 123 F.3d 1445, 1453 (Fed. Cir. 1997) ("a seller's intent, unless embodied in an enforceable contract, does not create a limitation on the right of a purchaser to use, sell, or modify a patented product so long as a reconstruction of the patented combination is avoided."); Fischer, Mark A. "Reserving All Rights Beyond Copyright: Nonstatutory Restrictive Notices" 34 *Journal of the Copyright Society*, 249, 251 (1987) ("Despite authority for the proposition that such non-statutory restrictive notices are unenforceable, the

In thinking about the non-contractual paradigm, it is useful to distinguish cases based on the effect a noncontractual release allegedly has on rights and privileges of the other party that would exist if the noncontractual terms were not present and the transfer of software nevertheless occurred. A noncontractual release or restrictive notice might seek either to expand or to limit the rights the transferee would have absent "waiver." While noncontractual releases that *expand* rights of the other party are likely to be enforceable when detrimental reliance occurs, releases that purport to withdraw rights will often be unenforceable. Indeed, in the latter case, the term "release" or "waiver" does not apply, except as a case of doublespeak, since the "release" restricts, not relinquishes rights.

Copyright notices that expand the rights of the party receiving them beyond what would otherwise exist are likely to be enforceable as to the *expanded rights* when the recipient reasonably relies on the noncontractual grant, at least until the grant is withdrawn and the effects of the detrimental reliance are alleviated as to future conduct. Enforceability in this sense, relates to the *licensee* enforcing it. Consider, for example:

Illustration 11.2

> A obtains a copy of a protected work in a transaction conducted in a manner that, without more, would allow A to make a single copy of the software into its computer. B, the licensor and copyright owner, places a notice on that work allowing any person in possession of a copy to make and distribute up to ten additional copies. Party A relies on this notice and does so; it is sued.

In this case, Party A should win under either implied license or estoppel theory. It reasonably relied on the noncontractual notice giving it authority to do what it would not otherwise be able to do and, having so relied,

---

attempts persist."). Compare *Micro Star v. Formgen Inc.,* 154 F.3d 1107(9th Cir., 1998) ("*But abandoning some rights is not the same as abandoning all rights, and FromGen never overtly abandoned its rights to profit commercially from new levels. Indeed, FormGen warned players* not to distribute the levels commercially and has actively enforced that limitation by bringing suits such as this one."); *Pubic Affairs Assoc. v. Rickover*, 284 F.2d 262 (D.C. Cir. 1960) (A notice did serve to preserve copyright rights in the creator, even when such a notice granted limited permission to use the work.).

should be protected from liability caused by B's change of mind.[36] Of course, if the notice were contractual in the transaction, Party B cannot contractually withdraw the permission. On the other hand, if Party A makes fifty copies of the work and is sued for making the additional copies, the limiting portion of the noncontractual is also likely to be enforced. Simply allowing some uses of property does not release all property rights.[37] Likewise, there is no basis for estoppel with reference to copies beyond the permitted ten since there could be no reasonable reliance that these were also allowed.[38]

There is a wealth of case law holding that *noncontractual* notices cannot limit the transferee's use of a protected work by reducing rights or privileges it would otherwise have in the transaction as it actually occurred. Most of the cases involve sales of copies or machines purportedly subject to restrictions. In *Bobbs-Merrill*,[39] the Supreme Court established the doctrine of first sale in copyright law, emphasizing that a use restriction on a book was non-contractual and that, as a result, it was ineffective to reduce the contractual rights. More recent cases reach the same result. In *Jazz-Photo*,[40] the Court of Appeals for the Federal Circuit refused to enforce a single use "restriction" on the sale of patent cameras where the use restriction was not part of the contract. There is no relevant, contrary authority where the purported limitation would reduce the rights that the transferee would otherwise enjoy as a result of the transaction in which it engaged. On the other hand, modern cases and most of the earlier cases indicate that use restrictions that are part of the contract are enforceable contracts.[41]

---

[36] The key in such cases is that enforceability benefits the recipient of the waiver and does not place new burdens, restrictions, or liability on the recipient. Having stated my view of the most likely outcome of such cases, however, I need to point out that there is a dearth of case law.

[37] See *Micro Star v. Formgen Inc.,* 154 F.3d 1107(9th Cir., 1998). See also Scanlon v. Kessler, 11 F. Supp. 2d 444 (SDNY 1998) (photographer voluntarily provided corporation with thousands of photographs for website, but withdrew consent; subsequent use on site was infringing).

[38] For an analogy in a contractual license, consider the conclusion that use outside the scope of a license is infringement regardless of whether the license remains effective and in force. See Sun Microsystems, Inc. v. Microsoft Corp., 188 F.3d 1115 (9th Cir. 1999).

[39] Bobbs-Merrill Co. v. Strauss et al., 210 U.S. 339 (1908).

[40] Jazz Photo Corporation v. Dynatec Int'l, 264 F.3d 1094, 1106 (2001).

[41] See, e.g., ProCD, Inc. v. Zeidenberg, 86 F.3d 1447 (7th Cir. 1996).

Most of the cases that have tested the enforceability of restricted waivers have involved transactions that would otherwise be a first sale (or exhaustion under patent law). But the principle applies more broadly. For example, if I contract to give you the right to make 1,000 copies, I cannot unilaterally alter that contractual grant by placing a restrictive notice on a copy that purports to preclude your making more than ten copies of the work. If the law or a contract gives the transferee rights, the licensor needs a contract to exclude those rights. Thus if the law gives you an implied warranty, a copyright owner needs a contract, not a notice or waiver, to get rid of it. And remember, the "contract" is not the standard form, but the legal effect of the entire interaction and bargain of the parties.

A noncontractual notice or waiver cannot impose restrictive rules outside the parties' relationship as established by a contractual transaction. This conclusion is buttressed by the historical reluctance of courts to permit anti-competitive leveraging of a copyright or a patent to cover conduct or products outside the scope of the right (e.g., warranty disclaimer, choice of law, coverage of licensee's own work).[42]

Indeed, to fully understand the problem, consider whether a non-FSOS ("proprietary") publisher could release software under a non-contractual license that excludes warranties, limits use, requires redistribution only under stated conditions, and imposes all of those conditions not only on an initial transferee, but on all subsequent transferees. If FSOS licenses can do so non-contractually, then other licensors can also do so. Yet, few non-FSOS publishers argue for that position and if they did so, they would likely fail.

3. FSOS licenses as contracts

The language of many FSOS standard forms contemplates contractual arrangements. This is indicated various ways, including in provisions referring to the effect of the "agreement" and provisions referring to what conduct constitutes acceptance of the license. The following language from several licenses suggests the pattern:

> **Academic Free License, Version 2.1[43]**
> If You distribute copies of the Original Work or a Derivative Work, You must make a reasonable effort under the circumstances to obtain the express assent of recipients to the terms of this License. Nothing else but this License (or

[42] See, e.g., Alcatel USA, Inc. v. DGI Technologies, Inc., 166 F3d 772 (5th Cir. 1999).
[43] See also Open Software License, ver. 2.1 (2004), drafted by same author.

another written agreement between Licensor and You) grants You permission to create Derivative Works based upon the Original Work or to exercise any of the rights granted in Section 1 herein … Therefore, by exercising any of the rights granted to You in Section 1 herein, You indicate Your acceptance of this License and all of its terms and conditions.

Apple Public Source License, Version 2.0

"Please read this License carefully before downloading this software. By downloading or using this software, you are agreeing to be bound by the terms of this License. If you do not or cannot agree to the terms of this License, please do not download or use the software."

Common Public License Version 1.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS COMMON PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

Eclipse Public License - v 1.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

Not surprisingly, such language is often coupled with other terms that do not expressly refer to promissory obligations, but that is a common contract drafting approach.

If one asks whether an FSOS license forms part of a contract, in addition to the language of the standard form, the context of use of the form controls and the standards for forming an FSOS contract are the same as for any other contractual license or, more correctly, for any standard form used in a contractual relationship. The issues are whether, *in the particular case*, the parties entered into a contractual relationship that incorporated the terms of the standard form FSOS license as part of that contract.[44] On this point, both the *Restatement (Second) of Contracts*

[44] Notice that the issues must be treated as analytically separate, although in many cases, assent to the standard form terms (or a lack of assent) will coincide with assent to a contract as a whole. The point is that the contractual enforceability of the FSOS, standard form is a question about what terms are part

and UCITA outline similar standards: for the form to become part of the contract and define its terms, the licensee must manifest assent to the FSOS license after having had an opportunity to review its terms.[45] Manifestation of assent is described in UCITA in the following terms, which correspond to the *Restatement* and, increasingly, are followed in reported case law:

> (a) A person manifests assent to a record or term if the person, acting with knowledge of, or after having an opportunity to review the record or term or a copy of it:
>
> (1) authenticates [signs] the record or term with intent to adopt or accept it; or
>
> (2) intentionally engages in conduct or makes statements with reason to know that the other party or its electronic agent may infer from the conduct or statement that the person assents to the record or term.
>
> (b) An electronic agent manifests assent to a record or term if, after having an opportunity to review it, the electronic agent:
>
> (1) authenticates the record or term; or
>
> (2) engages in operations that in the circumstances indicate acceptance of the record or term.[46]

The assent can be by conduct, words, or signature. But applying the test requires an inquiry into the circumstances of the transaction.

Case law routinely enforces shrinkwrap and online licenses properly presented to obtain assent. Depending on how they are used, FSOS standard forms have all of the characteristics of such licenses. They are subject to the same rules. As in shrinkwrap contexts generally, enforceability requires that the presentation and reaction to the FSOS license establish assent to it. In general, this requires that the licensee have reason to know that terms are or will be proposed, have an opportunity to say no after it has a chance to review the terms, and nevertheless take steps that it has reason to know will signal assent to the other party.

---

of the agreement, rather than solely whether any agreement occurred. This point is made clear in both the UCITA and the Restatement (Second) of Contracts approach to standard form contracting.

[45] UCITA § 208 (2000 Official Text); *Restatement (Second) of Contracts* § 211.

[46] UCITA § 112 (2000 Official Text). See also *Restatement (Second) of Contracts* § 19.

The same can be said with respect to standard forms used online. The framework can be seen in comparing two decisions of the Court of Appeals for the Second Circuit in an online context. In the first case, the court in *Specht v. Netscape Communications Corp.*[47] held that the terms of a license relating to the free Netscape software did not become part of a contract. While the court did not contest the idea that online assent creates enforceable contract terms, the license in this case was not presented in a manner such that the transferee would have reason to know that by downloading the software it would be considered to have assented to the license terms. The terms were referenced below the first screen and, since the software was "free" (e.g., available for no cost), there was no notice that terms were being proposed or assented to by downloading. As UCITA would provide, there was no reason to know that downloading was assent. The court commented:

> We are not persuaded that a reasonably prudent offeree … would have known of the existence of license terms. Plaintiffs were responding to an offer that did not carry an immediately visible notice of the existence of license terms or require unambiguous manifestation of assent to those terms. … Moreover, the fact that, given the position of the scroll bar on their computer screens, plaintiffs may have been aware that an unexplored portion of the Netscape webpage remained below the download button does not mean that they reasonably should have concluded that this portion contained a notice of license terms. … Plaintiffs testified, and defendants did not refute, that plaintiffs were in fact unaware that defendants intended to attach license terms to the use of SmartDownload. We conclude that in circumstances such as these, where consumers are urged to download free software at the immediate click of a button, a reference to the existence of license terms on a submerged screen is not sufficient to place consumers on inquiry or constructive notice of those terms. The SmartDownload webpage screen was "printed in such a manner that it tended to conceal the fact that it was an express acceptance of [Netscape's] rules and regulations." … When products are "free" and users are invited to download them in the absence of reasonably conspicuous notice that they are about to bind

[47] Specht v. Netscape Communications Corp., 306 F.3d 17 (2d Cir.2002).

> themselves to contract terms, the transactional circumstances cannot be fully analogized to those in the paper world of arm's-length bargaining.[48]

Specht did not hold that explicit notice of what specific steps constitute assent must be provided. The standards are far more flexible than that. This was underscored by the Second Circuit's analysis in *Register.com v. Verio, Inc.*[49] In that case, the question was whether a company that downloaded from the Register.com site factual data about domain name registrations was bound by the terms of the contract which limited its right to use the information. The court held that the license terms were enforceable even though the licensee was never asked to click and indicate express assent to the terms. Here, the site clearly indicated that downloading the data was assent to terms. Equally important, the particular licensee used the site frequently and, according to the court, clearly had notice that terms were being proposed for the privilege to do so. The court commented:

> Verio, however, cannot avail itself of the reasoning of *Specht.* In *Specht,* the users in whose favor we decided visited Netscape's web site one time to download its software. Netscape's posting of its terms did not compel the conclusion that its downloaders took the software subject to those terms.… There was no basis for imputing to the downloaders of Netscape's software knowledge of the terms on which the software was offered. This case is crucially different. Verio visited Register's computers daily to access WHOIS data and each day saw the terms of Register's offer; Verio admitted that, in entering Register's computers to get the data, it was fully aware of the terms on which Register offered the access. … We recognize that [many] contract offers on the Internet often require the offeree to click on an "I agree" icon. And no doubt, in many circumstances, such a statement … is essential to the formation of a contract. But not in all circumstances. [New] commerce on the Internet … has not fundamentally changed the principles of contract. It is standard contract doctrine that when a benefit is offered subject to stated conditions, and the offeree makes a

[48] Id.

[49] Register.com v. Verio, Inc., 356 F.3d 393 (2nd Cir. 2004).

