# NOTES

# Copyleft: Licensing Collaborative Works in the Digital Age†

Ira V. Heffan*

*Authors who wish to dedicate their works to the public may think they have no need for copyright or other intellectual property rights. However, if subsequent authors make contributions to an original author's work, those subsequent authors might be entitled to assert proprietary rights in their contributions, thereby defeating the intent of the original author to dedicate his work to the public. The GNU Project is a worldwide collaborative effort to develop high quality software and make it available to the general public. To ensure unrestricted public access, the GNU Project licenses its software under the GNU General Public License ("GPL"), which prevents users from establishing proprietary rights in either the works themselves or subsequent versions thereof. Richard Stallman, the founder of the GNU Project, refers to this type of agreement as "copyleft." In this note, Ira Heffan analyzes the enforceability of the GNU GPL by analogy to shrinkwrap and shareware license agreements. He describes and analyzes the GNU GPL and concludes that it is enforceable. He contends that copyleft is useful for other collaborative works distributed electronically because copyleft assures the works' continued availability to the public.*

* J.D., Stanford Law School, 1997. B.S., Union College, 1988; M.S., Boston University, 1992. I wish to thank: Richard Stallman, Robert Kleiger, Rey Barcelo, Ronan Kennedy, Charles Dennis, Sherri Lynn Wolson, Gregg Homer, and Professor Margaret Jane Radin for their comments on earlier drafts of this note; Professors John Barton, Paul Goldstein, and Carey Heckman for helpful discussion; the *Stanford Law Review* editors for their hard work and excellent editing; and Deborah Kanter Heffan for her patience and proofreading.

An earlier version of this note received first prize in the 59th Annual Nathan Burkan Memorial Copyright Competition at Stanford Law School, sponsored by the American Society of Composers, Authors & Publishers ("ASCAP"), and was entered in the national competition.

## INTRODUCTION

Computer and networking technologies fostered the development of new forms of literary works, such as hypertext-linked World Wide Web pages. At the same time, existing literary works, such as books, magazines, and pamphlets, are transformed in cyberspace because the microprocessor and the Internet allow users to copy, modify, and distribute works stored in electronic media. Authors who wish to share their work with the public can use computer technology to make their works widely available. The technology also allows co-authors to collaborate across great distances and even makes it possible for strangers to create literary and artistic works collaboratively.

Consider the following examples: A group of artists collectively creates works of art by passing each work from one artist to another.[1] People interested in a particular culture collaborate and produce a text about that culture, including historical information and traditional recipes, and they make that work available to anyone who is interested.[2] Amateur genealogists make their family history research available to others.[3] Fans of a musician create an unofficial biography about the musician's life, including a list of all past public appearances and upcoming concerts.[4] Research psychiatrists collaboratively develop test instruments, such as multiple choice surveys for evaluating a pa-

[1] Margaret Chon described a project that was coordinated by Professor Bonnie Mitchell at Syracuse University:

> The Chain Art project was collaborative and took place in a networked digital environment. Technically, each visual image began when a University of Oregon art student uploaded a digitized image onto an FTP site. The person "next in line" in that student's group (typically someone from another state or country) downloaded that image, manipulated it, and uploaded the changed image onto the FTP site. There were twenty-three groups of images, and ultimately 136 participants from ten countries. The final piece is flamboyantly collaborative: each image is attributed to an author and all images are housed together.

Margaret Chon, *New Wine Bursting from Old Bottles: Collaborative Internet Art, Joint Works, and Entrepreneurship*, 75 OR. L. REV. 257, 266-67 (1996) (footnote omitted). This collaborative art project is ongoing. *See* Bonnie Mitchell, *Chain Reaction: An International Collaborative Internet Art Experience* (visited Apr. 16, 1997) <http://chain.syr.edu/mainPages/chainmain.html>.

[2] *See* Craig Cockburn, *soc.culture.scottish FAQ* (visited Apr. 16, 1997) <http://www.scot.demon.co.uk/scotfaq/contents.html>. Most Usenet newsgroups include a frequently asked questions ("FAQ") list that is authored collaboratively. *See* Joel Risberg, *James Taylor's Got a Friend* (visited Apr. 16, 1997) <http://www.hooked.net/~jrisberg/JT/jtstory.html> (explaining that a FAQ is "like the collective knowledge of the group, digested and encapsulated in true Internet style for easy reading").

