GENERAL PUBLIC LICENSE 3.0: HACKING THE FREE SOFTWARE MOVEMENT'S CONSTITUTION

*Robert W. Gomulkiewicz**

TABLE OF CONTENTS

---

* Director, Intellectual Property Law and Policy Program and Associate Professor of Law, University of Washington School of Law. I owe thanks to the participants at the 2005 IPIL/Houston Santa Fe Conference: *Transactions, Information and Emerging Law*, and the Pacific Intellectual Property Scholars Conference for their comments on drafts of this Article. Thanks are also due to Signe Brunstad, Karen Copenhaver, Jonathan Franklin, David McGowan, Heather Meeker, and University of Washington School of Law students Sean Kellogg, Bill Snyder, and David Tilton for useful discussions. Jolene Lau Marshall provided valuable research assistance. Finally, I am grateful for research funding provided by the Shidler Center for Law, Commerce, and Technology.

"So, before we start changing [the GPL], we had better be aware how difficult this is going to be and of all the things that will come loose once we start shaking it."

Eben Moglen[1]

"In many ways, my only gripe with the GPL has been how many words it seems to need to say something very simple."

Linus Torvalds[2]

I. INTRODUCTION

The General Public License (GPL) enshrines a software hacker's[3] freedom to use code in important ways.[4] Hackers often refer to the GPL as the free software movement's "constitution."[5] Richard Stallman, founder of the Free Software Foundation (FSF),[6] wrote the most recent version of the GPL, version 2.0,

1. Peter Galli, *Rewriting GPL No Easy Task*, EWEEK, Feb. 2, 2005, *available at* http://www.eweek.com/article2/0,1759,1758752,00.asp (quoting Eben Moglen, Professor of Law and Legal History at Columbia Law Sch., Address at the OSDL Enterprise Linux Summit: GPL v3—Issues of Substance and Process (Feb. 1, 2005)).

2. Stephen Shankland, *Torvalds: A Solaris Skeptic*, CNET NEWS.COM, Dec. 21, 2004, http://news.com.com/Torvalds+A+Solaris+skeptic/2008-1082_3-5498799.html (quoting Linus Torvalds, creator of the Linux kernel).

3. Software developers who have a passion for programming are called "hackers." THE NEW HACKER'S DICTIONARY 233–34 (3d ed. 1996). Outside the software development community the term "hacker" often refers to a programmer who writes malicious code such as viruses and worms. *See id.* at 130, 234. However, serious programmers use the term "hacker" in a positive sense, as in: "I'm hacking some code to fix that bug." *See id.* at 231. Hackers call malicious programmers "crackers." *Id*. at 234. *See generally* STEVEN LEVY, HACKERS: HEROES OF THE COMPUTER REVOLUTION (1984) (describing hackers in the positive sense of the term).

4. For an explanation of these freedoms, see *infra* Part IV.

5. Richard Stallman & Eben Moglen, *GPL Version 3: Background to Adoption*, http://www.fsf.org/news/gpl3.html (last visited Nov. 11, 2005); *see also* The History of the GPL, http://www.free-soft.org/gpl_history (last visited Nov. 11, 2005) (explaining that the GPL "enables the development of *software by the people, of the people and for the people*"). The GPL is a license, of course, not a constitution, although hackers consider their licenses to be more like social compacts than contracts.

6. Lawrence Lessig, *The People Own Ideas!*, TECH. REV., June 2005, at 46, 48–49. Some say that the GPL is Stallman's most brilliant idea. *See, e.g.*, *id.*

back in 1991.[7] For a constitution, a fourteen-year-old document is young, but for a license, it is quite old. The revision process is finally underway, led by Stallman and Eben Moglen, FSF's general counsel.[8]

The release of GPL version 3.0 will be momentous for many reasons, but one reason stands out: The GPL governs much of Linux-based software[9] which is challenging the dominance of Microsoft's Windows platform.[10] Traditional computer industry players such as IBM, Hewlett-Packard, and Novell place Linux at the center of their business strategy.[11] New companies such as Red Hat, MontaVista Software, and MySQL have arisen to capitalize on the popularity of Linux.[12]

There are three particularly interesting aspects of the GPL revision: the process for changing it, the substance of the changes, and the form that the changes will take.

I have commented previously on the revision process for open source licenses.[13] Therefore, I will touch only briefly on the GPL revision process.[14] This Article explores the substantive

---

7. The History of the GPL, *supra* note 5.

8. *See* Paul Krill, *LinuxWorld: GPL Upgrade Due in 2007*, BIO-IT WORLD, Aug. 12, 2005, *available at* http://www.bio-itworld.com/newsitems/2005/Aug2005/08-12-05-news-gnu/view. Moglen estimates that a draft of GPL version 3.0 will be released sometime in late 2005 or early 2006. *Id*. Some have observed that this timing coincides nicely with the fifteenth anniversary of GPL 2.0. *See, e.g.*, *id.* (reporting that Eben Moglen felt it was "time for a change" and that "15 years is probably long enough"). Others have observed that this timing coincides strategically with Microsoft's next major update to the Windows operating system known as "Longhorn" or "Windows Vista." *See, e.g.*, Peter Galli, *GPL Could Put Heat on Microsoft*, EWEEK, Nov. 29, 2004, *available at* http://www.eweek.com/print_article2/0,1217,a=139964,00.asp (noting the similarity in timeframes). Stallman pledges to work on the revision until he is satisfied with the result. *See* Stallman & Moglen, *supra* note 5 (noting that as the creator and author of the GPL, Stallman has the "right to preserve its integrity as a work representative of his intentions").

9. *See* Stallman & Moglen, *supra* note 5 ("Linus Torvalds adopted GPL Version 2 for his operating system kernel, called Linux."). The Linux software is an operating system kernel, not a complete operating system, thus Linux must be combined with other software programs to make a complete operating system. MATT WELSH ET AL., RUNNING LINUX 1, 10 (3d ed. 1999). Linux is often packaged with Stallman's GNU software to form what is commonly called the GNU/Linux operating system. *See The GNU Operating System*, http://www.gnu.org (last visited Nov. 11, 2005) (describing the GNU/Linux system).

10. Brier Dudley, *Plugged in to Microsoft's Biggest Rival*, SEATTLE TIMES, Apr. 11, 2005, at C1 (describing Microsoft's competition with Linux).

11. *See, e.g.*, *id.* (discussing how Microsoft "woke up to the threat" Linux created and instigated a marketing and business strategy to face the threat).

12. *See* Steve Hamm, *Linux Inc.*, BUS. WK., Jan. 31, 2005, at 60 (discussing how distributors and software companies capitalize on Linux); MontaVista Software, http://www.mvista.com (last visited Nov. 11, 2005).

13. *See* Robert W. Gomulkiewicz, *De-bugging Open Source Software Licensing*, 64 U. PITT. L. REV. 75 (2002).

14. As this Article was going to press, the Free Software Foundation (FSF)

changes that the FSF is considering for GPL 3.0 and offers some approaches that the drafters could take. It then explains that the FSF intends to make only minimal changes to the GPL's form, and argues that failing to modify the GPL's form ignores one of the chief complaints about the GPL—that it is too wordy and unwieldy. Thus, the Article proposes a new form,[15] called the Simple Public License, which could serve as a model for a simplified GPL.[16]

As background, this Article begins with a brief software tutorial, a description of the principles of "free software," and a history of the GPL.

II. A QUICK SOFTWARE TUTORIAL

Software comes in two basic forms: object code and source code.[17] Programmers write software in source code form using a computer language such as Basic, C++, or Java.[18] Source code is human-readable code—it can be understood by any programmer proficient in the language.[19]

Programmers convert source code into object code using a tool called a compiler or interpreter.[20] Object code runs the computer.[21] Object code is machine-readable code—it consists of a series of ones and zeros that most humans cannot understand.[22]

---

announced that it would soon publish a document outlining the roadmap and process for providing input on the first draft of GPL 3.0. Peter Galli, *Coming Soon to a Kernel Near You: GPL 3*, EWEEK, Oct. 28, 2005, *available at* http://www.eweek.com/article2/0,1895,1879114,00.asp.

15. In the interest of full disclosure, I should mention that I served as an in-house licensing lawyer at Microsoft for ten years prior to joining the faculty of the University of Washington School of Law. Depending on your point of view, this either makes me particularly unqualified or particularly well-qualified to offer suggestions about the GPL. Indeed, the first time I wrote an article about open source software while I was still counsel at Microsoft, hacker commentators were surprised at my admiration for the open source movement but cautioned not to trust "Darth Vader." I do not work for Microsoft anymore; however, I still own some shares of stock in the company.

16. *See infra* Appendix I.