> decision to take the benefit with knowledge of the terms of the offer, the taking constitutes an acceptance of the terms, which accordingly become binding on the offeree.[50]

As this suggests, there are many ways in which assent to the license terms can be found, but there must be some indication that the assent was voluntary and made with reason to know that terms were being proposed. How this concept applies to FSOS licenses, of course, varies from case to case and context to context.

A further source of difficulty in a contractual analysis of FSOS licenses lies in "with whom the license terms are created?" Many FSOS licenses, including the GPL and LGPL, are written for use by multiple parties in a license chain. As a result, references in such licenses to "the licensee", "Licensor", "you", and the like are often ambiguous. This is compounded by the fact that some licenses either require that the licensee pass-through the license terms when it distributes the software, or that the terms automatically do so (in effect, running with the software). For example, the GPL states:

> Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. . . . You are not responsible for enforcing compliance by third parties to this License.[51]

The result is that, in cases where a transferee does not deal directly with the company or person that originated the software, there are potentially three different ways in which the license terms may directly impact the licensee.[52] Figure 11.2 illustrates the issue from the transferee's perspective:

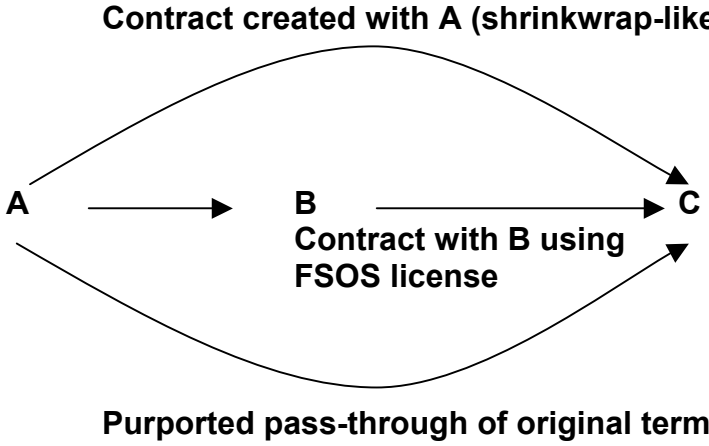
FIGURE 11.1
Three Party Relationship

---

[50] Id.

[51] GPL § 4.

[52] *Register.com* and *Specht*, of course, are illustrations of a two-party transaction in which the sole questions was whether the conduct of the transferee in context created contract terms between it and its immediate transferor. See Specht v. Netscape Communications Corp., 306 F.3d 17 (2d Cir.2002).



The possibilities are:

1. A contract formed between the remote parties (A and C in Figure 11.2) based on C's assent to terms proposed to it by A, the originator of the licensing chain. If present, this would resemble ordinary three-party transactions in shrinkwrap and similar transactions in the software industry.
2. A contract formed between the immediate parties (B and C in Figure 11.2) in which the parties use the FSOS license to establish terms of their direct, contractual relationship.
3. A pass through of terms established by A in the initial transaction and arguable enforceable whether or not the remote party (C in Figure 11.2) assents to those terms.

Of these three, the most problematic as a contract law matter is the third - the automatic pass through of terms. In the absence of assent by C, merely saying that the terms pass through to it does not form a contractual relationship between A and C with respect to restrictive terms. That being said, however, the fact that there may be no pass-through contract does not end the matter. An analogy to commercial licensing suggests that terms which, in effect, condition the distribution options of the party in the middle (e.g., Party B above) may have impact remote parties under intellectual property law if the intermediate party did not become the owner of the copy that it then distributed and made the distribution in violation of the "license" restrictions. Several courts have

held that an upstream limitation on how software can be distributed, if breached by the distributor, exposes the remote party to liability for infringement.[53] This is not a contractual argument, but an intellectual property law argument; the remote party is engaging in copying, redistribution, or other conduct that is not authorized by the copyright holder. The idea of bona fide purchase, which is important in ordinary goods, does not apply to the world of intellectual property rights.[54] The pedigree and provenance of the copy delivered has significance to the transferee.

What has just been said applies to code directly released by the copyright owner under an FSOS license. If the software received by last licensee includes elements produced by parties other than that original copyright owner, defining the last licensee's rights is more complicated. In a three-party chain of transactions, some code may be written and copyrighted by Party A, but additional code is written and added by Party B. The next person in the chain receives a composite that includes copyrightable material from both A and B. There are various relationships in law as to copyright ownership that might arise between Party A (the original author) and Party B (the modifier), but assume that Party B created a derivative work. This means that B owns the copyright in this work as to the new material.[55] For this new material, then, there cannot be pass-through rights from the remote copyright owner, nor is a contract with that copyright owner necessary or sufficient to allow use of the material. The only source of rights for the Transferee (Party C) comes from the relationship between Party B (the derivative work owner) and Party C (B's transferee).

4. GPL (and LGPL) as a contract or a restrictive notice

As we have seen, Eben Moglen, General Counsel of FSF, argues that the GPL is a non-contractual release or license, rather than a

---

[53] See Microsoft Corp. v. Harmony Computers & Elecs., Inc., 846 F. Supp. 208 (EDNY 1994);
Microsoft Corp. v. Grey Computer, 910 F. Supp. 1077 (D. Md. 1995); Novell, Inc. v. Unicom Sales, Inc., 2004 WL 1839117 (ND Cal. 2004).

[54] Novell, Inc. v. Unicom Sales, Inc., 2004 WL 1839117 (ND Cal. 2004).

[55] Section 103 of the Copyright Act states: "The copyright in a compilation or derivative work extends only to the material contributed by the author of such work, as distinguished from the preexisting material employed in the work, and does not imply any exclusive right in the preexisting material." 17 U.S.C. § 103(b).

contract.[56] In contrast, many in the FSOS communities view it as a contract.[57]

When the terms of GPL are examined, the only reasonable conclusion is that GPL contains language consistent both with an attempted contract and with a non-contractual release (restrictive notice). The relationship created, however, depends not on the terms of the standard form, but on how the form is used in a transaction. The GPL's status is not determined by the intent of the authors of the standard form, but by the objectively manifested intent of the parties to each actual transaction. The test of what relationship (if any) is created concerns how the standard form is actually used in a transaction. In this regard, the language of the form can be useful as an indicator, as can the statements of leaders in the development of GPL itself. But these at most contribute to the general background associated with use of the standard form. In addition, with respect to the contract/ noncontractual issue, there are divergent views. In any event, the belief of the author of the *standard form* cannot be binding on the users of the form in a particular transaction.

The language of the GPL and the LGPL on this issue creates an almost exquisitely contradictory document. GPL contains language that is consistent with intent to form a contract and language that is consistent with a document that does not rely on contract terms. On balance, however, contractual language predominates. In addition, the length and complexity of GPL and LGPL, including the creation of what appear to be affirmative obligations on licensees, argues that more than a non-contractual waiver is intended. Indeed, if "free software" requires an irrevocable license, as the FSF website suggests, irrevocability can be obtained for GPL only if the form as used represents a contractual obligation.[58]

The clearest part of GPL that indicates a possible, noncontractual basis is the following language in Section 6:

> "Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. … You

---

[56] Pamela Jones, "The GPL is a License, not a Contract, http://lwn.net/articles/61292. See also Eben Moglen, Quesitoning SCO: A Hard Look at Nebulous Claims, www.osdl.org/docs/osdl_eben_moglen_position_paper.pdf.

[57] Rod Dixon, Open Source Software Law (2004).

[58] See http://www.gnu.org/philosophy/free-sw.html [visited 12/28/2004].

> are not responsible for enforcing compliance by third parties to this License."[59]

This language contemplates an attempted pass-through license from the copyright owner to a remote transferee; on the face of the license, it cannot be based on a contractual relationship unless there is assent to the terms of *that* license by the remote transferee as to a relationship with the copyright owner.[60] The reference to an automatic license is far different from language found in shrinkwrap licenses in which the copyright owner offers a license to the remote licensee for its acceptance or refusal. If the language reflects the practice in a particular transaction, this GPL pass-through is noncontractual in nature.[61]

GPL does not, however, rely solely on this pass-through concept. For example, Section 1(b) states that the licensee must "cause any work that [it] distributes or publishes [containing the Program] to be licensed … under the terms of this License."[62] This contemplates some action between the parties to the subsequent transfer that "causes" the license terms to apply to that transaction. It does not occur automatically. Similarly, Section 4, in providing that rights under the license terminate if the licensee copies, modifies or distributes other than in compliance with the license, goes further to state that "parties who have received copies, or rights, from you under this License will not have their licenses

---

[59] GPL § 6.

[60] A semantic problem with GPL and other licenses that are drafted for use in multiple, different transactional contexts involving remote and immediate parties is that it uses the word "you" for every transferee, including both the "first transferee" (the person who dealt directly with the copyright owner) and the "remote transferee" (a later person who did not deal directly with the copyright owner). As a result, person in the chain of distribution may be referred to as "you" in the license in reference to several entirely different statuses within the same document.

[61] In this regard, recall the earlier discussion of the complexity added when one presumes there are more than one copyright owner because persons have independently contributed copyrightable material, creating a derivative work. See later discussion at ---.

[62] GPL § 1.(b). The reference to causing the "terms of this License" to apply could be referring to the GPL as a written template (e.g., B licenses this under the GPL template) or to the actual agreement between the transferor and the copyright owner by which the first transferee obtained rights in the software. If the latter case governs, GPL is contemplating a partial assignment of the license to subsequent parties. As with any other contract, the effectiveness of an assignment agreement would be judged under contract law.

terminated so long as such parties remain in full compliance." As this suggests, those subsequent transferees have received rights, presumably by contract, from the licensee whose rights may not be terminated.

Even more explicitly, GPL § 5 states:

> You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this license. Therefore, by modifying or distributing the Program … you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.[63]

This language, which is of a type used in contractual arrangements such as in commercial shrinkwrap licenses, clearly contemplates creation of a contractual relationship.[64] "Acceptance" of a license is not relevant to a non-contractual restriction. The restriction simply exists. If the model followed by the GPL were non-contractual in nature, the language referring to "acceptance" would not be used. Instead, the copyright owner might simply state: "You may copy, modify, and distribute this software, but only in compliance with the following conditions." Instead language of "acceptance" is present. This suggests a document that contemplates creating a contractual relationship if accepted by the other party.

The Preamble to GPL refers to the creation of obligations consistent with a contractual relationship. The Preamble states: "To protect your rights, we need to make restrictions … These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it." The reference to "responsibilities" indicates contractual obligations undertaken by accepting the terms of the license and creating a contract.

As is true with contractual licenses of various types, GPL has language that conditions the scope of the license grant and provides that actions outside these conditions are outside the scope of the rights granted without using promissory language. The relevant language in GPL reads in part as follows:

---

[63] GPL § 5. This same language appears in Section 9 of the LGPL.

[64] It should be noted that the language and open-ended context of GPL and LGPL create an ambiguity here about whether the contract is between the copyright owner and the remote transferee, or between the two parties to the immediate transaction, or both types of contract.

> **GPL § 1:** You may copy and distribute verbatim copies of the Program's source code as you receive it, … provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty … and give any other recipients of the Program a copy of this License along with the Program. …
> **GPL § 2**: You may modify your copy or copies of the Program or any portion of it … and copy and distribute such modifications or work under the terms of Section 1 … provided that you also meet all of these conditions:
>
> a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
>
> b) You must cause any work that you distribute or publish … to be licensed as a whole at no charge to all third parties under the terms of this License.
>
> **GPL 4:** You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. …

This language sets out what contract law would describe as language of condition,[65] or what intellectual property lawyers would describe as defining the scope of a license.[66] Under either view, the language is consistent with a contractual relationship.

The fact that GPL may contemplate creation of a contractual relationship does not mean that it will always be used in this manner in practice or that the contractual relationship will be established. To form an enforceable contract that includes the terms of the GPL (or LGP), the licensee must assent to the terms after having had an opportunity to review them.[67] We previously discussed this requirement, but the fundamental concept focuses on the context of the transaction and whether the licensee had reason to know that it was agreeing to the terms of the GPL license. Modern contract law allows assent by conduct. Indeed, case law consistently enforces online and shrinkwrap license terms when 1) the terms of the license were made available in a manner giving the licensee an opportunity to review them, and 2) conduct after

[65] Murray on Contracts § 89 (2001).
[66] See Sun Microsystems, Inc. v. Microsoft Corp., 188 F.3d 1115 (9th Cir. 1999).
[67] See UCITA § 208 (2000 Official Text); Restatement (Second) of Contracts § 211.

that opportunity indicates assent. The key to whether the terms become part of a contract often lies in their manner of presentation and whether the circumstances give reason to know that assent to terms is sought.

As a result, insofar as GPL terms as a contract are concerned, it is not useful to ask whether the software has been "GPL'd." That phrase means that a person has attempted to apply the GPL to his/her software. That decision is not dispositive on whether a contract was made and assent obtained. The legal question is whether the license is presented in a manner that creates a contract and that the transferee's actions constitute assent to that contract

There are three general observations about this issue:

> First, open source and free software practices show a wide variety in how the software is distributed, and at least in some cases, it is likely that the terms do not become part of an enforceable contract.
>
> Second, judging the enforceability of GPL in any context must take into account the extent to which the terms of GPL might become part of the agreement because, as occurred in Register.com,[68] the licensee had reason to know the impact of its conduct, in this case as part of the usage of trade or prior interactions with the licensor. A contract consists of an agreement that is enforceable in law, including course of dealing, usage of trade and the like.[69] Cases on licensing information have held that trade use can add terms to a contract.[70]
>
> Third, to be an enforceable contract, the parties' relationship must meet other conditions of contract law, such as consideration, mutuality and the fact that any assent to terms come from a person with authority to bind (or estop) the organization against which contract terms are asserted.