[3] *See* Marveen F. Knebel, *Montana Genealogy Research* (visited Apr. 8, 1997) <http://www.imt.net/~corkykn/montana.html>.

[4] *See* Risberg, *supra* note 2. Risberg described the collaborative effort needed to produce his FAQ, which is dedicated to the musician James Taylor:

> [F]or the first version of the FAQ, some research was in order. . . . A quick search in the not-yet-obsolete city library turned up plenty of information . . . and it all went into the alt.music.james-taylor FAQ version 1.0.
>
> That first version ran about 10 pages. Version 3.4 went out last month and spanned more than 30. I haven't had to do much research myself since the first version, though. Email comes in from all over with notes on bootleg recordings, concert dates, and sordid gossip about [James Taylor]'s 10-year marriage with Carly Simon. All I really have to do is cut and paste.

*Id.*

tient's mental state, that they want to make widely available to the medical community.[5]

In each of these examples, the groups of authors may live near each other, or they may be scattered throughout the world. The authors may be close friends, or they may be complete strangers linked only by a common interest. In any case, they create works that benefit society by promoting the sharing of ideas, and they have noneconomic reasons to make their works available to each other and the public. They are not trying to make money by selling copies; rather, they are contributing to the marketplace of ideas.

These collaborators use computer networks to provide copies of their works to themselves and others at almost zero cost. A recipient can download digital copies of a work, which are, by definition, identical to the original and can easily be modified. The recipient may update the work or otherwise contribute to it and then make the modified work available to others. In this manner, members of the public with noncommercial interests can collaborate in ways that were previously impossible.

Because these authors do not charge for copies of their works, they might assume that their works do not need copyright or other intellectual property coverage. But if these authors just abandon their copyright, others can establish their own rights in works based on the authors' original work. An author may want to share, but at the same time, want to prevent people from establishing proprietary rights in a derivative work. She might want the public to enjoy and build on her original creation, but she might not want any individual to appropriate her work for personal gain. She may want to ensure that her donation to the public will be used to foster more works that will also be donated to the public. Upon reflection, the author may find that she wants to retain her copyright in order to ensure that future collaborators will respect and support her commitment to unfettered public availability.

Authors and scholars disagree about whether to extend or abandon copyright coverage of digital works.[6] The dispute over the appropriate scope of copyright coverage for computer software during the past twenty years has been part of a broader debate over the appropriate scope of copyright in all

[5] *See* Groningen Dept. of Social Psychiatry, *Schedules for Clinical Assessment in Neuropsychiatry* (visited Apr. 16, 1997) <http://www.psy.med.rug.nl/0018>.

[6] *Compare* Information Infrastructure Task Force, Intellectual Property and the National Information Infrastructure: The Report of the Working Group on Intellectual Property Rights 216 (1995) (advocating extension of copyright to include digital works because "there is no reason to treat works that are distributed in copies to the public by means of transmission differently than works distributed in copies to the public by other, more conventional means"), *with* John Perry Barlow, *The Economy of Ideas: A Framework for Rethinking Patents and Copyrights in the Digital Age (Everything You Know About Intellectual Property Is Wrong)*, Wired, Mar. 1994, at 84, 85 (questioning how we can possibly protect property that can be "infinitely reproduced and instantaneously distributed all over the planet without cost, without our knowledge, without its even leaving our possession"). *See also* M. Ethan Katsh, Law in a Digital World 214-27 (1995) (explaining that intellectual property law must adjust to the complex dynamics of a new information system); Ithiel de Sola Pool, Technologies of Freedom 214 (1983) ("Established notions about copyright become obsolete, rooted as they are in the technology of print."); Richard Stallman, *Reevaluating Copyright: The Public Must Prevail*, 75 Or. L. Rev. 291, 292 (1996) ("[M]aking intellectual property decisions by analogy to physical object property, or to even older intellectual property policies, is a mistake.").

digital works.[7] Meanwhile, a proprietary model has emerged as the primary paradigm for software development, a model which does not support the collaborative development of software programs.

In software parlance, "code" refers to any set of instructions interpreted by a computer. Code appears in various forms. Two such forms are "source code" and "object code." The former is readable by humans and computers, the latter only by computers. It is difficult for a programmer to modify a computer program without the source code.[8]

Under the proprietary software model, most software developers withhold their source code from users. As users work with a given program, they grow accustomed to it. Over time, their needs may change and they may prefer to modify an existing program rather than to replace it with a new and unfamiliar variety. However, they will inevitably find that this preference is both legally and technologically prohibited.