17. *See* Apple Computer, Inc. v. Franklin Computer Corp., 714 F.2d 1240, 1243 (3d Cir. 1983) (describing object code and source code); *see also* Robert W. Gomulkiewicz, *How Copyleft Uses License Rights to Succeed in the Open Source Software Revolution and the Implications for Article 2B*, 36 HOUS. L. REV. 179, 180–81 (1999) (defining object code and open source).

18. Gomulkiewicz, *supra* note 17, at 180–81.

19. *Id.*

20. *See Apple Computer*, 714 F.2d at 1243; Lotus Dev. Corp. v. Paperback Software Int'l, 740 F. Supp. 37, 44 (D. Mass. 1990).

21. *See Apple Computer*, 714 F.2d at 1243 (describing how object code is used by computers).

22. *Id.*

This "ones and zeros" nature of object code leads some people to call it binary code.

III. WHAT MAKES SOFTWARE "FREE" (OR NOT)?

What makes free software free? Surely not price, because free software does not always come free of charge.[23] Many companies have based significant businesses on free software.[24] Instead, free software means software that comes with certain rights: the right to study the source code, the right to run the software for any purpose, the right to change the software in any manner, and the right to distribute the software and any changes.[25]

The free software movement is part of a larger movement called the "open source software" movement.[26] Some hackers coined the phrase "open source software" because the free software movement had become associated with an antibusiness attitude, thus limiting its attractiveness to commercial enterprises.[27] Stallman and others of his philosophical persuasion do not like the change in emphasis.[28]

---

23. The FSF contends that "'[f]ree software' is a matter of liberty, not price." It likes to say, "think of 'free' as in 'free speech,' not as in 'free beer.'" RICHARD M. STALLMAN, *The Free Software Definition*, *in* FREE SOFTWARE, FREE SOCIETY: THE SELECTED ESSAYS OF RICHARD M. STALLMAN 41, 41 (Joshua Gay ed., 2002), *available at* http://www.gnu.org/philosophy/fsfs/rms-essays.pdf.

24. *See* Robert W. Gomulkiewicz, *Entrepreneurial Open Source Software Hackers: MySQL and Its Dual Licensing*, 9 COMP. L. REV. & TECH. J. 203, 205–07 (2004) (describing various ways in which software companies profit from the use of free software).

25. The above explanation is a paraphrase of the FSF's definition of the freedom associated with free software. *See* STALLMAN, *supra* note 23, at 41.

26. *See generally* STEVEN WEBER, THE SUCCESS OF OPEN SOURCE (2004) (describing where and how the free software movement fits into the "open source software" movement). The open source philosophy is outlined in the Open Source Definition. Open Source Initiative, The Open Source Definition, http://www.opensource.org/docs/definition.html (last visited Nov. 11, 2005).

27. *See* Open Source Initiative, History of the OSI, http://www.opensource.org/docs/history.html (last visited Nov. 11, 2005). Prominent hacker Eric S. Raymond gets credit for popularizing the "open source" terminology. Gomulkiewicz, *supra* note 24, at 204–05. Raymond has written important works explaining the inner-workings of what he calls the "hacker tribe," such as *The Cathedral and the Bazaar*, *The Magic Cauldron*, and *Homesteading the Noosphere*. *See* Eric's Random Writings, http://www.catb.org/~esr/writings (last visited Nov. 11, 2005) (listing Eric Raymond's writings).

28. *See, e.g.*, RICHARD M. STALLMAN, *Why "Free Software" is Better than "Open Source," in* FREE SOFTWARE, FREE SOCIETY: THE SELECTED ESSAYS OF RICHARD M. STALLMAN, *supra* note 23, at 55, 55–56 (explaining the differences between the free software movement and the open source movement); Gomulkiewicz, *supra* note 13, at 78–79 (describing the various reasons hackers believe that software should be released as open source or free software); Stallman & Moglen, *supra* note 5 (arguing that "free software . . . is the only ethically satisfactory form of software development"); Richard Stallman, Letter to the Editor, *"Free" versus "Open Source*," CNET NEWS.COM, Feb. 3,

Free and open source software (often known by the acronym FOSS)[29] stands in contrast to most traditional commercial software, which holds source code as a trade secret and provides limited (or no) rights to make or distribute derivative works.[30]

There is much debate among software developers about what to call software that is not FOSS.[31] Common labels are commercial software, proprietary software, and closed source software.[32] Each of these labels has its strengths and weaknesses. The problem with using the term "commercial" is that FOSS may be used in both commercial and noncommercial settings.[33] The issue with the term "proprietary" is that FOSS is protected by a proprietary right,[34]

2005, http://news.com.com/2009-1081_3-5562233.html (objecting to the use of open source terminology as applied to FSF projects because of the differing philosophies behind the free software movement and the open source movement). To be diplomatic, some now refer to software as free and open source software, or FOSS for short. *See, e.g.*, STRATEGIC TECHNOLOGY CONSULTING, FREE/OPEN SOURCE SOFTWARE (FOSS): PRIMER—A GENERAL INTRODUCTION, http://www.stcpl.com.au/downloads/Foss_Primer.pdf (last visited Nov. 11, 2005) (explaining what constitutes FOSS software).

29. It is worth noting that there are at least three senses in which the terminology "free software" or "open source" software is used. First, it may refer to a philosophy (i.e., Stallman's "free" as in "freedom"); second, it may refer to a method of software development (i.e., Raymond's "[g]iven enough eyeballs, all bugs are shallow," ERIC S. RAYMOND, THE CATHEDRAL AND THE BAZAAR 27 (2000)); third, it may refer to a set of license agreements (i.e., those approved by the Open Source Initiative or FSF).

30. *See* Gomulkiewicz, *supra* note 13, at 78 (explaining why it is common for developers to hold source code as a trade secret); *see also* MICROSOFT CORPORATION ANNUAL REPORT 17 (2005), *available at* http://www.microsoft.com/msft/ar.mspx (listing "unauthorized disclosure of source code" as a significant "Risk Factor," and explaining the importance of confidential source code to the company and the harms that could result from unauthorized disclosure).

31. Labeling is a serious matter. Hackers debate the aptness of the "free" and "open source" labels, as mentioned *supra* note 28, but labeling also carries great importance in the FOSS community's rivalry with traditional commercial software publishers such as Microsoft. *See* David McGowan, *SCO What? Rhetoric, Law and the Future of F/OSS Production*, 2–3, 14–15 (Univ. of Minn. Law Sch., Research Paper No. 04-9, 2004), *available at* http://papers.ssrn.com/abstract=555851 (detailing how rhetoric can be used to shape opinions in the FOSS debate); *see also* Dudley, *supra* note 10 (noting Microsoft's strategy of shifting the debate in its rivalry with Linux). The FOSS movement realizes that labels such as "closed source" and "proprietary" paint non-FOSS developers in a negative light. *See, e.g.*, McGowan, *supra*, at 3, 14–15 (providing other examples of the rhetoric used by FOSS developers to bolster their positions). Non-FOSS developers play the same game, of course, labeling FOSS as anti-intellectual property and unconstitutional. *See* McGowan, *supra*, at 21–23 (describing rhetoric and exaggerations used by non-FOSS developers to discredit the FOSS movement).

32. *See generally* MICHAEL A. CUSUMANO, THE BUSINESS OF SOFTWARE (2004) (describing various business models in the software industry).

33. *See* Free Software Foundation, Categories of Free and Non-Free Software, http://www.gnu.org/philosophy/categories.html (last updated June 20, 2005) (stating that "there is commercial free software, and there is non-commercial non-free software").

34. Lawyers tend to contrast "proprietary" with "non-proprietary," that is to say

namely copyright law.[35] In fact, as hackers like to say, you need copy*right* before you can have a copy*left*.[36] The term "closed source" is misleading in that many traditional commercial software publishers do provide access to their source code, and this trend is on the increase.[37]

A better label might be "Binary Use Software" or "BUS" for short.[38] This label contrasts traditional commercial software and FOSS on two levels: access to the software's source code and the nature of a licensee's right to do things with the software. In terms of access to source code, FOSS licensing provides free and open access to source code whereas BUS normally provides only access to the software in binary code form.[39] As to the rights licensed, FOSS licensing grants the licensee the right to freely copy, distribute, and create derivative works of the software whereas BUS licensing simply provides the licensee with the right to use the functionality of the software.[40]

"owned" versus "not owned." Hackers use the term "proprietary" in a different sense: to communicate whether access to software is controlled or not controlled. *See* Bradley M. Kuhn & Richard M. Stallman, Freedom or Power, http://www.gnu.org/philosophy/freedom-or-power.html (asserting that "[p]roprietary software is an exercise of power" and control).

35. *See* RICHARD M. STALLMAN, *What is Copyleft?*, *in* FREE SOFTWARE, FREE SOCIETY: THE SELECTED ESSAYS OF RICHARD M. STALLMAN, *supra* note 23, at 89, 89 (illuminating how copyright can be used to guarantee freedom).