If GPL does not establish an enforceable contract, its terms may still have an impact under non-contractual analyses. Unlike in a contractual relationship, however, non-contractual relationships do not impose affirmative obligations on the licensee (e.g., there is no affirmative

---

[68] Register.com v. Verio, Inc., 356 F.3d 393 (2nd Cir. 2004).
[69] UCC § 1-201; UCITA § 102.
[70] Puget Sound Financial. L.L.C. v. Unisearch, Inc. , 47 P3d 940 (Wash. 2002).

obligation to obtain warranty disclaimers or to disclose source code). As we have seen, the enforceability of substantive, but noncontractual, restrictive notices or waivers is suspect under copyright law. Because of this, the best way to understand GPL as a noncontractual document is to start by asking what relationship exists if the GPL had no legal effect in the transaction. The answer sets a baseline. We can then evaluate the difference in the relationship that would result if GPL as used forms a contract, a noncontractual waiver, or an estoppel.

B. Ownership issues in free and open source

The free software and open source (FSOS) environment relies on the ability to control copyrights (and patents) associated with the software. This leads inevitably to the question of who "owns" software that, in many cases, has received creative contributions from a large community of participants, often acting independently of each other. Under copyright law, control vests in the person who owns the copyright, either as the original owner or as the recipient of a transfer of ownership. But in the open-ended world of FSOS software, who is the owner of the copyright?

The place to start in is with fundamental copyright law principles. In the absence of an employee work for hire context, copyright law assumes that ownership vests in the person who created the expressive material (the "author"). The Copyright Act allows transfers of ownership, but FSOS licenses typically do not provide for such transfers.[71] Indeed, one would expect that such a contract term would be inconsistent with community view of appropriate licensing.[72] Instead of attempting to bring ownership back up the chain to one central clearinghouse by a series of agreed transfers to upstream parties, licenses such as the GPL and LGPL rely primarily on forcing FSOS limitations downwards by placing restrictions on each subsequent licensees, although they do maintain the possibility of

[71] There are exceptions. Official versions of some FSOS software products are maintained by a central source or group. In such cases, while free to make and distribute its own variations, a licensee that desires its code to become part of the "official" version must submit it to that supervisory group.

[72] Indeed, while the grant-back approach allows a licensor to retain control of and access to developments associated with its technology, in the U.S. there is limited case law that suggests that in some circumstances such arrangements may constitute misuse of intellectual property. S*ee* Dratler, Licensing of Intellectual Property § 7.09 (2000); Raymond T. Nimmer & Jeff Dodd, Modern Licensing Law §§ 13:23 - 13:21 (West, 2005).

voluntary contributions to the "official" or centrally maintained version of the software.[73]

Because FSOS licenses generally do not establish mandatory grant back arrangements that pass ownership upstream, for any software where community modifications actually occur and those modifications entail expression that qualifies for copyright status, copyright law is likely to create split or joint ownership. This has potentially tremendous consequences that are a direct product of the mismatch between the horizontal image of software development espoused by open source and the authorship-based system set out in the Copyright Act. That is, the circumstance is not a flaw in the open-source system that might be corrected, but a by-product of the system itself.

The fact that copyright ownership vests in the creative author creates problems even in two-party transactions, but the problems multiply when one begins to consider that for some FSOS products, the number of independent contributors is quite large. As one author commented: "Given the growing expanse of users working collaboratively, today's Linux is less a seamless piece of coding than a tapestry of hundreds of hackers' contributions."[74] These hackers hold the copyright in their own works, even though they may not intend to assert that right against anyone.[75] The mere fact that an author may have incorporated their changes into a program and allowed it to be transferred does not mean that the author waived or abandoned the copyright.[76] But, of course, the fact that they own it means that others, including FSF, do not own the

---

[73] But see Apple Public License, ver. 2.0, § 3: "In consideration of, and as a condition to, the licenses granted to You [Licensee] under this License, You hereby grant to any person or entity receiving or distributing Covered Code under this License a non-exclusive, royalty-free, perpetual, irrevocable license, under Your Applicable Patent Rights and other intellectual property rights (other than patent) owned or controlled by You, to use, reproduce, display, perform, modify, sublicense, distribute and Externally Deploy Your Modifications of the same scope and extent as Apple's licenses under Sections 2.1 and 2.2 above."

[74] Glyn Moody, *The Greatest OS That (N)ever Was,* www.wired.com/wired/5.08/linux_pr.html (August 1997).

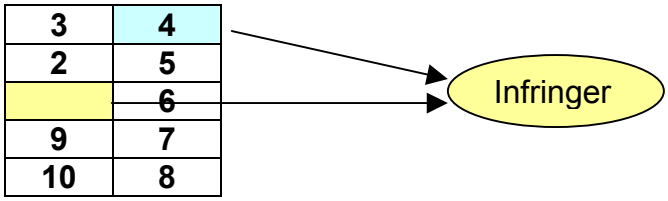
[75] This assumes, of course, that they have not separately transferred the copyright back up to a central group monitoring or controlling the "official" version of the software.

[76] See, e.g., *Storm Impact, Inc. v. Software of the Month Club*, 13 F.Supp.2d 782 (ND Ill. 1998); *Scanlon v. Kessler*, 11 F.Supp.2d 444 (S.D.N.Y. 1998) (Photographer had voluntarily provided the corporation with thousands of photographs for website; withdrawal of consent and subsequent use was infringing)..

copyright to the contributions and that these others cannot, therefore, enforce that copyright in order to protect the open-source environment or for any other purpose.

To obtain some sense of the difficulty that arises with multiple contributions and ownership, consider Figure 11.4. In that figure, the group of boxes constitutes an FSOS program, while each individual box is the contribution of a separate programmer, identified by his or her number.

Figure 11.4



In this illustration, we might view person 1 as the original developer (e.g., the FSF), while the others are persons who contributed expressive code to the work. Unless a concept of joint ownership applies, each individual controls the right to enforce the copyright in the work, by claiming infringement or by licensing use of the person's own work. That is, Person 1 cannot sue for infringement of the code contributed by Person 4. Nor can Person 1 license Person 4's work to third parties (assuming that 4's contribution occurred after 1) without Person 4's permission. In an infringement suit by Person 1, infringement would be found only if the alleged infringing work infringed that part of the program authored or owned by Person 1.

In practice, the situation of ownership of FSOS works may be even more complicated since contributions to FSOS software are often not discrete, expressive code added to a program, but modifications in existing code. It is also made more complex by the nature of copyright concepts associated with multi-party ownership.[77] If the creative input of several parties is substantial, the Copyright Act recognizes potential dual authorship as an alternative to vesting ownership in only one party. But three different concepts might apply: (1) a collective work or compilation, (2) a derivative work, or (3) a joint work. The last category equates in common parlance with the idea of coauthorship, but the other two do not. The legal consequences of the three formats vary.

[77] See discussion in Chapter 4.

In a joint work, for example, the coauthors are joint tenants with coequal rights to license the entire work on a nonexclusive basis subject to obligations to account to the other authors for profits. In effect, neither joint author holds fully exclusive rights in the work.[78] The downstream joint owner, if such exists, is separately capable of granting licenses to the copyrighted work. On the other hand, as a co-owner of the entire work, that joint author can also enforce the copyright even as to expression written by the other coauthor.

In contrast, a derivative work builds on a prior work (which may be owned by another party) and creates a new, separate work, owned by the new author but incorporating elements from the earlier work with permission of the first author.[79] The party executing the new work holds the copyright in the new elements in its own right and a right to control the whole as a unified, copyrightable product. Section 103 of the Copyright Act makes clear that, although the copyright owner need not grant the right to prepare derivative works, once she does so the owner of the copyright to the derivative work is the author of that derivative work. This copyright in the derivative work is independent of the underlying copyright, but extends only to the portions of the whole work created by the derivative work author.

A similar result occurs for a collective work or compilation. In both, the authorship involves selecting and/or arranging material into a new, combined work. The collective work author holds the copyright to the expressive compilation - the selection and arrangement. In contrast, if the work collects works of authorship by other authors with their permission, those original authors hold the copyright to their original works and control of those works other than as part of the authorized collection or revision thereof.[80]

Copyright Section 103(b) provides that the:

---

[78] See, e.g., Devereaux v. Colvin, 844 F. Supp. 1508 (MD Fla. 1994) (alleged owner of program copyright not entitled to preliminary injunction to prevent infringement; owner failed to establish likelihood of prevailing on the merits of where defendant owned a partial interest in at least one of the computer programs alleged to have been infringed); Anderson Consulting v. American Management Sys., Inc., No. 95 Civ. 5428 (KTD), 1995 WL 510042 (SDNY Aug. 28, 1995) (as joint owner, plaintiff lacks exclusive rights in work).

[79] Id. See Weissmann v. Freeman, 868 F2d 1313 (2d Cir. 1989). See generally David Nimmer, supra, § 3.01. Of course, if the use of elements from the earlier work is not authorized, it violates the exclusive right to make derivative works. 17 USC § 106.

[80] 17 USC § 201(c). See New York Times Co., Inc. v. Tasini, 533 US 483 (2001).

> copyright in a compilation or derivative work extends only to the material contributed by the author of such work, as distinguished from the preexisting material employed in the work, and does not imply an exclusive right in the preexisting material. The copyright in such work is independent of, and does not affect or enlarge the scope, duration, ownership, or subsistence of any copyright protection in the pre-existing material.[81]

Section 103 thus parses out the ownership rights. Essentially, those who prepare compilations (which includes collective works) or derivative works can only lay claim to their own material, and cannot claim any "exclusive right" in the pre-existing material. But equally as clearly, one of the collateral effects of Section 103(b) is the right for the derivative or collective work author to own a copyright independent of the author of the underlying copyright.

In general, the distinctions among these three types of works and the different form of ownership they create hinge on the intent of the parties, the nature of the contributions, and how the contributions are joined together. No co-ownership arises by operation of law unless the parties claiming ownership contributed expression to the eventual work product.[82] Given an adequate contribution, however, the major factor distinguishing among the three types of authorship centers on intent, both with respect to how the contributions are to be merged and on the issue of sharing ownership. In a collective work, the parties do not intend that the contributions be merged to the point of being indistinguishable. In a collective work and in a derivative work, no intent exists to jointly develop; the creator of the derivative works alone on an existing product. A joint work, in contrast, is "a work prepared by two or more authors with the intention that their contributions be merged into inseparable or interdependent parts of a unitary whole."[83] To constitute a joint work, the

---

[81] 17 U.S.C. § 103(b) (1999).

[82] See Childress v. Taylor, 945 F2d 500 (2d Cir. 1991); Erickson v. Trinity Theatre, Inc., 13 F3d 1061 (7th Cir. 1994).

[83] 17 U.S.C. § 101. See Respect, Inc. v. Committee on Status of Women, 815 F. Supp. 1112 (ND Ill. 1993) (textbooks are not joint works, because joint authorship results only if principal or dominant author intends to share authorship; here, intent of dominant author was to claim sole authorship).

parties must intend that their contributions merge into inseparable or interdependent parts of a whole.[84]

When one juxtaposes this statutory framework with the development model advocated by free and open source software, the potential complexity and the degree of fact-intensive analysis required to ascertain ownership in any case where community-wide development actually occurs (as compared to being potentially available) should be quite apparent. To "simplify" the analysis, three arrangements might exist with respect to each contributor to an FSOS product that is not a work for hire:

- If the contribution does not involve material that would be treated as expression under copyright law, the contributor has no ownership claim. The contributed material may be non-expressive because the code is straightforward, well-known or too limited in creativity, or because the contribution is too abstract and is a mere idea, rather than expression.[85]
- If the contribution is expression, but the intent of the parties was not to merge the new expression into "inseparable or interdependent parts of a unitary whole",[86] the result is either a derivative work or a compilation (collective work). In either case, the contributor holds a copyright to the new work that extends only to its new expression in that work. In a derivative work, this new expression will typically be in

[84] Joint authors separately own the work with rights to use and license it. See Words & Data, Inc. v. GTE Communications Servs., Inc., 765 F. Supp. 570 (WD Mo. 1991). See discussion in Raymond T. Nimmer & Jeff Dodd, Modern Licensing Law § 5:11 (West, 2005); Raymond T. Nimmer, The Law of Computer Technology § 4:15 -4:22 (West, 1997, 2004).

[85] See Ashton Tate Corp. v. Ross, 916 F2d 516 (9th Cir. 1990); MGB Homes, Inc. v. Ameron Homes, Inc., 903 F2d 1486 (11th Cir. 1990). As one court noted, "[to] be an author, one must supply more than mere direction or ideas: one must translate an idea into a fixed, tangible expression entitled to copyright protection.. . . The supplier of an idea is no more an author of a program than is the supplier of the disk on which the program is stored." SOS, Inc., 886 F2d at 1081. Compare Words & Data, Inc. v. GTE Communications Servs., Inc., 765 F. Supp. 570 (WD Mo. 1991) (coauthor of forms regardless of direct expression being contributed).

[86] 17 U.S.C. § 101 (definition of "joint work").

the new code added, while in a collective or compiled work, the expression may be new code, but may also be in the selection or arrangement of parts to create the new work.

- If the contribution creates new expression and the authors prepared the work with the intent of merging the contributions into inseparable or interdependent parts of a unitary whole, the work is a joint work. Each joint author owns the work as a coauthor, with the right to license it. Unlike with respect to the derivative work or compilation, the contributor's ownership is not limited to the new expression it created.