Richard Stallman, founder of the GNU Project[9] and President of the Free Software Foundation, confronted this problem in the context of computer software.[10] Stallman programmed computers before the sale of mass-market software became a multibillion dollar industry. He founded the GNU Project to provide an alternative to the proprietary model of software distribution.[11]

Stallman uses the analogy of buying a house to explain his objection to distributing software without the source code:

> [W]hat would it be like if the only person who could ever fix problems with your house was the contractor who built it originally? That is the kind of imposition that's involved in proprietary software. People tell me about a problem that happens in UNIX. Because manufacturers sell improved versions of

[7] Legal questions regarding copyright for software highlight similar questions about other digital works. For example, if a programmer can, within current copyright law, share his software with the public and ensure that the public will always have use of that work, authors of other kinds of works should be able to do the same.

[8] A computer program is actually composed of a list of code numbers that signify computer instructions. This arrangement of instructions is known as the "object code" or the "executable" program. Because humans cannot easily work with instructions in the format of numbers, computer software is usually written in a high level computer programming language, such as C, Basic, Fortran, Pascal, or Ada, in which commands are written in words, not numbers. High-level languages contain sophisticated shorthand instructions for many numerical instructions. The human-readable form of a program written in a high-level computer programming language is the "source code." Specialized computer programs known as compilers translate specific types of source code into object code. *See* Marci A. Hamilton & Ted Sabety, *Computer Science Concepts in Copyright Cases: The Path to a Coherent Law*, 10 HARV. J.L. & TECH. 239, 266 (1997).

[9] GNU is an abbreviation for "Gnu's Not UNIX." The self-referential name is reminiscent of "recursive" software programming in which a portion of code references itself as a subroutine. "To avoid horrible confusion, please pronounce the 'G' in the word 'GNU' when it is the name of this project." Richard Stallman, *The GNU Manifesto* (visited Apr. 16, 1997) <ftp://prep.ai.mit.edu/pub/gnu/etc/GnuManifesto> [hereinafter *GNU Manifesto*].

[10] *See* Free Software Foundation, *GNU's Who* (visited Apr. 16, 1997) <http://www.gnu.ai.mit.edu/people/people.html>.

[11] *See id.*

> UNIX, they tend to collect fixes and not give them out except in binaries. The result is that the bugs don't really get fixed.[12]

Stallman believes copyright protection turns people into bad neighbors because copyright prevents friends from sharing something that is essentially costless.[13]

The GNU Project distributes software created by Stallman and numerous other programmers under a "copyleft" agreement that permits reproduction and distribution of their works, but does not allow anyone to place further restrictions on them.[14] The GNU General Public License ("GPL") allows users to copy, modify, and distribute both the source and object code, but prevents them from establishing proprietary rights in the software or the changes they may make.[15] The GNU GPL works within the existing copyright doctrine to make the documents publicly available.

Many traditional literary works, such as books and movies, are increasingly found in the digital domain. Just like the GNU Project programmers, many authors of such works, especially works that are developed collaboratively, want to share their creations with the public while preventing third parties from establishing rights in the works that could prevent further sharing. GNU software develops collaboratively as various programmers throughout the world build on the works of others to create more advanced, more functional software. These improvements are licensed under the GNU GPL so that the public may continue to benefit from them. In fact, collaborators are partially motivated to put effort into GNU software by the knowledge that they will have access to everyone else's efforts.

This unique licensing paradigm could easily be applied to collaborative works other than software. Authors located throughout the world, united solely by their authorship of a collaborative work, have the same concerns as GNU Project programmers. They want to allow others to use and enjoy their works, but not to appropriate them.

This note investigates whether "copyleft" can ensure that digital works remain available to the public so that the public always has use of them. The GNU GPL is evaluated by comparison to other mass-market software licenses, specifically shrinkwrap and shareware license agreements.

Part I of this note examines the historical development of computers and computer networks, which are the context for the development of mass-market licensing, and focuses on the development of shrinkwrap license agreements[16]

[12] David Betz & Jon Edwards, *GNU's Not UNIX: Richard Stallman Discusses His Public-Domain UNIX-Compatible Software System with BYTE Editors, July 1986* (visited Apr. 8, 1997) <ftp://prep.ai.mit.edu/pub/gnu/GNUinfo/INTERVIEW>.