36. Hackers use the term "copyleft" to illustrate that the GPL reverses the exclusivity of the copyright holder's rights by broadly licensing those rights. *See id.* (explaining that the process of changing from copyright to copyleft guarantees users' freedoms rather than taking them away); Gomulkiewicz, *supra* note 17, at 185–86 (explaining the difference in licensing terms between typical commercial software and open source software); Free Software Foundation, *supra* note 33 (describing copylefted software). See *infra* Part IV for a discussion of the importance of licensing for free software.

37. *See* Gomulkiewicz, *supra* note 13, at 78 & nn.22 & 25–26 (explaining that many commercial publishers, such as Microsoft, publish source code without confidentiality restrictions); *see also* Microsoft Corp., Microsoft Shared Source Initiative Overview, Oct. 18, 2005, http://www.microsoft.com/resources/sharedsource/Initiative/Initiative.mspx (describing how Microsoft is sharing source code with outside entities).

38. I admit that the Binary Use Software (BUS) label is not perfect. For one thing, BUS often comes with a license to create some derivative works (such as derivative works of clip art or software libraries) or to make extra copies (such as a copy for laptop use). Nonetheless, I think it creates a better "apples to apples" comparison than any of the other labels presently in use.

39. *See* Gomulkiewicz, *supra* note 17, at 181 (explaining the difference between software in binary form and open source software).

40. Whether licensees actually value the additional rights granted in FOSS licenses is a major point of debate between the FOSS and BUS communities.

IV. HOW COPYRIGHT AND LICENSING CREATE THE FOUNDATION FOR SOFTWARE FREEDOM

A copyright on code, combined with a license of the code, creates the legal framework that guarantees the freedom of free software.[41] Copyright law protects software in its source and object code forms.[42] The moment a hacker writes code, copyright applies, giving the hacker a bundle of exclusive rights, including the rights to copy, distribute, and create derivative works.[43] A hacker who believes that software should be free, however, does not want to hold these rights exclusively.

The hacker gives away the exclusive rights by licensing them.[44] Licensing gives hackers the power and flexibility to grant the rights that they equate with software freedom.[45] Stallman calls this "copyleft"—using a license to reverse the exclusive rights under copyright. Although some commentators have questioned the legality and enforceability[46] of free software

---

41. *See* Gomulkiewicz, *supra* note 17, at 185–89 (explaining that licensing is the legal force behind free and open source software); *see also* Debian, What Does Free Mean? or What Do You Mean by Free Software?, http://www.debian.org/intro/free (last visited Nov. 11, 2005) [hereinafter Debian, What Does Free Mean] ("To stay free, software must be copyrighted and licensed."). The Debian Project is a nonprofit organization dedicated to creating Linux-based software. Debian, About Debian, http://www.debian.org/intro/about (last visited Nov. 11, 2005). Recently, the city of Munich announced that it chose to implement Debian's Linux distribution on a large scale computer installation. Stephen Shankland, *Debian Wins Munich Linux Deal*, CNET NEWS.COM, Apr. 28, 2005, http://news.com.com/Debian+wins+Munich+Linux+deal/2100-7344_3-5689003.html.

42. Apple Computer, Inc. v. Franklin Computer Corp., 714 F.2d 1240, 1249 (3d Cir. 1983). *See generally* Robert W. Gomulkiewicz, *Legal Protection for Software: Still a Work in Progress*, 8 TEX. WESLEYAN L. REV. 445, 449–51 (2002) (describing how software is protected by copyright, patent, trademark, trade secret, and contract law).

43. *See* 17 U.S.C. § 106 (2000) (listing the exclusive rights of a copyright holder).

44. For a discussion of mass-market licensing in the software industry, see generally Robert W. Gomulkiewicz, *Getting Serious About User-Friendly Mass Market Licensing for Software*, 12 GEO. MASON L. REV. 687, 689–90, 692 (2004) [hereinafter Gomulkiewicz, *User-Friendly Licensing*]. *See also* Robert W. Gomulkiewicz, *The License Is the Product: Comments on the Promise of Article 2B for Software and Information Licensing*, 13 BERKELEY TECH. L.J. 891, 896–99 (1998) (describing the prevalence and significance of mass-market licenses in the software industry); Robert W. Gomulkiewicz & Mary L. Williamson, *A Brief Defense of Mass Market Software License Agreements*, 22 RUTGERS COMPUTER & TECH. L.J. 335 (1996) (explaining how end user license agreements (EULAs) are useful and efficient for mass-market distribution of software).

45. *See* Gomulkiewicz, *supra* note 13, at 79–80.

46. *See, e.g.*, Christian H. Nadan, *Open Source Licensing: Virus or Virtue?*, 10 TEX. INTELL. PROP. L.J. 349, 361–64 (2002) (advising that "a user is not bound by a contract of which he is not made aware," which could render some open source licenses unenforceable); Daniel B. Ravicher, *Facilitating Collaborative Software Development: The Enforceability of Mass-Market Public Software Licenses*, 5 VA. J.L. & TECH. 11, ¶¶ 42–55 (2000), http://www.vjolt.net/vol5/issue3/v5i3a11-Ravicher.html (describing the enforceability debate). *See generally* Jason B. Wacha, *Taking the Case: Is the GPL Enforceable?*, 21 SANTA CLARA COMPUTER & HIGH TECH. L.J. 451 (2005) (addressing GPL enforceability).

licensing, it remains the cornerstone of the free software movement.[47]

Two license forms dominate the free software movement: The Berkeley Software Distribution (BSD) style license and the GPL. The University of California at Berkeley created the original BSD-style license to accompany its distribution of the UNIX operating system, known as the Berkeley Software Distribution.[48] Variations of this license are used with popular free software programs such as the Apache web server. Hackers often debate the issue of whether the GPL or the BSD-style license provides more freedom for programmers. Although it is difficult to assess which license is more popular, some evidence indicates that hackers use the GPL the most.[49]

For some recent cases that have addressed GPL enforceability, see Landgericht [LG] [Trial Court] Apr. 2, 2004, No. 21 O 6123/04 (F.R.G.), *available at* http://www.jbb.de/judgment_dc_munich_gpl.pdf (ruling GPL legally valid and enforceable under German law); Welte v. Fortinet UK Ltd., 5 WORLD E-COM. & INTELL. PROP. REP. (BNA) 11 (Apr. 12, 2005) (reporting that a Munich court ruled that a British software company violated the terms of the GPL and noting the company agreed to modify its end user license); SCO Group, Inc. v. IBM Corp., No. 2:03-CV-294 (D. Utah Feb. 9, 2005) (mem.) (dismissing motion to allow more discovery to determine enforceability of SCO's purported copyrights in UNIX and copyright claims against IBM); and Complaint, Drew Techs. Inc. v. Soc'y of Auto. Engineers, Inc., No. 2:03-CV-74535 (E.D. Mich. Nov. 10, 2003) (alleging that the defendant violated the GPL license by distributing the program without the requisite copyright and license notices). *See also* Ingrid Marson, *Fortinet Settles GPL Violation Suit*, CNET NEWS.COM, Apr. 26, 2005, http://news.com.com/Fortinet+settles+GPL+violation+suit/2100-7344_3-5684880.html (reporting a court injunction against Fortinet, which was imposed until it complied with GPL); Ingrid Marson, *Defender of the Linux Faith*, CNET NEWS.COM, Mar. 18, 2005, http://news.com.com/Defender+of+the+Linux+faith/2100-7344_3-5625667.html [hereinafter Marson, *Defender of Linux*] (reporting on the work of Harald Welte to personally enforce violations of the GPL as they relate to Linux). Welte is a co-author of Linux; he wrote its firewall. Marson, *Defender of Linux*, *supra*. Welte set up the website http://www.gpl-violations.org to prevent companies from violating the terms of the GPL. *Id.*

47. *See* Gomulkiewicz, *supra* note 17, at 189–90 (describing the significance of licensing to the success of the free software movement). "A license is the authors [sic] way of allowing use of their creation (software in this case), by others, in ways that are acceptable to them." Debian, What Does Free Mean, *supra* note 41.

48. *See* PETER H. SALUS, A QUARTER CENTURY OF UNIX, 142–43 (1994); Marshall Kirk McKusick, *Twenty Years of Berkeley Unix: From AT&T-Owned to Freely Redistributable*, *in* OPENSOURCES: VOICES FROM THE OPEN SOURCE REVOLUTION 31, 31–33 (Chris DiBona et al. eds., 1999).

49. Wikipedia, GNU General Public License, http://en.wikipedia.org/wiki/GNU_General_Public_License (last visited Nov. 11, 2005) (reporting that as of April 2004 the GPL was used for nearly 75% of the projects listed on the Freshmeat website and 68.5% of the free or open source software projects listed on the SourceForge website). For a breakdown of the statistics, see Freshmeat, Statistics and Top 20, http://freshmeat.net/stats/rating/?expand=rating (last visited Nov. 11, 2005), and Sourceforge, Software Map, http://sourceforge.net/softwaremap/trove_list.php?form_cat=14 (last visited Nov. 11, 2005).