In many FSOS settings, the normal expectation would seem to be that new contributions create derivative or collective works, rather than joint authorship. This is suggested by the structure of at least some of the licenses. Thus, while the GPL and the LGPL contemplate that the licensee may make changes in the program and that community development will occur, they refer to derivative works and collective works resulting from such changes and do not express any intent to co-author with subsequent contributors. The licenses do not expressly contemplate joint authorship or co-ownership. This is important because the existence of a joint work depends on the intent of the authors at the time the contributions occur. If the primary or first author does not manifest intent to become a coauthor, then the later, authorized modifications seem more likely to be separately authored derivatives of the first work.

This being said, in copyright law, the idea of joint authorship does not per se require that coauthors work at the same time on the work. It is sufficient that they both intend at the time that they create their expression that their "contributions [will] be merged into inseparable or interdependent parts of a unitary whole." The concept of joint authorship, thus, contrasts not only to a case where one party controls the creative process (a sole authored work), but also to situations involving separate works or expression with no intent to make a merger of expression, but to be blended into a "collective work," which is defined in copyright law as "a number of contributions, constituting separate and independent works in themselves, [that] are assembled into a collective whole."[87] In the absence of an express contract, if each author contributed expression to a program, the distinction between joint and collective works turns on

[87] 17 U.S.C. § 101.

elusive questions of intent. A leading treatise suggested the following analysis:

> [Joint] ownership would seem to be justified on two alternative bases. First where the respective contributions of each author are inseparable in the sense that they are not separately identifiable the only workable solution is to regard each author as the joint owner.... A second basis . . . occurs where the respective contributions are interdependent. Here, although . . . separately identifiable, each may be said to be written pursuant to an implied . . . agreement that the product of the several contributions will be jointly regarded as an indivisible work.[88]

One can readily visualize cases where this analysis will yield close questions of fact about the parties' intent, especially if interdependence entails some form of implied agreement to coauthor. After all, one purpose in releasing an FSOS product is to encourage the supposedly more effective application of a community of developers in the development of a final, effective software product.

C. Viral impact: unrestricted vs. copyleft software

The idea of "copyleft" license provisions is a characteristic part of at least a segment of the free software and open source software (FSOS) community.[89] Indeed, it is common in FSOS to view restrictive copyleft provisions as the hallmark of truly "free" software, as the community defines that term. From the perspective of nonbelievers, however, copyleft is the most controversial feature of free software and open source software because it affects the user's rights with respect to the FSOS software and may impact the user's control of software written entirely by it when used in conjunction with the FSOS software.

In a stunning example of double-speak, the Preamble to the GPL describes the reason for such provisions in the GPL in terms of protecting the licensee's rights:

> To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain

[88] David Nimmer, supra, § 6.02.
[89] See § 11:9.

> responsibilities for you if you distribute copies of the software, or if you modify it. For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. … We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.[90]

Under this view, the restrictive features of the GPL place responsibilities on the licensee; in order to protect the licensee's rights, these restrictions take away the right to transfer in whatever manner the licensee desires and, indeed, in the GPL require that any distribution be subject solely to the rights and restrictions set out in the GPL.

While proponents refer to such restrictions as creating "free" software, protecting rights, persons affected or potentially affected by the terms tend to refer to the risk of "viral" license terms that reach out to infect their own, separately developed software and of improper market leverage and misuse of copyright to control the works of other people.

If we put the labels aside, the issue should be framed in terms of to what extent FSOS software licenses have, or seek to have, an impact on transactions beyond the specific agreement or transaction with the first licensee. How do such terms affect subsequent licenses, remote from the first FSOS licensor. Framing the discussion in these terms, there are at least three different categories into which FSOS license fall. These are:

- **Pure unrestricted licenses.** These licenses grant rights to the licensee, but do not place significant substantive restrictions on the use or redistribution terms that the licensee can establish with subsequent transferees. The licenses may impose notice or similar conditions on a transfer, but the substantive terms of that subsequent license are not controlled (e.g., "you may copy or modify this software and distribute such copies or modifications in any manner you choose."). This type of FSOS license comes the closest to the idea of public domain, truly free software.
- **Pass-through licenses.** These licenses require that some or all of the substantive terms in the license to the first licensee carry forward to a subsequent transferee as

[90] GNU GPL, ver. 2.0, Preamble.

to the original licensed subject matter. The pass through can be sought as an automatic result said to occur directly between the original licensor and the subsequent licensee, or it can be described as a condition on the first licensee's right to distribute the licensed software (e.g., "you may transfer this software to third parties only if the third party receives all of the rights created under this license as to the original licensed subject matter"). In effect, this type of license tells the licensee that, as to the original licensed code, the licensee cannot alter (reduce) the terms of the original license when it conveys that software to another person. A pass-through term does not control the license terms that might be used with respect to new code, even when associated with the originally licensed code.

- **Expansive licenses.** These licenses permit modification and transfer of the licensed software, but require that in any transfer the terms of the initial license must be applied, in whole or in part, to the original subject matter and to the new material that a licensee might add to the software. In effect, this license reaches into work created by the licensee and requires that a distribution of this work along with all or part or the original software conform to the terms of the FSOS license. The licensee's work is thus caught in the FSOS framework. The extent to which viral effects are present vary depending on what type of new work is covered and what relationship it must have with the original code.

1. Pure Unrestricted Licenses

A "pure unrestricted" license is the least restrictive of the FSOS licenses in terms of limitations on the licensee. Under this framework, the licensor makes the software available (by contract or otherwise) under conditions that allow the transferee to copy, modify and distribute the software. The license may specify minor conditions for the right to transfer the software (such as retaining any copyright notice), but it does not mandate substantive terms that must be used in any transfer of the software to a third party. Those terms are be determined by the agreement between the transferor and transferee in that transaction. The licensor typically discloses or otherwise makes available the source code to the software in the initial transaction and the terms of the license do not

place limitations on its use. On the other hand, the license does not require that the source code be given to a subsequent transferee.

Many licenses approved by the Open Source Initiative fall within this "pure unrestricted" category. This includes most licenses based on the so-called BSD license model, a framework that relies on a form of copyright notice to give transferees rights to copy, modify and distribute, but does not refer to mandated substantive terms of any redistribution.[91] To many, this is the most "free" and "open" of the FSOS types. It leaves the licensee freedom to make decisions about how to structure subsequent transactions as to the original or modified code. It relies on the market and the continuing availability of the original software from the original licensor to "protect" the licensing model. The BSD license template provides:

> Copyright (c) <YEAR>, <OWNER>
> All rights reserved.
> Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
>
> - Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
> - Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
> - Neither the name of the <ORGANIZATION> nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.
>
> THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO ….[92]

Notably, except for the disclaimer language, the language of this license does not extend to control terms in any subsequent distribution of the software, with or without modification, by the licensee.

---

[91] See generally BSD License; MIT License, available at www/opensource.org.
[92] Bsd License, available at www/opensource.org.

While this framework optimizes licensee discretion (freedom?), it allows licensees to place restrictive terms around their versions of the software in transactions with third parties. This is not problem in cases where the distribution is of verbatim copies of the original software - a licensee objecting to the new terms can go to the original source for the software free of restrictions. It does create a greater risk in cases in which the first licensee adds significant value to the original software. In such cases, especially as the number and value of the additions increase, the effect may be to practically transform the original, free software into a non-FSOS ("proprietary") framework as a practical matter. At least this is what free software advocates who favor strong viral or pass-through terms describe to as a rationale for using different licenses to control subsequent license terms.[93] A fully unrestricted license leaves that protection to the market place and the self-determination of the licensee.

2. Pass Through License Terms

A "pass through" license exercises control of substantive terms of a re-distribution of the software to third parties, but it limits that control to terms concerning the program as delivered to the licensee that later redistributed the software. The license terms do not extend to code or other expression independently created by the licensee.

> The rationale is straightforward. The licensor opted to make its software freely available for redistribution and modification. Persons desiring that software on that basis can go directly to the original licensor. Pass through terms ensure that the same rights in the same software are obtained from third parties who redistribute it. Pass through provisions are most common with respect to rights to distribute the original software in verbatim copies.

3. Expansive Licenses

Viral or expansive licenses implement copyleft concepts of the Free Software Foundation (FSF). These terms not only pass through as to the original software, they place restrictions that dictate that modified versions of the software must be made subject to the original license, even as to new material created by the licensee. This potential impact on new software code created by the original licensee is what generates the term "viral" license - the terms may reach beyond their original host.

---

[93] GPL, version 2.0, Preamble.

The purpose, as expressed in FSF literature, is to prevent the process of modification from effectively taking a previously "free" software package out from the free environment by cabining it in with proprietary modifications. While that is clearly part of the motivation, however, these terms are also aimed at expanding the range of free or open source software by bringing in software developments that might otherwise be kept "proprietary".

For the licensee, however, a viral provision diminishes the licensee-developer's control over its own work and, when it requires source code disclosure, threatens to destroy potentially valuable trade secrets embodied in source code. The extent of risk, of course, depends on the scope of the viral impact in the FSOS license and on the licensee's intended use of the FSOS software. Unfortunately, as we discuss later, the most widely used, viral FSOS license (the GPL) is a study in uncertainty on this issue, creating significant risk for users of GPL-covered software who intend to redistribute code in their own products.

Viral or expansive effects are typically reserved for circumstances in which the licensee distributes modified code and do not apply to modifications made and retained for personal use. The term "distribute" or its equivalent as used in FSOS licenses, however, is not fully defined. For example, does the use of a third party to outsource or to repair and maintain a program constitute distribution when that person does so by remote access to the software or by receiving a copy from the client's system?

A common approach in FSOS licenses is to require that the terms of the original license be extended to cover distribution of "derivative works." For example the Open Software license provides:

> Licensor hereby grants You a world-wide, royalty-free, nonexclusive, perpetual, sublicenseable license to do the following:
>
> a) to reproduce the Original Work in copies;
>
> b) to prepare derivative works ("Derivative Works") based upon the Original Work;
>
> c) to distribute copies of the Original Work and Derivative Works to the public, with the proviso that copies of Original Work or Derivative Works that You distribute shall be licensed under the Open Software License; …[94]

[94] Open Software License, Version 2.1, § 1.

The term “derivative work” is used in the Copyright Act. In that statute, it refers to:

> A work based upon one or more preexisting works, such as a translation, musical arrangement, dramatization, fictionalization, motion picture version, sound recording, art reproduction, abridgment, condensation, or any other form in which a work may be recast, transformed or adapted. A work consisting of editorial revisions, annotations, elaborations, or other modifications which, as a whole, represent an original work of authorship, is a “derivative work.”[95]

Case law establishes that, to be a derivative work, the author of the modifications must have added new expressive material to the preexisting work.[96] Beyond that, however, what constitutes a derivative work is a wavy and imprecise one.

Some licenses attempt to draw a firmer line. The Mozilla Public License, for example, provides:

> The Modifications which You create or to which You contribute are governed by the terms of this License…. The Source Code version of Covered Code may be distributed only under the terms of this License … and You must include a copy of this License with every copy of the Source Code You distribute. You may not offer or impose any terms on any Source Code version that alters or restricts the applicable version of this License or the recipients' rights hereunder... Any Modification which You create or to which You contribute must be made available in Source Code form under the terms of this License ….
>
> [For purposes of this agreement, modification means] means any addition to or deletion from the substance or structure of either the Original Code or any previous Modifications. When Covered Code is released as a series of files, a Modification is:

---

[95] 17 U.S.C. § 101.

[96] See Raymond T. Nimmer, The Law of Computer Technology ch. 4 (3d ed. 1997, 2005 Supp.)..

A. Any addition to or deletion from the contents of a file containing Original Code or previous Modifications.

B. Any new file that contains any part of the Original Code or previous Modifications.[97]

Whether this definition is materially more understandable than a general reference to "derivative work" might be questioned. In both cases, however, the apparent intent is to focus on expressive modifications of the original software, rather than on all related or collateral code or systems that might be designed to *work with* the original, FSOS software.

Beyond contract interpretation issues, the viral terms of licenses present legal issues about enforceability. One basis for challenge lies in the concept of misuse.[98] This theory precludes enforcement of intellectual property rights that have been misused by the rights owner, including at least in some cases by attempting to leverage the rights into control of products or work that falls outside the scope of the licensor's property interest. Clearly, the viral terms in an FSOS license do this, but whether courts would hold that they are justifiable by the nature of the context and the purpose behind the license terms remains to be seen.

4. General Public License (GPL)

Because it is the most widely used free software license and has the most aggressive terms limiting the licensee's options in retransferring the software, the GNU General Public License (GPL) merits specific attention with respect to viral and pass-through license terms. The GPL is widely used in part because of its association with Linux. On the other hand, GPL is the least clearly drafted of the major FSOS licenses and the most problematic for licensees. Published "interpretations" of it by the Free Software Foundation (FSF) often reflect more of the commitment of that organization to the idea of free software than supportable positions clearly stated in the license and the law surrounding it.

GPL has a schizophrenic approach as to whether it is grounded in contractual or non-contractual terms. That uncertainty carries over to the GPL provisions that limit what terms can be used by the licensee in redistributing the original or a modified version of the software. The

[97] Mozilla Public License 1.1, available at www/opensource.org

[98] See Alcatel USA, Inc. v. DGI Technologies, Inc., 166 F3d 772 (5th Cir. 1999).

uncertainty contributes to uneasiness in many quarters about using software covered by this license in proximity to other important software.[99]

In understanding the impact of GPL on a licensee's options in redistributing software, we need to distinguish between distribution of verbatim copies of GPL-covered software and cases of distribution of modified software or of GPL-covered software combined with other products that are not otherwise covered by GPL.

a. Pass through terms. The GPL treatment of distribution of verbatim copies by a licensee is a relatively straightforward application of a pass-through license approach. GPL permits copies to be made and distributed, but requires that the distribution be according to GPL terms, including terms relating to source code disclosure. This result comes from a confluence of three GPL sections, as follows:

> **GPL Section 1:** "You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you 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 this License along with the Program."
> **GPL Section 3:** "You may copy and distribute the Program … in object code or executable form under the terms of [Section 1] above provided that you also do one of the following:
>
> a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
>
> b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under

[99] See Gregg Vetter, "Infectious" Open Source Software: Spreading Incentives or Promoting Resistance?, -- Rutgers L.J. -- (2005).