[13] *See GNU Manifesto*, *supra* note 9 ("I consider that the golden rule requires that if I like a program I must share it with other people who like it.").

[14] *See* Free Software Foundation, *GNU General Public License Version 2, June 1991* (visited Apr. 16, 1997) <ftp://prep.ai.mit.edu/pub/gnu/COPYING-2.0> [hereinafter *GNU GPL*] (included in the Appendix *infra*).

[15] *See id.*

[16] A shrinkwrap license agreement is a list of terms that may bind a consumer upon purchase and subsequent use of the product, effective once the shrinkwrap plastic around the copy is broken. *See* text accompanying notes 64-70 *infra*.

along with the software market. There are few reported court cases that deal directly with shrinkwrap licenses, but scholars have written a great deal about whether they should be enforceable.[17] Because shrinkwrap doctrine forms the most relevant existing case law and commentary, it is the starting point for this analysis. Part I provides a brief review of the shrinkwrap cases and the analyses used by those courts that have encountered shrinkwrap license issues.

Although shareware has not previously been analyzed by any court or scholarly commentary, this note uses shareware as a transition from shrinkwrap license agreements to copyleft because shareware encourages distribution by the end-user. Courts and scholars have largely ignored shareware, probably because it would be impractical for a small shareware developer to enforce her rights in court and because there is a general perception that shareware is not very important to the software industry.[18] However, shareware is a popular distribution system enabled by digital technology and therefore useful as a steppingstone to the analysis of copyleft.

Part II explores the implementation of copyleft under the GNU GPL and examines the similarities between copyleft, shrinkwrap, and shareware license agreements. Applying the analysis developed in the shareware discussion, Part II concludes that the GNU GPL is an enforceable and effective way for authors to ensure that both their own works and subsequent contributions thereto remain available to the public. In the digital age, authors of collaborative works will find agreements like the GNU GPL useful for facilitating creative, cooperative development.

## I. Software License Agreements

### A. *Historical Development of Computer and Network Technologies*

In the 1950s and 1960s, most organizations that owned computers owned mainframes.[19] IBM dominated the market with its 360 line of mainframe computers.[20] "Typically, one big machine served an entire organization. Often it lay behind a plate glass window, people in white gowns attending it, and those who wished to use it did so through intermediaries."[21]

The 1970s brought technological advances that allowed more transistors to be etched into silicon, thereby permitting the production of more powerful, complex, and affordable computers.[22] Companies like Digital Equipment Corporation and Data General released smaller and less expensive minicomputers,[23] but each brand of minicomputer used different software because the underlying technology was different. Computer hardware manufacturers sim-

[17] *See* text accompanying notes 74-85 *infra*.

[18] *See, e.g.*, John Foley, *The Shareware Alternative*, InformationWeek, Aug. 14, 1995, at 32, 32 ("The shareware market, worth an estimated $300 million, is dwarfed by the multibillion-dollar commercial software business.").

[19] *See* Tracy Kidder, The Soul of a New Machine 11-12 (1981).

[20] *See id.*

[21] *Id.*

[22] *See id.* at 12-14.

[23] *See id.* at 15-16.

ply gave their customers whatever operating systems and software they had free of charge along with the hardware. Corporations large enough to afford the computers paid programmers to develop new applications or modify existing ones.[24] Since so little software was available, scientists and academics commonly shared the software they developed with each other.[25] "[R]esearchers typically swapped programs, embellishing one another's work without much attention to taking credit or nailing down commercial rights."[26]

During the 1980s, the personal computer emerged. Intel, for example, developed a family of microprocessors that could be used in computers that individuals could afford. Based on the Intel processor, IBM produced a "personal computer," which became hugely successful. For the first time, large numbers of computers could run the same software programs, and a mass market in personal computers was born.[27]

As computer hardware became faster and cheaper, networking technology improved. In 1968, the Advanced Research Project Agency ("ARPA"), a division of the United States Department of Defense, first funded the computer network now known as "the Internet."[28] ARPA sought a mechanism for communicating command and control information that could continue to operate even if the network was partly destroyed.[29] Bolt, Beranek and Newman, a government contractor, developed the core packet-switching technology and set up the network after winning the contract from ARPA.[30] From the four sites initially connected in 1968, the network gradually grew. In 1983, the entire network, then connecting approximately 500 academic computer science departments and military laboratories, switched to the TCP/IP networking protocol,[31] the same protocol used on the Internet today. The flexibility of the underlying protocol has allowed the Internet to continue to function despite its incredible growth.[32]