V. CREATION OF THE GPL

Richard Stallman created[50] separate licenses for the early versions of his GNU Emacs software,[51] GNU Debugger, and GNU Compiler Collection.[52] Stallman wanted to create a single license form, however, which could apply to any software project. Using terms from these prior licenses, he created a license that could "apply to any program without modification, no matter who is publishing it."[53] This became the GNU General Public License version 1.0.[54] One of the unique features of GPL 1.0 was that a copy of the license need not accompany the software licensed under it—"All that's needed is a brief notice in the program itself, to say that the General Public License applies."[55] In other words, software could be licensed simply by reference to the GPL standard form.[56]

---

50. Stallman seems to have written the GPL with the assistance of legal counsel, although he is generally credited with its authorship. *See* The History of the GPL, *supra* note 5 (""The actual document [GPL] consists of several pages of rather complicated legalbol that our lawyers said we needed.'" (quoting *What is Copyleft?*, GNU BULL., June 1988, *available at* http://www.gnu.org/bulletins/bull5.html)). Regardless of who wrote the words, there is no doubt that the GPL reflects Stallman's philosophy. *See* Brian W. Carver, *Share and Share Alike: Understanding and Enforcing Open Source and Free Software Licenses*, 20 BERKELEY TECH. L.J. 443, 443–48 (2005) (noting that the GPL implements Stallman's vision).

51. GNU is recursive for "GNU's Not UNIX." This came out of Stallman's disagreement with AT&T over its restrictive licensing of UNIX source code. Ganesh C. Prasad, The Practical Manager's Guide to Linux, http://www.li.org/papers/1999-pracmgr/Manager's-Guide-to-Linux.html (last visited Nov. 11, 2005). The license for Emacs is entitled the "Emacs General Public License." Emacs General Public License, http://www.free-soft.org/gpl_history/emacs_gpl.html (last visited Nov. 11, 2005).

52. Jack Schofield, *Inside IT*, GUARDIAN (London), Jan. 8, 2004, Guardian Life, at 16.

53. *The General Public License as a Subroutine*, *supra* note 53.

54. The History of the GPL, *supra* note 5; *see also* GNU General Public License Version 1, Feb. 1989, http://www.gnu.org/copyleft/copying-1.0.html.

55. *The General Public License as a Subroutine*, GNU BULL., Jan. 1989, *available at* http://www.gnu.org/bulletins/bull6.html.

56. There is some question about whether this method of licensing creates an enforceable contract. *See* Specht v. Netscape Commc'ns Corp., 306 F.3d 17, 32 (2d Cir. 2002) (holding where reference to an existing license was printed in a manner that tended to conceal the nature of acceptance, downloading the program did not manifest assent to license). An enforceable contract requires a meaningful opportunity to review the license terms and manifestation of assent. *Id.* at 28–35; *see also* Gomulkiewicz, *User-Friendly Licensing*, *supra* note 44, at 688 & n.7, 691–92 (discussing court decisions on mass-market licensing). The FSF's general counsel, Eben Moglen, sidesteps this issue by arguing that the GPL is not a contract; rather, he argues it is a pure license and as such does not need to adhere to contract formation formalities. Eben Moglen, Free Software Matters: Enforcing the GPL, I (Aug. 12, 2001), http://emoglen.law.columbia.edu/publications/lu-12.pdf; *see also* LAWRENCE ROSEN, OPEN SOURCE LICENSING: SOFTWARE FREEDOM AND INTELLECTUAL PROPERTY LAW 51–71 (2005) (discussing whether open source licenses are contracts or pure licenses and the implications of the distinction). *But see* Wacha, *supra* note 46, at 458, 481–83 (stating that "[t]he GPL likely is a contract," but as a license it is

The January 1991 issue of the *GNU Bulletin* discussed the prospects for a new version of the GPL that would apply to GNU library software (this became the Library GPL, now called the Lesser GPL) as well as a new version of the regular GPL: "We will also be releasing a version 2 of the ordinary GPL. There are no real changes in its policies, but we hope to clarify points that have led to misunderstanding and sometimes unnecessary worry."[57]

VI. WHY STALLMAN CARES ABOUT LICENSING AND THE ROLE THE GPL PLAYS IN SOFTWARE FREEDOM

Most software users care little or nothing about the terms of software licenses.[58] Stallman, however, learned that ignoring a license can have serious consequences. He experienced the frustration of being accused of violating a license in his development of the GNU Emacs software.[59]

Stallman created the first Emacs software in 1975.[60] James Gosling, a hacker who is famous for many software innovations including the Java language, wrote the first C language version of Emacs, and reportedly allowed free distribution of his Gosling Emacs source code.[61] Eventually, Gosling sold his rights in

---

difficult to invalidate); Margaret Jane Radin, Address before the Association of American Law Schools (Jan. 6, 2005) (on file with the Houston Law Review) (concluding that open source licenses are contracts). The question of enforceability may depend on the context. Between hackers, industry custom may be sufficient—hackers are fully aware that the GPL's terms apply to any code that claims to be GPL licensed. *See* Wacha, *supra* note 46, at 491 ("A software engineer who is well acquainted with the existence of the GPL might have trouble arguing with a straight face that she was unaware that, for instance, the FSF intends the GPL to apply to Linux."). A detailed discussion of the pure license versus contract debate is beyond the scope of this Article. It is worth observing, however, that the FOSS license most resembling a pure license in the sense of doing little more than granting permission is the BSD-style license. In contrast, the GPL requires the licensee to agree to several important terms—whether these are covenants or license conditions or whether this distinction even matters is an open question.

57. *GNUs Flashes*, GNU BULL., Jan. 1991, *available at* http://www.gnu.org/bulletins/bull10.html. Since the release of this version of the GPL, most clarifications can be found on the FSF's website in its Frequently Asked Questions section. Free Software Foundation, Frequently Asked Questions about the GNU GPL, http://www.fsf.org/licenses/gpl-faq.html (last visited Nov. 11, 2005).

58. *See generally* Gomulkiewicz, *User-Friendly Licensing*, *supra* note 44 (describing how users typically ignore software licenses and proposing solutions).

59. *See* The History of the GPL, *supra* note 5 (noting that GNU Emacs software is an extensible text editor similar to those Stallman developed in the past and telling the story behind Stallman's experience with using free code that turned into proprietary code).

60. *Id.*

61. *See id.* ("'Gosling originally had set up his Emacs and distributed it free and gotten many people to help develop it, under the expectation based on Gosling's own words in his own manual that he was going to follow the same spirit that I started with

Gosling Emacs to a company called UniPress.[62] Meanwhile, Stallman received a copy of Gosling Emacs source code from a friend and used it in his initial version of his GNU Emacs.[63] UniPress told Stallman that he was not allowed to use Gosling Emacs.[64] Stallman's friend lost the message from Gosling granting him rights in Gosling's Emacs, and Gosling denied ever having given permission.[65] Consequently, Stallman was forced to rewrite the Gosling Emacs code that was contained in GNU Emacs.[66]

This experience convinced Stallman that he needed to find a way to keep free software free. In other words, once a programmer released code as free code, Stallman wanted to guarantee that the code remained available for hackers to exercise the four basic freedoms described previously.[67] The GPL became the mechanism for maintaining software freedom.[68]

The GPL accomplishes this goal by granting full rights to create derivative works of a program with one important condition: that the licensee also grants full rights to create derivative works of any program based on the original program, at no charge, and to pass this obligation on.[69] Under the GPL, a programmer may modify Software Program 1 to create Software Program 2, but if the programmer does so and distributes

---

the original Emacs . . . .'" (quoting Richard M. Stallman, Lecture at the Royal Institute of Technology (Oct. 30, 1986) (transcript available at http://www.gnu.org/philosophy/stallman-kth.html))).

62. *Id.*

63. Ultimately Stallman says he included only a "small fraction" of Gosling's code in GNU Emacs. Stallman, *supra* note 61.

64. The History of the GPL, *supra* note 5.

65. Stallman, *supra* note 61 ("[T]hey . . . put up a message on the network saying that I wasn't allowed to distribute the program. They didn't actually say that they would do anything, they just said that it wasn't clear whether they might ever someday do something. And this was enough to scare people so that no one would use it any more, which is a sad thing.").

66. *Id.* ("So I was forced to rewrite all the rest that remained, and I did that, it took me about a week and a half. So they won a tremendous victory.").

67. *See supra* note 25 and accompanying text.