> the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
> c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)"[100]
>
> **GPL Section 6**. Each time you redistribute the Program … the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein.[101]

Taken together, these provisions require that verbatim copies made and distributed under GPL be passed onward with the same terms as those under which they were received, coupled with the disclosures and notices outlined in sections 1 and 3. Section 6 makes clear the intent that GPL terms relating to use (e.g., copying, distribution, disclosure, etc.) are both the minimum and the maximum permitted.

This approach can be described as "pass through" in part because of the language in Section 6. Under that language, the "original licensor" grants a license of its copyright automatically to any recipient of the Program. There are two issues here. The first lies in identifying who is the "original licensor" of the Program. One theory would be that this refers to the entity or person that originally introduced the original version of program under GPL. This interpretation works well with respect to unmodified copies since it means that the intermediate distributors (licensees) are not involved, a reasonable structure since the "original" party holds the copyright and other intellectual property rights. As we will see in the next section, however, this interpretation creates complex issues when software is transferred as modified. In such cases, the "Program" is defined as any program to which GPL is applied.[102] Is the

---

[100] GPL Section 3 defines source code as "the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable." GPL § 3.

[101] General Public License (GPL), version 2.0, §§ 1, 3, 6.

[102] General Public License (GPL), version 2.0, § 0.

"original licensor" the very first licensor or the most recent licensor who modified the software creating a new program? What are the circumstances of intermediate licensees who modify and transfer on the software? Is everyone the "original licensor" or just the very first or the very last?

The second issue concerns whether this is a contractual license. The language does not appear to require or ask for the creation of a contractual relationship. As a result, the legal effect may depend on the development of case law on restrictive notices or conditional waivers. Ultimately, whether the pass-through concept has legal effect in limiting the terms of redistribution may depend on how upstream transaction occurred, including whether they involved sales of copies.

b. New Material. The second situation in which GPL attempts to control the terms under which the licensee redistributes software involves cases where the software covered by GPL was modified or added to by the licensee, or where the GPL software is distributed as part of a collection including software created by the licensee or third parties. The issues here truly implicate the potential viral impact of GPL. While one can speak about passing through GPL terms for the original GPL software, in cases of modifications or collections, there is a further issue about the extent to which the GPL terms attach to the new material created by the licensee or other third party. In working through GPL provisions on this issue, there are numerous points of uncertainty, ranging from questions about whether the GPL terms are contractual in nature (affecting the remedy and enforceability) to questions about what certain words used in the GPL mean when juxtaposed to software technology and marketing practices.

GPL Section 2 provides that, subject to conditions laid out in the license, a licensee may modify the copies of the GPL-covered program and copy and distribute those modified copies. For present purposes, we will put to one side the question of what the term "modify" means. The license, however, distinguishes between the "Program" and a "work based on the Program". The definitions are too important to simply summarize and contain too many ambiguities to gloss over. The definitions are:[103]

> **Program:** "This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General

[103] General Public License (GPL), version 2.0, § 0.

> Public License. The "Program", below, refers to any such program or work …"
> **Work Based on the Program: "**a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".)"

Both terms are laced with uncertainties.

As defined, the "Program" covers any program distributed under the GPL. In a situation in which an originally released program is modified by successive programmers who distribute their modified software under GPL, the term "Program" may have multiple and changing meanings. Thus, under one characterization, the "Program" is the last transferred software, including all modifications made before the transfer. This would be the most straightforward interpretation for understanding the transactional dynamics of the GPL world, but it would tend to conflict with the idea, present in the minds of many, that the original creator of the original software also gives a non-contractual license to each licensee. Thus, an alternative view is that, in addition to the last revised version, the term also refers separately to all prior versions in the distribution chain to which the copyright owner has elected to, or been required to, apply the GPL. Under this view, any reference to the "Program" refers separately to all copyrightable, prior versions and their copyright owners.

The term "work based on the Program" suffers from the same issue, but more importantly creates an internal inconsistency that potentially muddies its scope. To understand that internal problem, ask whether the term is limited solely to modifications of a program that create a "derivative work" under copyright law? Some argue that the term is so limited, while others point to language in the definition that is broader than copyright law's conception of derivative work. We will return to that issue shortly, but first pause to layout what is at stake.

As we have seen, GPL allows redistribution of verbatim copies of the Program subject to GPL terms in the transfer. With exceptions that we will discuss shortly, GPL also provides for a right to distribute the modified program ("work based on the Program"), but requires this to be under the terms of the GPL and with source code disclosure. The relevant provisions state in part:

GPL Section 2: You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

> a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
>
> b) 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.
>
> c) If the modified program normally reads commands interactively when run, you must cause it … to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. …[104]

These requirements apply to the modified work as a whole.

GPL Section 3: You may copy and distribute … a work based on [the Program] under Section 2 in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following: [the source code disclosure provisions set out above]

GPL Section 6. Each time you redistribute … any work based on the Program…, the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.[105]

---

[104] GPL makes an exception to this requirement. "Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement." GNU General Public License (GPL), Version 2.0, § 2.

[105] General Public License (GPL), version 2.0, §§ 2, 3, 6.

The effect of these provisions is to cause distribution of the entire "work based on" the GPL-covered program to occur under the terms of the GPL, along with required notice and disclosures of source code, and to preclude "any further restrictions" on the third party's exercise of rights in the modified program. This applies to the new work as a whole, including any and all code or other expression created independently by the licensee. As with other GPL terms, this requirement is enforced by providing that a breach of the terms set for distribution causes termination of the license as to the party in breach and is outside the license.[106]

GPL makes at least three exceptions to aspects of this effect and, for companies desiring to use GPL-covered software in their products, these exceptions can have great significance. The first exception is:

> **Separate Distribution (GPL Section 2):** "If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it."

This language limits the effect of GPL to distributions where the new code and the original GPL code are being distributed as one product. If, instead, a developer creates code that alters the performance of a GPL product, but distributes that code as "separate works", the separate distribution is not covered by GPL, even if, when blended together in a user's system, the GPL and new code form an integrated whole.[107] In effect, this language hinges coverage of GPL viral provisions at least in part to the method of distribution chosen by the new author.

> **Mere Aggregation (GPL Section 2)**: "In addition, mere aggregation of another work not based on the Program with

[106] See General Public License (GPL), version 2.0, § 4. See also discussion of remedies, infra.
[107] General Public License (GPL), version 2.0, § 2 (emphasis added).

> the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License."

The term, "mere aggregation," has no legally established meaning, but would seemingly cover cases where the presence of a GPL program and another program is based on convenience, rather than function. The term should be read in juxtaposition to the coverage of derivative works under GPL and to issues about whether GPL applies to "collective works" as that term is defined in the Copyright Act. Apparently, the intent here is to exclude coverage of independently created software that might be covered solely because the software and a GPL product are included on a single storage medium for convenience. Where the joint storage reflects an expressive or technological purpose other than mere convenience, however, this exception may not apply.

> **Operating System Elements (GPL Section 3):** "[As] a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable."

This exception applies only to the source code disclosure rules. The remainder of GPL continues to apply.[108]

c. Work Based on the Program

Under the GPL, then, "works based on" the GPL program can be distributed, but only if the distributor imposes the GPL terms on both the GPL program and the new code and distributes with no other terms imposed. This clearly limits the distributor's options; the original license either reaches out and captures the new code under its terms or prevents distribution of the program as modified. The GPL terms, including source code disclosure, apply unless an exception exists (e.g., separate distribution of separable work, mere aggregation) or the new product is not a "work based on" the GPL program. It is to that latter issue we now turn.

Stated simply, GPL emits conflicting signals as to what the term "work based on the Program" means. These conflicting signals, when

[108] General Public License (GPL), version 2.0, § 3.

coupled with the inherent difficulty of fitting copyright and license terms to rapidly changing and variously described software technology approaches, leaves large areas of uncertainty in terms of to what types of changes and new code do the terms of the GPL apply. The basic issue centers on whether the coverage of GPL is limited to derivative works distributed as such, or whether it also includes "collective" or other works that include the original program or parts of it and are more than "mere aggregations" of programs on a single media. There is also the underlying question of what constitutes a "derivative work" in the complex world of programming.[109]

The language of the GPL contains hints that could point in several directions. A sampling of the relevant language follows:

> **GPL Section 0: "**a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".)"
> **GPL Section 2(b):** "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."
> **GPL Section 2: "**Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program."

In addition to including the original Program within the definition of a "work based on the Program," Section 0 of the GPL includes "any derivative work under copyright law." The other language in the definition in Section 0, however, is broader and does not parallel the copyright law language or idea of a "derivative work." In addition, Section 2 refers to a "collective work", which is a concept distinct from that of a derivative work, and to a work that simply "contains" the Program.

d. Derivative Work

[109] See discussion in chapter 4.

In addition to including the original Program within the definition of a "work based on the Program," Section 0 of the GPL includes "any derivative work under copyright law." Assuming that the reference is to United States copyright law, this brings in the following statutory definition of a "derivative work":

> a work based upon one or more pre-existing works, such as a translation, musical arrangement, dramatization, fictionalization, motion picture version, sound recording, art reproduction, abridgment, condensation or any other form in which a work may be recast, transformed or adapted. A work consisting of editorial revision, annotations, elaborations or other modifications, which, as a whole, represent an original work of authorship, is a "derivative work."[110]

The thrust of this definition is that the original work (program) be modified, altered or otherwise adapted by the new author in a way that contributes new expression to the work.

The case law gloss elaborates on this in two respects, both of which are relevant to understanding the GPL and its viral or expansive provisions. The first point is that, to be a derivative work, the work must include copyrightable content from the original (the prior work from which it derives) sufficient to justify the conclusion that the second work is at least in part based on the first in a copyright sense. The new work must be *substantially* similar to the underlying work.[111] As the David Nimmer treatise notes,

> Unless sufficient of the pre-existing work is contained in the latter work so as to constitute the latter an infringement of the former, the latter by definition is not a derivative work. Therefore, if the latter work does not incorporate sufficient of the pre-existing as to constitute an infringement of either the reproduction right, or the performance right, then it likewise will not infringe the right to make derivative works because no derivative work will have resulted. Countless works are 'inspired by' or 'based on' copyrighted works, and in that lay sense constitute derivative works. But unless the product is

[110] 17 U.S.C. § 101 (1999) (emphasis added).
[111] *Id.* § 3.01, at 3-3 (1999).

> substantially similar to its forbear, it remains nonactionable.[112]

This sets a baseline for when GPL exposure occurs: if new or independently created code is combined into a single work with GPL-covered code, GPL applies to the work as a whole so long as the included GPL material consists of copyrightable expression from the original work. On the other hand, if all that is taken from the GPL software are the ideas and processes it contains, or other non-copyrightable material, then no derivative work is created and the GPL viral rules would not apply.

The relative amount (quantity) involved is not pertinent, but the presence of copyrightable expression carried into the new, composite work is significant. This leads some to conclude that merely including one line of a GPL program in a new work makes any and all of the work subject to the GPL. On the broadest possible interpretation, one writer noted,

> Commercial entities wishing to use GPL'd software in their own products need to be aware of one thing above all others. The GPL is sometimes referred to as the 'General Public Virus' because it 'infects' derivative works. In simple terms, incorporating one line of GPL'd code in your software makes the entire work subject to the GPL, with the corresponding requirement to provide full source code, and allow unlimited distribution, to any one who asks, for only the cost of reproduction.[113]

---

[112] 4 NIMMER §8.09[A]. *Accord* Alcatel USA, Inc. v. DGI Technologies, Inc., 166 F3d 772, 787 (5th Cir. 1999) (in order to violate the exclusive right to prepare derivative works "the infringing work must incorporate a sufficient portion of the pre-existing work so as to constitute an infringement of either the reproduction right, or of the performance right. ...the finished product must be 'substantially similar' to its forebear." In the words of the House Report relating to the Copyright Act's adoption, "to constitute a violation [of the copyright owner's rights with respect to derivative works], the infringing work must incorporate a portion of the copyrighted work in some form; for example a detailed commentary on a work or a programmatic musical composition inspired by a novel would not constitute infringements. . . ." NIMMER. Appendix 4, at 4-29.

[113] John R. Ackerman, *What's GNU is Software Licensing: The Open Source Revolution* 7-11 (2000) (paper on file with author; emphasis in original).

This would not be true under ordinary connotations of the term "derivative work" unless the single line carries expressive content from the original sufficient that the new work is "based on" the old. When this occurs, however, is a matter of factual and legal analysis applicable to the particular case. If GPL only extends out to "derivative works", then one must ask whether the line of code brought into the new work is expressive in nature (which will often be not true) and, if so, whether the taking would be actionable as infringement unless protected by a license. We are dealing here with blurry, rather than bright lines under copyright law let alone under the language of the license and that, in itself, is a problem.

GPL Section 0 further muddies the analysis by elaborating on what it means as a "derivative work", stating:

> derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".)[114]

This follow-on language seems to indicate that, for this agreement, the term "derivative work" requires only "a work" that "contains" the program or a portion of it, whether that portion is verbatim or modified. While this language contributes to a problem that is discussed below with respect to coverage of "collective works", the better reading of it in this context is that it is subject to the general copyright law concept that a work is not a derivative of another (or a collective work) unless the parts included are expressive in content. Thus, the portion contained in the new work must be expression taken from the original work.