As the network developed, it became easier for computer users across the country and the world to share data and digital works such as software. For example, in 1985, the National Science Foundation ("NSF") developed the NSFNET high speed "backbone" to connect more United States universities.[33] The network was also connected to other networks overseas. By January 1997,

[24] *See* Anthony Lawrence Clapes, Softwars: The Legal Battles for Control of the Global Software Industry 243 (1993).

[25] *See id.*

[26] G. Pascal Zachary, *Free for All: Richard Stallman Is Consumed by the Fight to End Copyrighting of Software*, Wall St. J., May 20, 1991, at R23.

[27] For a discussion of the development of the personal computer, see Bill Gates, The Road Ahead 14-35 (1995).

[28] *See* Andrew S. Tanenbaum, Computer Networks 47 (3d ed. 1996).

[29] *See id.*; Mark A. Lemley, *Shrinkwraps in Cyberspace*, 35 Jurimetrics J. 311, 312 (1995) ("The government hoped that the Internet would be able to survive a nuclear war which damaged or destroyed some nodes of the network.").

[30] *See* Tanenbaum, *supra* note 28, at 47-48.

[31] *See id.* at 52 ("The number of networks, machines, and users connected to the ARPANET grew rapidly after TCP/IP became the only official protocol on Jan. 1, 1983.").

[32] *See id.* at 52-53.

[33] *See* Daniel C. Lynch, *Historical Evolution*, *in* Internet System Handbook 3, 12 (Daniel C. Lynch & Marshall T. Rose eds., 1993).

there were an estimated sixteen million host computers on the Internet.[34] In fact, the number of host computers on the Internet has doubled every year for the past five years.[35]

The rapid growth of computer and networking technologies increased consumer demand for software, and software developers began exploring new ways to define the legal rights of users. The next sections of this note describe the evolution of software licenses from individually negotiated license agreements to shrinkwrap and shareware license agreements.

### B. *Software Licensing Prior to the 1976 Copyright Act*

Professional software developers in the late 1960s and early 1970s wrote specialized software for a particular client or group of clients.[36] The scope of a programmer's copyright and patent rights in software was unclear, so developers tried to cover their software with state trade secret and contract law.[37] Instead of transferring title in the software to their clients, developers retained ownership of the software and "licensed" the software to customers.[38] A term of art in property law, a license is essentially a permit—an intermediate point on the blurry line between property ownership and trespass.[39] Therefore, a customer who has a license to use a copy of a program is a rightful possessor but not an owner of his or her copy.

Early computer software license agreements were based on the developer's property rights in the physical copy of the software.[40] The license agreement gave the customer permission to use the developer's physical property and specified the terms of transfer.[41] If these license agreements transferred intellectual property rights at all, they usually gave the user permission to copy the software only in a manner consistent with the rest of the agreement.[42] License

[34] *See* Network Wizards, *Internet Domain Survey, January 1997* (visited Apr. 21, 1997) <http://www.nw.com/zone/WWW/report.html>. This amazing growth has occurred because of the confluence of computer and networking technology. John Curran, chief technical officer at Bolt, Beranek and Newman, stated: "[T]he Net came at a unique time when the computers and networks that existed were waiting for a way to bring them together." *A Changeling's Tale*, ECONOMIST, July 1, 1995, at 7, 7 (special survey supplement section).

[35] *See* Network Wizards, *supra* note 34.

[36] Most of the software written during this period was created for mainframe computers. *See* Robert W. Gomulkiewicz & Mary L. Williamson, *A Brief Defense of Mass Market Software License Agreements*, 22 RUTGERS COMPUTER & TECH. L.J. 335, 338 (1996) (citing 2 DAVID BENDER, COMPUTER LAW § 4A.141 (1994)).

[37] *See* William H. Neukom & Robert W. Gomulkiewicz, *Licensing Rights to Computer Software*, *in* TECHNOLOGY LICENSING AND LITIGATION 778 (P.L.I. Patents, Copyrights, Trademarks & Literary Property Course Handbook Series No. G4-3897, 1993) (explaining that software publishers license rather than sell software in order to negate the doctrine of first sale, define warranty terms, provide other terms and conditions, and establish rights in the absence of credible intellectual property rights).