68. Eric S. Raymond doubts that a license is necessary for this purpose anymore. *See* Federico Biancuzzi, *ESR: "We Don't Need the GPL Anymore*," O'REILLY ONLAMP.COM, June 30, 2005, http://www.onlamp.com/pub/a/onlamp/2005/06/30/esr_interview.html ("It's 2005 not 1985. We've learned a lot in the past 20 years. The fears that originally led to the reciprocity stuff in GPL are nowadays, at least in my opinion, baseless. People who do what the GPL tries to prevent . . . wind up injuring only themselves."). *But see* Federico Biancuzzi, *RMS: The GNU GPL is Here to Stay*, O'REILLY ONLAMP.COM, Sept. 22, 2005, http://www.onlamp.com/lpt/a/6222 (arguing that the GPL does a good job of preventing the loss of software freedom).

69. GNU General Public License, *in* FREE SOFTWARE, FREE SOCIETY: THE SELECTED ESSAYS OF RICHARD M. STALLMAN, *supra* note 23, at 195, § 2(b), at 196; *see* Gomulkiewicz, *supra* note 13, at 88–92 (discussing the complexity of GPL section 2(b)).

Software Program 2, any other programmer must be given the right to modify Software Program 2 to create Software Program 3 through infinity. If the programmer does not agree to this condition, then he or she has no rights to create derivatives of Software Program 1. Any unlicensed derivatives infringe the copyright in Software Program 1.

VII. GPL 3.0: THE REVISION PROCESS

The GPL is showing its age, as any fourteen-year-old license would. While there is no doubt the GPL needs to be fixed, revising the GPL will be particularly complex because it represents the "*de facto* constitution for the Free Software movement."[70] The GPL's stakeholders range from individual hacker hobbyists to corporations such as IBM and countries such as China. When Stallman created GPL 2.0 it was essentially a license for a small group of serious hackers;[71] currently, as Moglen observes, "'There are billions of dollars riding on this now; lots of people's livelihoods depend on us getting this right.'"[72] The challenge posed by updating the GPL will be more akin to the challenge of amending the U.S. Constitution or negotiating a treaty than updating a typical software license.[73]

The process for updating the GPL is unfolding.[74] Moglen and Stallman have been discussing proposed changes at public events and in private correspondence with a select number of hacker leaders.[75] They plan to seek widespread public

---

70. The History of the GPL, *supra* note 5; *see also* Galli, *supra* note 1 (quoting Eben Moglen as stating "'In my career of almost 20 years as an educator, I have never faced a problem as complex as this.'"). Moglen's new Software Freedom Law Center will play a role in collecting comments on draft GPL 3.0. *See* Stephen Shankland, *Lawyers Ride Shotgun for Open Source*, CNET NEWS.COM, Jan. 31, 2005, http://news.com.com/Lawyers+ride+shotgun+for+open+source/2100-7344_3-5557962.html.

71. *See* Galli, *supra* note 14 (interview with Con Zymaris, CEO of Cybersource Pty. Ltd.) (discussing how GPL 2.0's "perceived importance was relatively minor due to the minimal spread of free software"); Stallman & Moglen, *supra* note 5 (describing the founding of the free software movement).

72. Galli, *supra* note 1, at 3 (quoting Eben Moglen).

73. *See* Gomulkiewicz, *User-Friendly Licensing*, *supra* note 44, at 699–705 (describing the challenges of improving typical mass-market software licenses).

74. Given the current significance of the GPL to a wide variety of constituencies, I have proposed the creation of an open source license standards organization to update and maintain important open source licenses such as the GPL. *See* Gomulkiewicz, *supra* note 13, at 96–103 (outlining the role of the proposed organization).

75. Michael Singer, *Insider Hints at GPL Changes*, INTERNETNEWS.COM, Apr. 7, 2005, http://www.internetnews.com/dev-news/article.php/3495981 (noting that Sleepycat Software CEO Mike Olson is one of the select members of the free-software community being consulted about revisions to the GPL).

comment.[76] Although they have not set a firm date for finalizing changes, Moglen and Stallman anticipate the updating process will take at least a year.[77] However, one important feature of the process is crystal clear: Stallman will make the final decision.[78]

VIII.GPL 3.0: THE SUBSTANTIVE ISSUES

The GPL revision will address a number of substantive issues.[79] This Part explores some of the major issues.

A. A Few Important Definitions to Fix

1. The Heart of Copyleft—GPL Section 2(b). GPL section 2(b) implements the free software movement's "share back" objective. It does so as follows: In order to gain the right to create a derivative of a GPL-licensed program, a programmer must agree to give back to the community by causing "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

---

76. *See* Galli, *supra* note 1 ("Everyone seeking input will be given a chance to comment and propose changes to the license . . . ."); *see also* Gomulkiewicz, *supra* note 13, at 102–03 (urging FOSS developers to seek widespread commentary and participation by all stakeholders, particularly those in the legal field, in updating key FOSS licenses such as the GPL).

77. Galli, *supra* note 1.

78. In this sense, the process will not be like a constitutional convention at all. "'Like it or not, Mr. Stallman's decision on the license is final.'" *Id*. (quoting Eben Moglen). The *Background to Adoption* puts it this way: "Stallman remains the GPL's author, with as much right to preserve its integrity as a work representative of his intentions as any other author or creator. Under his guidance, the Free Software Foundation, which holds the copyright of the GPL, will coordinate and direct the process of its modification." Stallman & Moglen, *supra* note 5; *see also* Peter Galli, *Moglen: GPL 3.0 Rewrite Drive Is No Democracy*, EWEEK, Nov. 2, 2005, *available at* http://www.eweek.com/article2/0,1759, 1881088,00.asp. This is a powerful point, but what will happen if the FSF produces a GPL 3.0 that is unacceptable to key constituencies? Some hackers may stick to GPL 2.0. Others could take the bold step of creating an alternative license based on GPL ideas. But it is unlikely things will come to that because even though debate between hackers is ruthless, they are adept at reaching consensus to avoid forking. *See* WEBER, *supra* note 26, at 64 (describing the practice of forking and how rare a major fork is among hackers).

79. *See* Stephen Shankland, *Sprucing Up Open Source's GPL Foundation*, CNET NEWS.COM, Dec. 23, 2004, http://news.com.com/Sprucing+up+open+sources+GPL+ foundation/2100-7344_3-5501561.html (chronicling the likely changes to the GPL in an effort to modernize it). The issues include: definition of "works based on" a GPL-licensed program; definition of "distribution" when software functionality is delivered as a web service; international intellectual property issues; implied patent licensing; assertion of patents; term and termination; implications of trusted computing; DMCA; license compatibility between the GPL and other open source and proprietary licensed code. *See id.*

the terms of this License."[80] The net effect of this language is to require a developer who distributes code based on a GPL-licensed program, to deliver the source code, and to allow anyone to create derivative works of it, free of charge.[81]

Two aspects of this wording have proven difficult to interpret. First, when is software "derived from"[82] or when does it "contain" GPL-licensed software? Second, when has software been "published" or "distributed"? The answers to these questions matter because they dictate whether the programmer is required to deliver the program's source code and license derivative works. If a developer's program has never been published or distributed, or if the developer's program is neither derived from nor contains GPL-licensed code, then section 2(b) does not come into play.

Here are some situations that have puzzled programmers:[83]

- Is a program that dynamically links to another program "derived from" or does it "contain" the program?
- When a program uses services of another program, such as an operating system kernel, is the program "derived from" or does it "contain" the program?[84]
- If a firm participates in a joint venture or a multinational corporation ships code to its subsidiaries, has it distributed or published the code?

---

80. GNU General Public License, *supra* note 69, § 2(b), at 196.

81. *See* Gomulkiewicz, *supra* note 13, at 88–90 (analyzing the scope of section 2(b) and its requirement that developers publish their source code for free use).

82. A frequently cited issue on the agenda for GPL 3.0 is international copyright. A primary concern is the international diversity in defining "derivative works." *See* Mikko Välimäki, *GNU General Public License and the Distribution of Derivative Works*, J. INFO. L. & TECH., 2005, http://www2.warwick.ac.uk/fac/soc/law/elj/jilt/2005_1/valimaki (explaining the differences between the U.S. definition of derivative works and international treatment of the derivative works concept).

83. *See* Greg R. Vetter, *"Infectious" Open Source Software: Spreading Incentives or Promoting Resistance?*, 36 RUTGERS L.J. 53, 88–94 (2004) (providing a broader discussion of problems confronted by programmers when they attempt to interpret section 2(b)).

84. See Linus Torvalds's clarifying note to the GPL as applied to Linux:

> NOTE! This copyright does *not* cover user programs that use kernel services by normal system calls—this is merely considered normal use of the kernel, and does *not* fall under the heading of "derived work". Also note that the GPL below is copyrighted by the Free Software Foundation, but the instance of code that it refers to (the Linux kernel) is copyrighted by me and others who actually wrote it.