A second case law gloss on "derivative work" in copyright law focuses on the content of the new material. These cases hold that no derivative work exists unless the new contributions or modifications themselves add expression to the original.[115] This conclusion is supported in part by the fact that copyright law treats a derivative work as a new work of authorship, independently copyrightable. Thus, the changes and code

---

[114] General Public License, Version 2.0, § 0.

[115] *See* Gracen v. Bradford Exchange, 698 F.2d 300, 304-05 (7th Cir. 1983) (concept of originality important for derivative works to avoid competing claims between the creator of a derivative work that only trivially alters the original and the person who, with authority of the copyright owner, copies the original work; "a derivative work must be substantially different from the underlying work to be copyrightable").

must be expressive. Purely automated changes (e.g., compiling the GPL source code) or simple, non-expressive additions would not qualify. In the GPL context, however, this distinction is less relevant than it might seem - if the changes do not a derivative work make, they leave the program covered under the basic scope of GPL applicable to the Program itself.

e. Links and Collective Works

Subject to the limitations discussed above, the viral or expansive terms of GPL force application of this license to circumstances where GPL-covered code is actually modified or where the new code is fully integrated into it and the modifications or integrated code entails expressive work, creating a derivative work under copyright law.[116] This, however, leaves open for debate a fair number of cases. The two that have attracted the most attention are cases where 1) GPL software is collected and distributed along with non-GPL software, and 2) where non-GPL software, separately developed and marketed, interacts (links) with the GPL software. In both cases, the question is whether the non-GPL software has become so connected to the GPL software as to invoke application of the "works based on the Program" language in GPL, and thus brought within the mandatory use of GPL license for the otherwise non-GPL software.

The important fact to recognize here is that, while the technology and how it is used may influence a judgment about whether or not GPL applies, in either of these cases, the ultimate question is a legal question or at most a question of intent as manifested by the parties to the transaction. If we focus on this, there is significant room for debate, but there are several clear points of demarcation. These include:

1. If the linked or collected material is associated with the GPL software in a way that makes the composite a "derivative work", then the GPL requires that the new work as a whole be governed by the mandated terms and source code disclosure rules of GPL when or if it is distributed to others.
2. The GPL provides that "mere aggregation … on a volume of a storage or distribution medium does not bring the other work under the scope of this License."[117] This excludes some collections of programs, but hinges on the

[116] General Public License, Version 2.0, § 2.
[117] General Public License, Version 2.0, § 2.

interpretation of the term “mere aggregation.” What is the result when more is involved in grouping programs together than mere convenience? Take, for example, a case in which a GPL program and an independently developed program are intended to interact with each other to produce a significant process and are included on a single medium in order that they can interact more effectively, and are sold as a composite. Does GPL apply to the independently developed program? Perhaps.

3. Under GPL Section 2, if the independently created code can be reasonably treated as a separate program and is separately distributed, then GPL does not apply. This, the GPL tells us, is because the free software drafters of the GPL did not intend to “claim rights or contest your rights to work written entirely by” the licensee. When the independent code is separately distributed, it is indeed a work created entirely by the GPL licensee.

These themes obviously leave room for debate. One point of debate has been whether GPL mandated terms are restricted in application to the original work and any derivative work, covering nothing else. If the combination of the two programs is not technically a “derivative work”, might it nevertheless be included within the scope of the GPL coverage of “works based on” the program? One answer is “no”. This answer hinges on a strict reading of the GPL definition of “works based on” the program, the definition refers to “derivative work” under copyright law. This analysis argues that subsequent, explanatory material should not be used to expand that definition.

The alternative view argues that the entire definition of “work based on the Program” must be read as a whole, and that this definition and other aspects of the GPL suggest that coverage is not limited to derivative works. As previously noted, the definition explains it coverage in a manner that would not fit a derivative work under copyright law. This second view argues that GPL thus suggests that it is covering more than what would be true under the copyright law concept. Otherwise, the explanatory material would have no meaning. To understand this, compare the definition of “derivative work” quoted earlier with the explanatory language of GPL Section 0.

> The "Program", below, refers to any such program or work, and a "work based on the Program" means either the

> Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language.[118]

The explanatory material does not require that the new work be based on the original, but mere that it contain verbatim or modified part of the original. The fact that broader coverage is intended is further suggested by language in Section 2 stating that the intent of GPL "is to exercise the right to control the distribution of derivative or collective works based on the Program."[119] The reference to "collective work" brings in an additional type of copyrightable work.

Since a collective work is a form of "compilation under copyright law, the following quotes both definitions:

> A "compilation is a work formed by the collection and assembling of preexisting materials or of data that are selected, coordinated, or arranged in such a way that the resulting work as a whole constitutes an original work of authorship. The term "compilation" includes collective works.
> A "collective work" is a work … in which a number of contributions, constituting separate and independent works in themselves, are assembled into a collective whole.[120]

This concept requires that the selection, coordination, or arrangement constitute expressive authorship and that the assembled part be separately copyrightable works. When that occurs and the works are treated as a whole, they form a copyrightable, collective work. The argument for including programs grouped together with a GPL program thus asserts that the composite arrangement is treated and distributed as

---

[118] General Public License, Version 2.0, § 0.

[119] Moreover, after the terms and conditions of the GPL the authors append a statement about how users can make their programs GPL'd, in which they state: "This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License." *See* GPL section entitled "How to Apply These Terms to Your New Programs".

[120] 17 U.S.C. § 101.

a whole as a copyrightable work and that, as required by the definitions of GPL, that work contains a copy of the original, GPL-covered program.[121] The collection (aggregation?) of programs is, one could argue, not excluded by GPL's exclusion of "mere aggregations" because, by definition, the aggregation itself follows an expressive theme and qualifies for copyright; it is not a "mere" aggregation.

There are supporters for each of the two interpretations outlined here, and no case law that answers the question of which view governs. Yet, it is also important to note that the degree of difference in practice between a collective work and a derivative work in the programming environment is uncertain. Both types of works are "based on" the original work in the sense they include and rely on all or part of it. To the extent there is a difference, in the one ("collective work"), the expression lies in creative selection or organization of separable works that are not necessarily themselves changed, while in the other ("derivative work") the expressive elements are often presented as a unitary whole with the original recast, revised or reformed in the new work.[122] The reshaping or adapting of an original work is distinctively part of the derivative work concept, while retaining and melding separable contributions is distinctively a collective work concept.[123] If collective works are within the scope of the GPL mandated terms, then its coverage threatens a number

---

[121] Indeed, this is buttressed by the GPL exclusion for mere aggregations. The difference between a collective work and a "mere aggregation" seems to be that the latter is done for pure convenience, while a collective work entails organization or selection with a creative purpose in mind.

[122] *See, e.g.* Lee v. A.R.T. Co, 125 F.3d (7th Cir. 1997) (mounting works on ceramic tile did not create derivative work, any more than displaying a painting in a frame or altering how it may be displayed; hence, purchaser in a first sale did not infringe by placing works on tile). Indeed, the law seems well-settled that the derivative work must be original. More fundamentally, the courts seem to embrace the standard articulated in L. Batlin & Son, Inc. v. Snyder, 536 F.2d 486 (2nd Cir. 1976) (for a work to be copyrightable a work must contain "some substantial, not merely trivial, originality"), *cert. denied* 429 U.S. 857 (1976); Durham Industries, Inc. v. Tomy Corp., 630 F.2d 905 (2nd Cir. 1980); Entertainment Research v. Genesis Creative Group, 122 F.3d 1211, 1220 (9th Cir. 1997) (applies test).

[123] Castlerock Entertainment v. Carroll Publishing Group, 150 F.3d 132, 143(2nd Cir. 1998). *Cf.* Harris Custom Builders, Inc. v. Hoffmeyer, 92 F.3rd 517, 518, 520 (7th Cir. 1996) ("if the secondary works sufficiently transforms the expression of the original work such that the two works cease to be substantially similar, then the secondary work is not a derivative work and, for that matter, does not infringe the copyright of the original work.").

of products that collect programs intended to work together, albeit as separately identifiable parts of a collection.

This brings us to the issue of linking. If one program is linked to a GPL program, does it fall within the GPL mandate when the two are distributed together?

Initially, it is important to recognize that a work is not based on another work merely because it contains cross-references to that other work. To be a derivative work the work must (i) borrow original and expressive content from another preexisting work and (ii) recast, transform or adapt the preexisting work upon which it is based.[124] Mere pointers or references are not sufficient.

In software products, however, linking typically involves more than a simple reference. Some discussions of the scope of GPL mandated or viral terms distinguish between static and so-called dynamic linking. Insofar as it has a legally-relevant basis, the distinction suggests that programs that link only as part of engaging in operations (e.g., dynamically) do not create derivative works for purposes of GPL when distributed in inactive form. That is not, however, stated in GPL. An indication of the FSF viewpoint on linking, however, can be gleaned from the Preamble to the GNU Lesser General Public License, a standard form promulgated by FSF to deal with licensing of some libraries. The Preamble states:

> When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.[125]

This treats a "statically" linked program and its linked GPL program as a "combined" work, which the author of the Preamble views as a derivative work.

To the extent the distinction between dynamic and static linking can be supported in law, that support would come from the requirement, suggested in a Ninth Circuit decision, that a derivative work requires that

---

[124] *See* 1 PAUL GOLDSTEIN, COPYRIGHT § 2.16.2, at 2:214 (2d ed. 2000).
[125] GNU Lesser General Public License, Preamble.

there be a work (e.g., the product of the dynamic link) that must exist in a "concrete or permanent form" for a sufficient (albeit brief) time.[126]

While language in GPL may suggest that some linking of programs is within the scope of its mandated terms coverage, there is only sparse case law on how to treat linking among programs for copyright purposes. Most of the copyright cases on linking deal with linking to remote sites on the Internet.[127] An exception is *Dun & Bradstreet Software Services, Inc. v. Grace Consulting, Inc.*,[128] where the court affirmed a copyright infringement claim against a consulting firm that had developed a program that, by using call and copy commands, worked off of the plaintiff's program. One defense was that the copying was de minimis. The court rejected that argument on the following basis:

> In supporting its de minimis defense, Grace [the defendant] asserts that the quantitative infringement amounted to only twenty-seven lines out of 525,000 lines. This argument is irrelevant as a matter of law…. The unrefuted trial testimony was that if one considers Grace's use of Copy and Call commands to gain access to PAYTXABR and the Geac [the plaintiff] code, the CNR W-2 program actually consists of 62% Geac code, and the GMI W-2 program possesses approximately 43% of Geac's code. Much more significant, however, than the quantity of copy is the quality of the material purloined. A de minimis defense does not apply where the qualitative value of the copying is material. [The technical experts] agree that Geac's software would not work if PAYTXABR were removed from it and that Grace's infringing W-2 software would not work without its copies of PAYTXABR. Thus, the information Grace copied was highly critical.

The court in *Grace Consulting* was not asked to decide whether the new program was a derivative work. It was sufficient to conclude that the defendant's work copied expression from the original software. But, were

---

[126] Lewis Galoob Toys, Inc. v. Nintendo of America, Inc., 964 F.2d 965, 967 (9th Cir. 1992); *see also* Micro Star v. FormGen Inc., 154 F.3d 1107, 1111 (9th Cir. 1998). *See* 2 PAUL GOLDSTEIN, COPYRIGHT § 5.3.1, at 5:84 –2, n 21 (2d ed. 2000).

[127] See discussion in chapter 1.

[128] Dun & Bradstreet Software Services, Inc. v. Grace Consulting, Inc., 307 F3d 197 (2d Cir. 2002).

that work viewed in other terms, in light of the courts analysis it is likely that the defendant's program would be viewed as a derivative of the original software.

D. License Terms Relating to Patent Rights Issues

While open source and free software (FSOS) licenses originally focused on rights related to copyright law, it has been clear from the mid-1990's that patent rights are also important. The treatment of patent rights, however, is both spotty and incomplete in most FSOS licenses. Many licenses ignore the question and focus on copyright. Others attempt to deal with patent-related questions, but the FSOS community has yet to develop a consensus on this issue that would protect licensees from patent infringement risk, while not creating an obligation that reaches into a participant's patent portfolio too broadly to be commercially reasonable. Patent rights are more troublesome to deal with in an open environment contemplated by FSOS software since the claims they generate are not bounded by access to and copying from an identifiable work.[129]

GPL discusses patent issues, but contains no express grant of patent rights or of rights associated with the patent law exclusive rights to make, use, sell or offer to sell the patented invention.[130] Instead, GPL simply states what would be true in any event - a mandate derived from patent law or an agreement to license patent rights that is inconsistent with the terms of the GPL does not absolve the licensee from complying with those terms. It states:

> If, as a consequence of a court judgment or allegation of patent infringement or for any other reason … conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then … you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way

[129] See Raymond T. Nimmer, the Law of Computer Technology ch 2. (1997, 2005 Supp.)
[130] 35 U.S.C. § 271(a).

> you could satisfy both it and this License would be to refrain entirely from distribution of the Program.[131]

This language does not give any comfort or protection to the licensee whose use of an FSOS product might infringe a patent. Rather, it accentuates the problem - if a settlement or other effort to avoid patent infringement liability occurs, unless the settlement produces terms consistent with GPL, the settlement may block the licensee's continued use of the software. This may not be a unique effect of licensing regimes, but unlike with most commercial licenses, the licensee faced with this dilemma is not likely to have the capability of negotiating an exception from the GPL terms to accommodate a patent license or settlement - indeed, the FSOS philosophy would make such an exception difficult. Viral FSOS licenses tend to create a rigidity in terms that inhibits such specialized negotiation.