GNU General Public License (1991), http://www.linux.de/linux/gnu.html.

- If a firm runs a web business on software derived from GPL-licensed code, should the web business be required to share the source code?[85]

2. Clarifying GPL Section 2(b). Programmers hope GPL 3.0 will clarify section 2(b). There are several approaches the GPL's revisers could use to do so.

As an interpretive aid, the GPL's authors could tie section 2(b) more clearly to the Copyright Act's terminology, in particular its definitions of "publish,"[86] "distribute,"[87] and "derivative work."[88] If the GPL's authors take this approach, they should eliminate the word "contain" in section 2(b) because it is not a term of art under copyright law and use of "contain" in section 2(b) has caused confusion: Is the GPL merely describing a subset of derivative works or is it referring to collective works?[89]

A potential advantage[90] of tying the GPL more closely to the Copyright Act nomenclature is that the GPL could draw on

85. If the GPL's authors decide to include this case as GPL code under section 2(b), one suggestion is to work with the "public performance" right under copyright law rather than the "distribution" right. *Compare* 17 U.S.C. § 106(3) (2000) (defining the distribution right), *with* 17 U.S.C. § 106(5) (defining the public performance right).

86. 17 U.S.C. § 101; *see also* Getaped.com, Inc. v. Cangemi, 188 F. Supp. 2d 398, 400–01 (S.D.N.Y. 2002) (discussing cases on "publication" of works).

87. 17 U.S.C. § 106(3).

88. 17 U.S.C. §§ 101, 106(2); *see also* Lydia Pallas Loren, *The Changing Nature of Derivative Works in the Face of New Technologies*, 4 J. SMALL & EMERGING BUS. L. 57, 62 (2000) (elaborating on a copyright holder's privilege to control the creation of derivative works).

89. *See* 17 U.S.C. § 101 (defining a "collective work" as one "in which a number of contributions, constituting separate and independent works in themselves, are assembled into a collective whole."). If section 2(b) is also referring to collective works, it should use that term as well as the term "derivative work." I am not suggesting that either interpretation is more plausible, only that the authors should clarify which interpretation they mean. *See* Gomulkiewicz, *supra* note 13, at 88–92 (analyzing whether GPL section 2(b) applies to derivative works and noting the "murky" intent of the GPL); *see also* ROSEN, *supra* note 56, at 118 (pointing out that GPL section 2 may only apply to derivative works and not to collective works).

90. Another advantage cited by my student Sean Kellogg is that to the extent the GPL purports to be a copyright license rather than a contract, the license should conform closely to the copyright holder's exclusive copyrights.

the cases that construe the terms "publish,"[91] "distribute,"[92] and "derivative work"[93] in the context of software source code. These cases could serve as an interpretive guide for users of the GPL 3.0.[94] Unfortunately, the case law in this area is sparse and more perplexing than illuminating, which degrades its usefulness as a tool to interpret contractual language.[95]

Alternatively, the GPL's authors could purposely choose to *avoid* Copyright Act terms of art.[96] They could use programming terms rather than legal terms. For example, the GPL could continue to use the "work based on a program" phrasing, but

---

91. *See, e.g.*, Computer Assocs. Int'l v. Quest Software, Inc., 333 F. Supp. 2d 688, 698–99 (N.D. Ill. 2004) (discussing the terms "publication" and "limited publication" in a software context); Telecomm Technical Servs., Inc. v. Siemens Rolm Commc'ns, Inc., 66 F. Supp. 2d 1306, 1322–23 (N.D. Ga. 1998) (analyzing the issue of when "a distribution is considered a 'limited publication' for copyright purposes" (citations omitted)). Just as the authors of the GPL would do well to eliminate the term "contain," the same is true of the term "publish," which is subsumed in the exclusive right to distribute. *See* 17 U.S.C. § 106(3); H.R. REP. NO. 94-1476, at 62 (1976), *as reprinted in* 1976 U.S.C.C.A.N. 5659, 5675–76 (discussing public distribution and noting that § 106(3) establishes the exclusive right of publications).

92. *See, e.g.*, Hotaling v. Church of Jesus Christ of Latter-Day Saints, 118 F.3d 199, 203 (4th Cir. 1997) (holding that the addition of a work to a library's collection is dissemination to the public and sufficient to establish "distribution"); Nat'l Car Rental Sys., Inc. v. Computer Assocs. Int'l, Inc., 991 F.2d 426, 434 (8th Cir. 1993) (noting that "the distribution right is only the right to distribute *copies*" and does not extend to the function of computer software); *see also* 17 U.S.C. § 106(3) (defining the right to "distribute copies or phonorecords of [a] copyrighted work").

93. *See, e.g.*, Dun & Bradstreet Software Servs., Inc. v. Grace Consulting, Inc., 307 F.3d 197, 212 (3d Cir. 2002) (defining derivative work "as 'a new created work based on the original copyright work'" (quoting the jury instructions given by the trial court)); Micro Star v. Formgen Inc., 154 F.3d 1107, 1110 (9th Cir. 1998) ("[A] deriviative work must exist in a concrete or permanent form, . . . and must substantially incorporate protected material from the preexisting work . . . ." (internal quotation marks and citations omitted)); Lewis Galoob Toys, Inc. v. Nintendo of Am., Inc., 964 F.2d 965, 967–68 (9th Cir. 1992) (concluding that a derivative work does not require fixation); Playmedia Sys., Inc. v. Am. Online, Inc., 171 F. Supp. 2d 1094, 1105–06 (C.D. Cal. 2001) ("[E]ven if the subsequent program borrowed very little of the previous program, a court could still conclude that the subsequent program was a 'derivative' if it was comprised mostly of the previous program.").

94. This approach is not groundbreaking—if a court interpreted the GPL today, it is very likely that the court would look to cases construing the Copyright Act to help it construe similar terms in the GPL. *See* Wacha, *supra* note 46, at 486–87 (recognizing that courts may defer to copyright law as a guide).

95. *See, e.g.*, Loren, *supra* note 88, at 92–93 (indicating that copyright case law regarding the definition for derivitive works "is confusing and, at times, contradictory"); Sean Hogle, *Unauthorized Derivative Source Code*, COMPUTER & INTERNET LAW., May 2001, at 1, 6–8 (noting that "commentators have condemned the expansive breadth courts have accorded to the definition of derivative works"). *See generally* Ralph S. Brown, *The Widening Gyre: Are Derivative Works Getting Out of Hand?*, 3 CARDOZO ARTS & ENT. L.J. 1 (1984) (discussing the cases and commentaries on developments in the body of law encompassing copyright).

96. *Cf. Playmedia Sys.*, 171 F. Supp. 2d at 1104–06 (describing the parties' intent with the phrase "in conjunction with" rather than "derivative work").

make it clear that the term is not synonymous with a Copyright Act "derivative work."[97] From there, parties using the GPL will have to wrestle with the meaning of the GPL's words in a given case relying on industry custom and other traditional methods of construing ambiguous contractual language.

Rather than simply allowing the words to speak for themselves, however, the GPL's authors could provide some assistance. The FSF already does this to a degree through its Frequently Asked Questions (FAQ) document.[98] The FAQ, as the name suggests, provides responses to frequently asked questions about the GPL.

The difficulty with using the FAQ to interpret the GPL is that once a programmer uses the GPL template, it is the programmer and his or her licensee's intended meaning that controls, not that of the FSF. Linus Torvalds's note clarifying what the GPL template means when licensing Linux provides a famous illustration of this point.[99] That is not to say that the FAQ is irrelevant. It is probably fair to say that many programmers deploy the GPL with full knowledge of and agreement with the FAQ's illumination of the GPL.[100] The trick is in knowing that this is the case in any particular license. Did the licensor agree with the FAQ? What about the licensee? Did they agree with all the FAQ or only parts?

The FSF could make the FAQ more useful by turning it into a collection of illustrations.[101] The GPL could explicitly reference these illustrations. In this era of electronic contracts, it would be trivial to provide a hypertext link from the GPL to the illustrations.[102] Indeed, users of the GPL template could link to the specific illustrations that they agree with rather than the whole set provided by the FSF. The FSF's illustration bank could

---

97. To implement this suggestion, the GPL's authors would eliminate the tie between "works based on the Program" and "derivative work" in GPL section 1. *Cf.* Loren, *supra* note 88, at 59 (coining the term "integrated work" to describe works that digitally reference preexisting works but do not copy preexisting material, so as to "avoid confusion with the legally significant term 'derivative works'").

98. Free Software Foundation, *supra* note 57.

99. *See supra* note 84 (clarifying the GPL's application to the Linux kernel).

100. Torvalds recognizes this implicitly because his clarifying note is responding to a conflicting interpretation in the FSF's FAQ. Gomulkiewicz, *supra* note 13, at 91 n.121.

101. For an example of this device, see RESTATEMENT (SECOND) OF CONTRACTS (1979).

102. *See* Gomulkiewicz, *User-Friendly Licensing*, *supra* note 44, at 705 (urging license drafters to use tools such as hypertext linking to detailed explanations of terms and illustrations to improve readability and usability of mass-market licenses).

grow and evolve over time. Other organizations could provide illustrations as well.[103]

B. Some Problems with Patents

Most hackers hate software patents. Stallman sees patents as one of the greatest threats to the free software movement.[104] Some corporate boosters of free software, most notably patent powerhouse IBM, have pledged to use their patents to promote rather than impede free software.[105] Nonetheless, hackers remain wary.[106]

Patents play into the GPL in two ways. First, does the GPL grant rights under the licensor's patents as well as its copyrights? Although the GPL's grant language speaks mainly in terms of copyrights, is there also an implied right under patents?[107] If there is an implied patent license, what is its scope?

---

103. The Open Source Initiative would be an obvious candidate. *See* Open Source Initiative, Licensing, http://www.opensource.org/licenses (last visited Nov. 11, 2005) (providing a collection of approved licenses along with guidance for their use).

104. Richard S. Stallman, *The GNU Operating System and the Free Software Movement*, *in* OPENSOURCES: VOICES FROM THE OPEN SOURCE REVOLUTION, *supra* note 48, at 53, 67–68.

105. *See, e.g.*, IBM, IBM Statement of Non-Assertion of Named Patents Against OSS, http://www.ibm.com/ibm/licensing/patents/pledgedpatents.pdf (last visited Nov. 11, 2005) (pledging free use of five-hundred patents to promote open source software); Red Hat, Inc., Statement of Position and Our Promise on Software Patents, http://www.redhat.com/legal/patent_policy.html (last visited Nov. 11, 2005) (promising to allow infringement of their patents as long as the software is FOSS); Stephen Shankland, *Nokia: Linux Kernel May Use Our Patents*, CNET NEWS.COM, May 25, 2005, http://news.com.com/Nokia+Linux+kernel+may+use+our+patents/2100-7344_3-5720696.html (noting that Nokia promised free use of its patented technology in the Linux kernel); Stephen Shankland, *Novell Vows Patent Defense of Open Source*, CNET NEWS.COM, Oct. 12, 2004, http://news.com.com/Novell+vows+patent+defense+of+open+source/2100-7344_3-5406571.html (noting Novell's vow to use its patents to defend open source software). Sun Microsystems offered the unfettered use of 1600 patents with software licensed under its Community Development and Distribution License. Stephen Shankland, *Sun: Patent Use OK Beyond Solaris Project*, CNET NEWS.COM, Jan. 31, 2005, http://news.com.com/Sun+Patent+use+OK+beyond+Solaris+project/2100-7344_3-5557658.html.

106. Bruce Perens, *Perspective: The Open-Source Patent Conundrum*, CNET NEWS.COM, Jan. 31, 2005, http://news.com.com/The+open-source+patent+conundrum/2010-1071_3-5557340.html (observing the limited scope of the Sun and IBM patent grants and commenting that the threat of patent infringement lawsuits still exists). Perens wrote the influential *Open Source Definition* which defines the criteria for an open source license. *See* Bruce Perens, *The Open Source Definition*, *in* OPENSOURCES: VOICES FROM THE OPEN SOURCE REVOLUTION, *supra* note 48, at 171; *see also* Stephen Shankland, *Open-Source Allies Go on Patent Offensive*, CNET NEWS.COM, Aug. 11, 2005, http://news.com.com/Open-source+allies+go+on+patent+offensive/2100-7344_3-5827844.html (detailing the threat of infringement suits by Microsoft, but pointing out efforts to counter patent threat to open-source developers, such as an open-source patent pool).

107. *Compare* GNU General Public License, *supra* note 69, *with* Mozilla Public License Version 1.1, § 2.1, http://www.mozilla.org/MPL/MPL-1.1.html (last visited Nov. 11, 2005).

In particular, does it apply to hardware-software combinations? Does it apply to derivative works?

Second, what happens if someone asserts patent rights against the licensee of GPL-licensed code? The GPL says that the GPL terminates if the licensee cannot continue to abide by its terms "as a consequence of a court judgment or allegation of patent infringement."[108] As a solution, the GPL could adopt alternatives offered by other open source licenses such as the Mozilla Public License (MPL), IBM's Common Public license (CPL), or the Open Source License (OSL) written by former Open Source Initiative general counsel Lawrence Rosen.[109]

The MPL, for example, states that if a MPL licensee sues any upstream licensor over patents *related to* the licensor's software, then the licensee loses both its patent and copyright licenses.[110] In addition, if a MPL licensee sues any upstream licensor over patents *unrelated to* the licensor's software, the licensee loses its patent licenses.[111] The latter provision enables parties to place all patent "weapons" back on the table in patent litigation.

C. Other Issues on the Table

All the talk about a new version of the GPL has unearthed a wish list of issues that GPL 3.0 could address. This Article will not explore those issues, but below is a representative list:

- Internationalizing the GPL: This would entail revising the GPL so that it fits more comfortably with copyright law from a variety of jurisdictions.[112]

---

108. *See* GNU General Public License, *supra* note 69, § 7, at 198.

109. *See* ROSEN, *supra* note 56, at 179–227 (explaining each part of the Open Source License and providing comparisons to other template licenses). Hacker Eric S. Raymond hopes GPL 3.0 can do even more: "We need to find some way to monkey-wrench the awful, broken software-patent oligopoly before it does more serious damage . . . . If GPL (version) 3 can help do that, it would be extremely valuable." Shankland, *supra* note 79 (internal quotation marks omitted) (quoting Eric S. Raymond, president emeritus and a founder of Open Source Initiative).

110. Mozilla Public License Version 1.1, *supra* note 107, § 8.2(b).

111. *Id.*

112. *See* Stallman & Moglen, *supra* note 5 (noting the global reach of the GPL and the need to ease internationalization difficulties to achieve the ideal of a global copyright license). Primary concerns would be the difference in the definition of derivative works and treatment of moral rights. On the issue of moral rights, see *U.S. Adherence to the Berne Convention: Hearing on the Implications, Both Domestic and International, of U.S. Adherence to the International Union for the Protection of Literary and Artistic Works Before the Subcomm. on Patents, Copyrights and Trademarks of the S. Comm. on the Judiciary*, 99th Cong. app. at 458–67 (1987), and 3 MELVILLE B. NIMMER & DAVID NIMMER, NIMMER ON COPYRIGHT § 8D.02[D][2] (2005). *See also* IRINI A. STAMATOUDI, COPYRIGHT AND MULTIMEDIA PRODUCTS: A COMPARATIVE ANALYSIS 160–64 (2002)

It could also involve providing non-English language translations of the GPL.

- Trusted Computing: There is concern among hackers that, in the name of creating more secure software systems, BUS licensors will create software that will not allow software created by the FOSS community to interoperate with it. For example, a BUS operating system would only allow access if an application was digitally signed in a certain way. In the diabolical case, that operating system could refuse to accept calls from any application licensed under the GPL.
- Digital Millennium Copyright Act (DMCA): Some people in the FOSS community would like to find a way for the GPL to carve back the effects of the DMCA on the use of technological measures to control software.

IX. GPL 3.0: HACKING A SIMPLER GENERAL PUBLIC LICENSE FORM

In creating GPL 3.0, Stallman seems committed to minimizing changes to its form. He hopes changes to the length will not exceed ten percent, and he plans to make the license appear as similar as possible to GPL 2.0.[113] This approach misses a golden opportunity to improve the GPL in an important way.

The GPL in its current form is not the epitome of clarity. Some say the wording is not artful, and some say it is artfully ambiguous so as to sweep as much code into its code-freeing license condition as possible.[114] Regardless, the GPL has known bugs.[115] Simplifying the GPL may be the most valuable revision that could be made to it. Some leading figures in the free software movement have few quibbles about the GPL other than its wordiness.[116]

(discussing the varied treatment of moral rights in Europe as they relate to software).

113. Galli, *supra* note 1.

114. *See, e.g.*, Richard Raysman & Peter Brown, *Using Open Source Code in Proprietary Products*, N.Y.L.J., Dec. 11, 2001, at 3 (stating that while portions of the GPL may be ambiguous, the provisions on modification and distribution seem directed at ensuring that GPL-licensed software remains free software that cannot be made proprietary with downstream licenses).

115. *See* Gomulkiewicz, *supra* note 13, at 83–92 (listing some of the GPL's bugs).

116. According to Linus Torvalds: "'In many ways, my only gripe with the GPL has been how many words it seems to need to say something very simple.'" Shankland, *supra* note 2. "'I don't think the GPL is perfect, and one of my issues has been how verbose it is.'" Peter Galli, *Torvalds: GPL Needs Minor Work*, EWEEK, Nov. 29, 2004, *available at* http://

In an attempt to stimulate discussion about what could be done to simplify the GPL, the Appendix contains a proposed license.[117] I call this the Simple Public License ("SimPL").[118] The SimPL revises the form but does not intend to alter the substance of GPL 2.0.