Three methods of dealing with patent infringement issues have surfaced in FSOS licensing. One deals with infringement warranties or patent indemnification duties. The second focuses on encompassing patent grants in the FSOS license. The third deals with "patent retaliation" that create negative consequences if a party in an FSOS chain asserts a patent infringement claim. We discuss the latter two approaches in the following sections, but defer the warrant-indemnification question to our later discussion of warranties in general.[132]

1. Grant of an FSOS Patent License

While some FSOS licenses, including GPL, do not create a patent license, many licenses approved by the Open Source Initiative contain not

---

[131] GPL § 7. See also LGPL § 11 (same language). The Preamble to GPL, in typically ambiguous and potentially misleading form, comments: "Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all." The comment is misleading because, if a process is patented and the patent reads on some aspect of an FSOS product, licensing the right to use the patented process, for example, does not necessary step into the realm of copying, distribution or modification of the software code, which is what the GPL is concerned with. Similarly, if a patent is in fact issued that covers part of the GPL-covered software process, the GPL is not in itself sufficient to allow end users or others to use that patented process (or make or sell it) without permission from the patent owner.

[132] §§ 11:48 - 11:50.

only a copyright grant, but a grant of a license to patent rights. Version 2.1 of the Open Software License, for example, states:

> 2) … Licensor hereby grants You a world-wide, royalty-free, non-exclusive, perpetual, sublicenseable license, under patent claims owned or controlled by the Licensor that are embodied in the Original Work as furnished by the Licensor, to make, use, sell and offer for sale the Original Work and Derivative Works.….
> 4) …. Nothing in this License shall be deemed to grant any rights to trademarks, copyrights, patents, trade secrets or any other intellectual property of Licensor except as expressly stated herein. No patent license is granted to make, use, sell or offer to sell embodiments of any patent claims other than the licensed claims defined in Section 2.[133]

The scope of the patent license is relatively narrow. It refers to only the patent rights of a particular licensor and only as those claims might be embodied in software as delivered to the licensee by that licensor.

This language reflects an effort to tailor the patent license to as narrow of a scope as is needed to implement the FSOS concept. One risk in establishing a patent license chain lies in binding the patent holder to too broad of a grant and, thus, rendering FSOS prohibitive for patent holders. For example, a grant of patent rights in any process associated with the FSOS software would reach rights that might not have existed when the software was transferred and might, depending on the language, grant rights outside the particular FSOS product (e.g., a patented process to compress data that might be incorporated in FSOS software solely for use in that software or for use in any other context).

In the Open Software License, the term "licensor" refers only to the owner of the copyright in an original work of authorship ("Original Work") covered by the license. The license, thus, does not cover patents controlled by others, even if they are in the distribution chain, except to the extent they create modifying code. Even then, the grant runs downstream, not upstream. Also, it does not cover patent claims of the licensor that might apply to the modified software as modified, rather than as originally delivered. Finally, the grant is restricted to use of the patent claim in reference to the "original work" and "derivative works", thereby excluding from the license scope any other use of the patent.

---

[133] Open Source License §§ 2,4 (2003).

That being said, this approach to a patent license contributes greatly to the sustainability of FSOS and impinges on the interests of patent owners in only a limited way. Under such circumstances, the patent grant is what would be expected to make the transaction commercially sensible.[134]

Since patent law creates rights in inventive processes whether or not the alleged infringer copies or was even aware of the patent, the significance of any existing patent to an FSOS product can change as the product morphs through community modifications. Software implementing a non-infringing process as delivered might later morph into an infringing product. This presents two issues important to FSOS with respect to patent claims. First, to what extent *should* a licensor (original or later transferor) be required to grant patent rights under its control as to all versions of the FSOS software, even those that arise because of modifications after the software leaves the licensor's control? Second, as to contributors who modify or use the software, should they also grant rights under the patents they control, regardless of whether they contribute expressive (e.g., copyrighted) code to the work? There are various ways to answer these questions, but as the scope of patent grants expands, there will be an increasingly intrusive impact on property interests of commercial and other participants in the FSOS process.

The Realnetworks Public Source License Version 1.0 created a patent cross-license in which licensee's (users of the software under the license) grant to the licensor and third parties:

> (a) You grant to Licensor and all third parties a non-exclusive … license under Your Applicable Patent Rights and other intellectual property rights owned or controlled by You, to make, sell, offer for sale, use, import, reproduce, display, perform, modify, distribute and Deploy Your Modifications of the same scope and extent as Licensor's licenses ….; and
>
> (b) You grant to Licensor and its subsidiaries a non-exclusive … license, under Your Applicable Patent Rights and other intellectual property rights owned or controlled by You, to make, use, sell, offer for sale, import, reproduce,

[134] Indeed, in transactions that would be covered by UCC Article 2 or by UCITA, an implied warranty relating to non-infringement would be present and, like the license grant here, it would be limited to the software as delivered, rather than as modified or as used. See UCITA § 401 (2000 Official Text); UCC §2-312 (1998 Official Text).

> display, perform, distribute, modify or have modified (for Licensor and/or its subsidiaries), sublicense and distribute Your Modifications, in any form and for any purpose, through multiple tiers of distribution.[135]

For both the licensor and the licensee, the patent grant extends to after-acquired patent rights.[136] But while the licensor's grant necessarily applies to the modifications it makes, the licensor's grant is limited to claims that "are necessarily infringed by using or making the Original Code alone and not in combination with other software or hardware." This, again, reflects the need to tailor the patent grant to the primary focus of the FSOS structure - grants and restrictions associated with the reproduction and distribution of the copyrighted code.

The Apache License, Version 2.0 conveys a broad patent grant from all contributors to the software to all licensees:

> Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted.[137]

---

[135] Realnetworks Public License Version 1.0, § 3(a)(b).

[136] The license defines "applicable patent rights" as follows: ""Applicable Patent Rights" mean: (a) in the case where Licensor is the grantor of rights, claims of patents that (i) are now or hereafter acquired, owned by or assigned to Licensor and (ii) are necessarily infringed by using or making the Original Code alone and not in combination with other software or hardware; and (b) in the case where You are the grantor of rights, claims of patents that (i) are now or hereafter acquired, owned by or assigned to You and (ii) are infringed (directly or indirectly) by using or making Your Modifications, taken alone or in combination with Original Code." Realnetworks Public License Version 1.0, § 1.1.

[137] Apache License, Version 2.0 § 3 (January, 2004). See also Mozilla Public License 1.1 (MPL 1.1) ("Subject to third party intellectual property claims, each Contributor hereby grants You [licensee] a world-wide, royalty-free, non-exclusive license … **(b)** under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to

As this indicates, the grant is limited to the contributed portions interacting with the work "to which [the contribution] was submitted." Again, this reflects a useful restriction of the grant, limiting it to an interaction or product that the licensor controls or creates, and not covering changes by others over which it has no control. The language, however, does not indicate whether or not it applies to after-acquired patents.

Overall, then, there has been a reaching out in FSOWS licenses into the realm of patent law as a part of the FSOS license model. Importantly, however, none of these licenses covers the licensee's risk of liability for infringement of third party rights. To the extent that they deal expressly with such claims, FSOS licenses disclaim any warranty of non-infringement regardless of the scope of their respective patent and copyright grants.[138]

2. Patent retaliation clauses

A second approach to patent issues in an open source or free software (FSOS) project involves "patent retaliation" clauses. The following language from the Apache license illustrates the approach. Language to a similar effect is found in most FSOS licenses that deal with patent rights.

> If You [licensee] institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.[139]

In this license, the reference to "You" means "an individual or Legal Entity exercising permissions granted by this License", essentially any licensee of the work. The intent is to use the leverage and sunk costs that may arise from a person having used the FSOS software as a licensee to forestall patent litigation against any participant in the license chain.

---

make, use, sell, offer for sale, have made, and/or otherwise dispose of: 1) Modifications made by that Contributor (or portions thereof); and 2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).").

[138] See § 11:50.

[139] Apache License, Version 2.0 § 3 (January, 2004).

Notably, this termination rule is not restricted to the contributor's code, but applies to any patent claim against the software in original or modified form. In effect, this would create an umbrella for the FSOS project in respect to patent claims that exceed that granted with respect to copyright claims. There is no prohibition on a suit for copyright infringement in the foregoing language in the event a third party "contributes" copyrighted code to the software, but the copyright is held by another contributor who elected not to contribute that code.

There is some doubt about whether a court would enforce this automatic termination provision. Courts in this country are often unwilling to enforce clauses that would work a forfeiture of rights (essentially, impose a loss on the breaching party far in excess of relatively minor breaches).[140] Additionally, the clause uses the leverage of the FSOS software to control rights that are outside the scope of the FSOS property. This is one example of FSOS terms that might raise questions about patent or copyright misuse, or even antitrust concerns. But, as with most of the FSOS framework, there has yet to be significant case law on point.[141]

Retaliation clauses are one of the clauses that has created uneasiness among commercial parties using or contemplating use of FSOS software. The issues focus on business risks involved in losing potentially important licenses simply because the party undertakes to enforce a property right. Another issue, of course, is that the equivalent to a waiver of such patent rights against the FSOS occurs in a supposedly free or open license where full consideration of what was being given up may not occur.

In this regard, the Apache license quoted above is a relatively narrow approach since it terminates on "patent licenses" granted under the FSOS structure. Consider, instead, the language from the Open Software License (Version 2.0), which is also present in the Academic Free License, the Common Public License, the Mozilla license,[142] and others:

---

[140] See Murray on Contracts § 107 (2001).

[141] On the misuse issue, see Alcatel USA, Inc. v. DGI Technologies, Inc., 166 F3d 772 (5th Cir. 1999).

[142] See Mozilla Public License, Version 1.1, §8.2: "If You initiate litigation by asserting a patent infringement claim (excluding declatory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You file such action is referred to as "Participant") alleging that: (a) such Participant's Contributor Version directly or indirectly infringes any patent, then any and all rights granted by such Participant to You under Sections

> This License shall terminate automatically and You may no longer exercise any of the rights granted to You by this License as of the date You commence an action, including a cross-claim or counterclaim, for patent infringement (i) against Licensor with respect to a patent applicable to software or (ii) against any entity with respect to a patent applicable to the Original Work (but excluding combinations of the Original Work with other software or hardware).[143]

The term "original work" refers to the first version of the software and to any modified version that constitutes an original work of authorship under copyright law covered by the license.

The feature of this language that distinguishes it from the Apache language is that the language purports to terminate the entire FSOS license, rather than merely rights under a patent license. Secondly, subsection (i) is not limited to a patent claim filed with respect to the FSOS software, but refers to any patent claim filed against the licensor for any "patent applicable to software." Subsection (ii) refers to a suit against "any entity" with respect to any patent "applicable" to the Original Work. In effect, this uses the threat of termination of the license to stifle enforcement of a related patent against anyone. This, of course, heightens the cost to a participating company in terms of lost rights and increases the risk of inadvertently losing the licensed rights.

Cognizant of this problem, Larry Rosen, General Counsel of the Open Source Initiative, in 2004 proposed a revision of the patent retaliation clause in the two licenses his group controls. The new language, which appears in version 2.1, is as follows:

> This License shall terminate automatically and You may no longer exercise any of the rights granted to You by this License as of the date You commence an action, including a cross-claim or counterclaim, against Licensor or any

---

2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively, unless if within 60 days after receipt of notice You [cure the problem] or (ii) withdraw Your litigation …. (b) any software, hardware, or device, other than such Participant's Contributor Version, directly or indirectly infringes any patent, then any rights granted to You by such Participant under Sections 2.1(b) and 2.2(b) are revoked effective as of the date You first made, used, sold, distributed, or had made, Modifications made by that Participant."

[143] Open Software License, version 2.0, § 10.

> licensee alleging that the Original Work infringes a patent. This termination provision shall not apply for an action alleging patent infringement by combinations of the Original Work with other software or hardware.[144]

Clearly an improvement and a narrower intrusion on intellectual property rights, the language of this clause is somewhat ambiguous because of the reference to "You" (the particular licensee) and the "Original Work" (the software as conveyed in this particular license). Rosen, however, commented that the "new section 10 defensive termination provision terminates the license to this Original Work only if the licensee asserts a patent claim against this Original Work. [This] meaningfully reduce[s] the scope of the patent termination provision and make[s] it friendlier to patent-owning companies. That's the whole point. Such companies can now feel more comfortable in-licensing open source software. The community will grow and more open source software will be created."[145]

E. Warranties

Warranties about quality and about the risk of non-infringement are an important feature of ordinary software licensing. In the free software and open source context, however, the routine approach has been to disclaim all warranties in the standard form license, but to permit (expressly or implicitly), any particular licensor to make express warranties to its licensee. The effectiveness of the disclaimer format is questionable in several respects that are outlined below, but even more importantly, the lack of warranty protection for licensees is a defining characteristic of this licensing model.

In one analysis of FSOS software, the FDIC issued the following guidance regarding FSOS software and warranty issues:

> Proprietary software licenses customarily include a warranty that the software will achieve a specified level of performance and an indemnity that the vendor will defend the user in the event of an infringement lawsuit. In contrast, FOSS is customarily licensed "as is," without warranty or indemnity. Recently, VARs have begun to market FOSS with dual licenses. The first license is usually some form of the

[144] Open Software License, version 2.1, § 10 (2004).
[145] E-mail from Lawrence Rosen to OST license-discuss listserv, dated March 5, 2004.

> GPL, and it covers the rights and obligations associated with the use of the software. The second license describes support services to be provided by the VAR and may include performance warranties and indemnities. In some cases, the VAR may agree to support a particular version of the FOSS for a set time. Institutions should evaluate carefully the terms of any indemnification offered by a VAR, as well as its financial capacity to provide a robust defense. Institutions may also consider third-party insurance, if available.[146]

As this indicates, issues about warranty and indemnification present business concerns about the comparative value of different software. As discussed earlier, the existence of dual licensing models, described in this FDIC comment indicates that the business choice between FSOS and "proprietary" is not always a clear choice. The degree to which competing considerations exist is heightened as "proprietary" vendors increase warranty and indemnification protection. Thus, Microsoft adopted a practice of giving patent infringement indemnification obligations to volume licensees. That practice does not exist in FSOS.

Licenses in the open source and free software communities routinely disclaim warranties about the quality of the software. The following language from the BSD license is illustrative:

> THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE,

[146] FDIC, Risk Management of Free and Open Source Software, p5 (Oct. 21, 2004).

> EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.[147]

Some FSOS licenses, most notably the GPL also obligate the licensee to accompany redistribution of the software with a disclaimer effective at least as to prior parties in the distribution chain. The GPL states:

> You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you 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 this License along with the Program.

Virtually none of the standard FSOS licenses contain affirmative warranties as to quality. Several, however, allow a licensee to create and give warranties as to its responsibility if it so chooses. Some licenses go even further in creating a presumption of protection for persons involved in a chain of distribution or development. The Common Public License, for example, provides:

> Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its

[147] BSD License.

> distribution of the Program in a commercial product offering.[148]

The rationale for disclaiming warranties lies in protecting the alleged volunteer and no cost nature of contributions to FSOS software. Individuals involved in upstream participation without charging a fee, the theory goes, should be protected from risk. As far as this goes, the premise has merit. But it is less clear for situations where FSOS software is sponsored by a commercial developer that elects to use FSOS licensing for its own product that it developed alone.

A general practice of disclaiming implied and express warranties and substituting for them a limited, express warranty is common in "proprietary" software.[149] It has been transported into the FSOS world, at least as to the disclaimers, although there are numerous transactions in which no substitute express warranty is created. In many states, however, such disclaimers are ineffective.

If the disclaimers are effective, this creates an obvious issue for end users and developers who employ FSOS code. The issue is present even if one assumes, as do adherents to FSOS, that community development creates higher quality software than other development. Higher quality may or may not be present in a particular product and, whether or not it is, that quality may still fail to meet the licensee's needs. The issue ultimately places distributors of FSOS software on the opposite side of the table from end users and those who will use the code to make their own products.

Both in cases where community development in fact has occurred and cases where it has not but the FSOS model permits it to occur later, an additional argument for allowing disclaimers lies in the character of the distribution itself. Thus, some claim that no warranties need be present because source code and the right to modify the software is made available to the licensee, who can correct any qualitative problems. But, of course, this is like saying that automobile manufacturers need make no warranties about the transmission in their cars because buyers are free to

---

[148] Common Public License, Version 1.0, § 4. See also Eclipse Public License -v 1.0 (same language); CUA Office Public License Version 1.0 ("You must make it absolutely clear than any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.").

[149] See chapter 6.

modify and correct problems in those transmissions for themselves. Obviously, many users lack the time or expertise to make the adjustments and, especially for implied warranties, the point of creating warranty in law is to place some baseline of quality requirements on products put into the stream of commerce.

But are the FSOS disclaimers effective?

To answer this question, we need first to ask what law applies that might create the warranties in the first place. Implied warranties about quality exist in appropriate cases (unless disclaimed) if the transaction is governed by UCC Article 2, Article 2A, or by UCITA.[150] As discussed elsewhere, a number of courts have held that software delivered on tangible media falls with the scope of UCC Article 2, although proposed revisions to Article 2 may reverse that result and more recent cases have been less willing to follow earlier precedent on this issue.[151] In the increasingly common case where software is obtained by downloading it, the leading case to consider the issue suggested that such downloaded software could scarcely be considered tangible goods.[152] Except in states that have enacted UCITA, it is appropriate to treat the status of such software in reference to Article 2 coverage as unresolved. It is also appropriate to conclude that Article 2 should not apply since the precondition for its application (a sale of tangible goods) is not present.

If applicable, Article 2, Article 2A, or UCITA recognize that "express warranties" can be created by conduct or words that become part of the parties' bargain, and that, in certain cases, implied warranties of merchantability or fitness for a particular use may arise unless disclaimed.[153] For purposes of the free software and open source communities, however, the important point is that the existence of an implied warranty under these laws requires the existence of the preconditions for the warranty. For example, in Article 2, Article 2A, and UCITA, an implied warranty of merchantability exists only if the transaction involves a seller or licensor who is a merchant with respect to goods of the

---

[150] See UCC §§ 2-313, 314, 315 (1998 Official Text); UCC §§ 2A-210, 212, 213 (1998 Official Text); UCITA § 402, 403, 404 (2000 Official Text).

[151] See generally ch,. 6 (1997, 2004). The proposed revisions of Article 2 specifically exclude "information" from the definition of goods. The term information has been interpreted in numerous other contexts to include computer programs.

[152] Specht v. Netscape Communications Corp., 306 F.3d 17 (2d Cir.2002).

[153] See chapter 6.

type.[154] What constitutes a merchant can be debated in some cases, but the focus is on people who are in business of providing the subject matter. This presumptively excludes the college student who voluntarily contributes code to an FSOS product. This fact cuts in several directions. It provides a useful protective umbrella for many FSOS participants, but on the other hand vitiates some of the rationale for disclaimers. Companies like IBM, Red Hat or Apple are merchants, even when involved in FSOS transactions, but why should their contributions when embodied in a commercial product they distribute be shielded from warranty obligations that might exist for "proprietary" products?

Section 410 of UCITA states:

> (a) **[Free software defined.]** In this section, "free software" means a computer program with respect to which the licensor does not intend to make a profit from the distribution of the copy of the program and does not act generally for commercial gain derived from controlling use of the program or making, modifying, or redistributing copies of the program.
>
> (b) **[Implied warranties inapplicable.]** The warranties [against infringement and of merchantability] do not apply to free software.[155]

This excludes from implied warranty coverage all of the volunteer and similar, non-commercial participants in the FSOS communities. It does not exclude commercial, profit-making entities who are left on an equal playing field with other software producers.

If implied warranties would otherwise exist, Article 2, Article 2A, and UCITA require that the disclaimer be conspicuous and contain certain, specified language. Some FSOS disclaimers do not meet these requirements. Even more important, however, the disclaimer must be part of the contractual relationship from which the warranty arose. Purely non-contractual notices do not meet this condition and may be ineffective. Proposed revisions of Article 2 alter the required disclaimer language and, if enacted, would require rewriting <u>all</u> FSOS licenses with respect to <u>disclaimers</u>.

---

[154] See UCC §§ 2-313, 314, 315 (1998 Official Text); UCC §§ 2A-210, 212, 213 (1998 Official Text); UCITA § 402, 403, 404 (2000 Official Text).

[155] UCITA § 410 (2002 Official Text).

Finally, in more than ten states, state law precludes disclaimer of the warranty of merchantability. Under federal law, for transactions involving "consumer goods", even if those consumer goods are provided to a business, the merchantability warranty cannot be disclaimed if the provider gives a written warranty.[156] Whether software of any type comes within this law is questionable, but if it does, FSOS software would be treated the same as any other software.

The open source and free software model that attempts to shield contributors from implied and other warranty obligations with respect to the quality of the software's performance (unless the participant expressly take on such obligations) is also applied in practice to claims relating to intellectual property rights infringement. Many FSOS licenses routinely disclaim warranties of title and non-infringement with respect to third-party claims. Although the disclaimer language varies, the language of the Realnetworks license is illustrative, although more explicit than many FSOS licenses:

> You expressly acknowledge and agree that although Licensor and each Contributor grants the licenses to their respective portions of the Covered Code set forth herein, no assurances are provided by Licensor or any Contributor that the Covered Code does not infringe the patent or other intellectual property rights of any other entity. Licensor and each Contributor disclaim any liability to You for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, You hereby assume sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow You to make, use, sell, import or offer for sale the Covered Code, it is Your responsibility to acquire such license(s).[157]

Inexplicably, the GPL, which expressly disclaims implied warranties of quality, makes no specific reference to disclaimer of any non-infringement warranty or warranty of good title. The license does, however, provide that the software is provided ""AS IS" WITHOUT

---

[156] This, of course, is the Magnuson-Moss Warranty Act. See discussion in chapter 6.

[157] Realnetworks Public Source License, version 1.0.

WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED …."[158] This language presents the issue of whether FSOS software licensed under this language does or does not contain an implied warranty that the software does not infringe third party rights. No such warranty apparently exists to begin with unless the transaction comes with the terms of UCC Article 2 or UCITA.[159] While the case law is split and there is doubt that Article 2 should apply to downloaded software in any case, if Article 2 applies, the GPL language may not be sufficient to disclaim a non-infringement warranty.[160] As a result, one can argue that the Article 2 warranty binds all parties in the distribution chain.

Under Article 2, a transferor that is a merchant "regularly dealing" in goods of the kind makes the following warranty:

> That the goods shall be delivered free of the rightful claim of any third person by way of infringement or the like[161]

This language, by referring to "merchants" who "regularly deal" in the type of property, excludes many casual, voluntary contributors that might be involved in a truly community-driven open source development. It does not exclude commercial vendors. Even as to them, however, the warranty is limited to the goods (software) as delivered. This means that it does not extend to modifications made by the licensee or other transferees after the particular merchant was involved.

Although this warranty arises by operation of law, rather than by agreement, it is not described as an "implied" warranty. This is significant for disclaimer purposes because GPL, LGPL and licenses following their lead in this regard, seemingly rely on the general reference to "as is" and "no warranties" for purposes of disclaiming this warranty. Article 2's rule on the effect of an "as is" clause states:

---

[158] GPL § 11. See also LGPL § 15 (using the same language). Compare the Eclipse Public License § 5: "…THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE."

[159] There is no significant case law support for finding such a warranty under the common law of licensing. See discussion in Raymond T. Nimmer & Jeff Dodd, Modern Licensing Law §§ 8:6 - 8:13 (2004).

[160] On this issue, as on many others, UCITA tracks the Article 2 approach. See UCITA § 401 (2000 Official Text); UCC § 3-312 91998 Official Text).

[161] UCC § 2-312(3) (1998 Official Text).

> unless the circumstances indicate otherwise, all implied warranties are excluded by expressions like "as is", "with all faults" or other language which in common understanding calls the buyer's attention to the exclusion of warranties and makes plain that there is no implied warranty [emphasis added][162]

This language seems to limit the impact of "as is" to eliminating implied warranties. Indeed, with reference to the warranty of "good title", which is contained in the same Article 2 section as the warranty of non-infringement, Article 2 states that disclaimer requires specific language or circumstances that give reason to know that no claim of title is being made.[163] Under that analysis, the disclaimer language in the Realnetworks license suffices, but the GPL language does not.

A caveat must be placed on the conclusion that the general language of the GPL disclaimer does not suffice. The Article 2 language is specifically directed to disclaimer of the "good title" warranty and, in what may be a drafting mistake, no similar provision is specifically directed to the infringement warranty. While most observers conclude that the "specific language" rule applies to the infringement warranty, if it does not, we are left with a situation in which no disclaimer language applies ("as is" is applicable to implied warranties, "specific language" would be as to "good title", and the rules regarding quality warranties are limited to those implied warranties). This might be interpreted to mean that the warranty cannot be disclaimed. For cases governed by it, however, UCITA resolves the issue in the following terms:

> [A] warranty under this section may be disclaimed or modified only by specific language or by circumstances that give the licensee reason to know that the licensor does not warrant that competing claims do not exist or that the licensor purports to grant only the rights it may have. … [Language] in a record is sufficient if it states "There is no warranty against interference with your enjoyment of the information or against infringement", or words of similar import.[164]

---

[162] UCC § 2-316(3)(a) (1998 Official Text).
[163] UCC § 2-313(2) (1998 Official Text).
[164] UCITA § 401(d) (2000 Official Text).

The noninfringement issue has relevance both to licensees that intend to redistribute the FSOS software and to those that simply plan to employ the software as an end user. For the latter type of licensee, both copyright and patent infringement risks are present. Use of the software in a computer will often involve making copies internal to the machine under copyright law.[165] These copies are not authorized by law unless the licensee is the owner of a copy.[166] With respect to patent rights, the owner of a patent has the right to preclude others from using its patented process and this directly related to end users. An FDIC guidance, quoted on a different point earlier, makes the following observations regarding patent risks:[167]

> Institutions that use computer software run the risk of being sued for either copyright or patent infringement. However, the potential for an infringement lawsuit is more likely if the institution is using FOSS because, unlike proprietary software, FOSS is developed in an open environment where code is shared and modified by numerous unaffiliated parties. This code sharing increases the possibility that proprietary code may be inserted in the FOSS at some point during the development process. Institution can mitigate this risk by
>
> - Retaining qualified legal counsel to advise the institution concerning FOSS licensing.
> - Implementing enterprise-level policy and business rules that mandate strict adherence to license terms and conditions.
> - Using automated tools to track licenses and changes.
> - Understanding the consequences of combining FOSS and proprietary software.
> - Evaluating the strength of any indemnities.
> - Developing contingency plans that will allow the institution to continue operating even if infringing code is taken out of production.

---

[165] See MAI Sys. Corp. v. Peak Computer, Inc., 991 F.2d 511 (9th Cir. 1993).
[166] 17 USC § 117.
[167] FDIC, Risk Management of Free and Open Source Software, p5 (Oct. 21, 2004).

- Using a control mechanism to ensure that all code contributed to FOSS projects is original and written onsite, such as a “clean room.”

Some “proprietary” software vendors such as Microsoft have begun to provide patent indemnity protection for licensees, but that practice is not generally present in FSOS software.