X. CONCLUSION

The FSF plans to update the venerable GPL. Previous GPL revisions have passed virtually unnoticed outside the hacker community, but this one is different. Since the last revision, many lawyers, government officials, businesspeople, and software programmers have developed an interest and a stake in GPL 3.0. The FSF says that it welcomes input. Hence, this Article proposes that the GPL's form can be simplified dramatically, and according to at least some hackers, this type of update could be the most useful one of all.[119]

---

www.eweek.com/article2/0,1759,1731874,00.asp (quoting Linus Torvalds).

117. *See infra* Appendix I (containing the Simple Public License ("SimPL") form); *see also infra* Appendix II (annotating the SimPL and explaining the rationale for its provisions).

118. The SimPL is designed to provide a simple copyleft.

119. My license for the SimPL license form is simply this: "***You may do anything that you want with it.***" To avoid confusion, however, if you change the license form in any way, then you may not call your license the Simple Public License or the SimPL. My license differs from the license for the GPL 2.0 license form, which says, "Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed." GNU General Public License, *supra* note 69.

APPENDIX I

Simple Public License (SimPL)

The SimPL applies to the software's source and object code and comes with any rights that I have in it. ***You agree to the SimPL by copying, distributing, or making a derivative work of the software.***

You get the right to:

- Use the software for any purpose;
- Make derivative works of it (this is called a "Derived Work");
- Copy and distribute it and any Derived Work.

If you distribute a Derived Work, you must give back to the community by:

- Prominently documenting any changes that you make to the software;
- Leaving other people's copyright notices in place;
- Providing the source code of any Derived Work in a form that is easy to get and use;
- Letting anyone make, free of charge, derivative works of any Derived Work;
- Licensing any Derived Work under the SimPL.

There are some things that you must shoulder:

- The software comes with NO WARRANTIES of any kind. None;
- If the software damages you in any way, you may only recover direct damages up to the amount you paid for it (that is zero if you did not pay anything). You may not recover any other damages, including those called "consequential damages." (The state or country where you live may not allow you to limit your liability in this way, so this may not apply to you);
- Follow all export control laws.

The SimPL continues perpetually, except it ends automatically if:

- You do not abide by the "give back to the community" terms (your licensees get to keep their rights if they abide);
- A patent holder prevents you from distributing the software under the terms of the SimPL.

APPENDIX II

Simple Public License (SimPL): Annotated

The SimPL applies to the software's[120] *source and object code and comes with any rights*[121] *that I have in it.* ***You agree to the SimPL by copying, distributing, or making a derivative work***[122] ***of the software.***[123]

You get the right to:

- Use the software for any purpose;[124]
- Make derivative works of it (this is called a "Derived Work");[125]

120. "Software" refers to the software that the developer will associate with the SimPL. Ideally, the developer will give potential licensees a meaningful opportunity to review the SimPL and obtain the licensee's assent. GPL "pure license" purists who believe that this practice is unnecessary or abhorrent should at least put the potential licensee on notice that the software is licensed under the SimPL. *See* GNU General Public License, *supra* note 69, app. (demonstrating the use of a notice to apply the license terms to a new program and providing a sample notice).

121. This makes it clear that the license grant is under all potential intellectual property rights, including copyrights and any applicable patents.

122. The GPL uses the terminology "modifying the Program or works based on it" to describe the action that demonstrates assent. *Id.* § 5, at 198. The SimPL uses the terminology "derivative" here because it uses the term "derivative works" in the license grant rather than the GPL's "work based on the Program" nomenclature.

123. The SimPL follows the GPL's model of assent by action. *Id*. However, the SimPL moves this wording up to the beginning of the license to give the potential licensee up-front notice of the actions that will signify assent.

124. The GPL assumes that the user has the right to use the software. *See id.* § 0, at 196. The SimPL says the obvious for the sake of clarity. *See* Gomulkiewicz, *supra* note 13, at 84–86 (explaining that the GPL's ambiguity concerning the right to run a program results in confusion and "[t]he GPL's approach would be more coherent if it simply granted the right to run the program").

125. *See* GNU General Public License, *supra* note 69, § 2, at 196–97. The SimPL uses the copyright term of art "derivative works" rather than the GPL's "work based on the Program" nomenclature. The primary disadvantage of the term "derivative works" is that it is not particularly clear or definite because courts in the United States and around the world have and can in the future construe it in various ways. *See supra* notes 86–95 and accompanying text (explaining the advantages and disadvantages of adopting copyright terms of art). However, the GPL uses the "derivative works" nomenclature at present, so it already suffers from this deficiency; it has the additional deficiency of confusingly using both the "derivative works" nomenclature and the "work based on the Program" nomenclature. *See* Gomulkiewicz, *supra* note 13, at 88–92 (discussing the confusion surrounding the nomenclature). The SimPL does not use the word "contains" as found in GPL section 2(b) because to the extent this word is referring to derivative works it is redundant and to the extent it is referring to collective works the concept is picked up in the right to copy and distribute the software. *See* 17 U.S.C. § 201 (2000) (providing that the owner of a copyright in a collective work has acquired only the right to reproduce and distribute a contributed work as part of the collective work).

- Copy and distribute[126] it[127] and any Derived Work.

If you distribute a Derived Work, you must give back to the community by:

- Prominently documenting any changes that you make[128] to the software;
- Leaving other people's copyright notices in place;[129]
- Providing the source code of any Derived Work in a form that is easy to get and use;[130]
- Letting anyone make, free of charge, derivative works of any Derived Work;[131]
- Licensing any Derived Work under the SimPL.[132]

There are some things that you must shoulder:

- The software comes with NO WARRANTIES of any kind.[133] *None*;
- If the software damages you in any way, you may only recover direct damages up to the amount you paid for it (that is zero if you did not pay anything). You may not recover any other damages, including

---

126. The SimPL uses the word "distribute" because it is one of the exclusive rights granted under copyright law. Internationally, the term is equivalent to the "making available" right. The right to "publish" is subsumed by the right to "distribute" so the SimPL does not use the term "publish."

127. The SimPL assumes that the licensed software includes its source and object code forms. Thus, the SimPL's license to copy and distribute subsumes the license grants to copy and distribute source and object code in GPL sections 1, 2, and 3.

128. *Cf.* GNU General Public License, *supra* note 69, § 2(a), at 196 (stating that modified files must have prominent notices of changes).

129. The GPL mentions that the licensee should retain appropriate copyright notices. *Id.* § 1, at 196. Such notices are not required under the Berne Convention, although they provide some advantages under U.S. copyright law. U.S. COPYRIGHT OFFICE, COPYRIGHT BASICS 4 (2004), *available at* http://www.copyright.gov/circs/circ01.pdf (explaining that one benefit of using a copyright notice is that usually an infringer's assertion of innocent infringement to mitigate damages will be given no weight). Recent amendments to U.S. copyright law make it illegal to strip out copyright management information such as copyright notices. 17 U.S.C. § 1202(b). As such, arguably the SimPL does not need a provision addressing the retention of copyright notices. Nonetheless, copyright notices provide attribution for prior authors, and attribution is often an important goal of open source licensing.

130. The SimPL leaves the exact method up to industry practice and custom. *Cf.* GNU General Public License, *supra* note 69, § 3, at 197 (stating that source code should be distributed on a medium "customarily used for software interchange"). In today's world, posting the source code on the Internet is relatively easy to do.

131. *See id.* § 2(b), at 196.

132. *See id.* §§ 1–2, 6, at 196, 198.

133. *See id.* § 11, at 199.

those called "consequential damages."[134] (*The state or country where you live may not allow you to limit your liability in this way, so this may not apply to you*);

- Follow all export control laws.[135]

The SimPL continues perpetually, except it ends automatically if:

- You do not abide by the "give back to the community" terms (your licensees get to keep their rights if they abide);[136]
- A patent holder prevents you from distributing the software under the terms of the SimPL.[137]

134. *See id.* § 12, at 199–200.

135. *See* Wacha, *supra* note 46, at 467–68 (discussing the application of export control laws to FOSS). U.S. export control regulations allow the unregulated export of "publicly available" software. 15 C.F.R. § 734.3(b)(3) (2005). However, the picture is a bit more complicated if the software contains encryption source code. *See id.* § 740.13(e)(3) (requiring notification prior to publication of certain encryption source code); *id.* § 740.13(e)(2)(ii) (requiring that encryption not be exported to embargoed countries, such as Cuba or North Korea).

136. *See* GNU General Public License, *supra* note 69, § 4, at 198.

137. *See id.* § 7, at 198.