[38] *See id.*

[39] *See Foreword* to RESTATEMENT (THIRD) OF PROPERTY: SERVITUDES (Tentative Draft No. 1, 1989) ("[M]any legal disputes in [the field of property law], real world and theoretical, involve the question of whether a transition to property right has been made, or whether the relationship remains a 'mere license' or, worse, an unlawful trespass.").

[40] *See* Mark A. Lemley, *Intellectual Property and Shrinkwrap Licenses*, 68 S. CAL. L. REV. 1239, 1243-46 (1995) (describing the use of licenses to provide trade secret coverage for software).

[41] *See id.*

[42] *See id.* at 1246.

terms were usually negotiated individually with each client, and the licensor could contractually prevent the client from using the software in a certain way or sharing the software with others.[43] Often, license agreements required the licensee to keep the software confidential.[44] Because only object code was provided to the customer and because it cannot be easily read or understood by humans, the software developer could claim that state trade secret law continued to cover the structure of his software even after a copy was handed over to a client.[45]

### C. *Software Licensing Under the 1976 Copyright Act*

Copyright law gives an author the right to control copies of his work. Originating with the advent of the printing press as a way for the Crown to control criticism,[46] copyright developed into a doctrine that allows authors to extract a return on the investment required to create a work. Because content is relatively expensive to create but cheap to copy,[47] intellectual property rights provide an increased incentive to develop creative works by granting authors a monopoly over their work for a limited term.[48] During this term, authors have the exclusive right to copy, adapt, distribute, publicly display, and publicly perform their work.[49] Because digital works are so much easier to copy and modify than analog works, copyright is potentially a powerful right in the digital age.[50]

As the market for computers and software grew, it became impractical for developers to negotiate individual license agreements with each purchaser. The market for software began to look like markets for other works of authorship in that software programs became expensive to create yet relatively inexpensive to copy and distribute.[51] Consequently, commercial software developers sought

[43] *See id.*

[44] *See id.*

[45] *See id.* at 1243-45.

[46] The Statute of Anne provided British authors a monopoly in books for 14 years from the date of publication. 8 Anne, ch. 19 (1710) (Eng.). For a history of early copyright law, see BENJAMIN KAPLAN, AN UNHURRIED VIEW OF COPYRIGHT 1-12 (1967).

[47] *See* KAPLAN, *supra* note 46, at 7-8.

[48] This incentive forms the economic rationale for copyright as provided in the United States Constitution. *See* U.S. CONST. art. I, § 8, cl. 8 (granting Congress the power "[t]o promote the Progress of Science and useful Arts, by securing for limited Times to Authors and Inventors the exclusive Right to their respective Writings and Discoveries").

[49] Section 106 of the 1976 Copyright Act gives copyright owners the following exclusive rights: (1) to reproduce the copyrighted work; (2) to prepare derivative works based upon the copyrighted work; (3) to distribute copies of the copyrighted work by sale, rental, lease, or lending; (4) to perform the copyrighted work publicly; and (5) to display the copyrighted work publicly. *See* 17 U.S.C. § 106 (1994).

[50] Some argue, however, that because digital works are so easy to copy, copyright will be practically unenforceable and therefore worthless. *See, e.g.*, Barlow, *supra* note 6, at 85.

[51] The fixed cost is high due primarily to the creation of code and documentation. The marginal cost of each copy is the price of the media, copying, and transportation. If the work is distributed on the Internet, distribution is virtually costless. Similarly, the cost of copying and the material cost of media are essentially zero. *See* Gomulkiewicz & Williamson, *supra* note 36, at 341-42.

copyright coverage for computer programs.[52] In 1980, Congress responded by amending the Copyright Act to cover software.[53] As an added benefit for developers, under the reciprocity provisions of various international copyright conventions, United States software programs were covered by copyright in other countries without need for additional treaties.[54]

Once copyright law covered software, software developers retained statutory rights in the software even after transferring title in a specific copy to a user.[55] Many of these rights were the same rights previously covered by license agreements. In fact, some licensing lawyers believe that, "[t]oday, sufficient protection is available under the copyright and patent laws of most industrialized countries that contractual treatment may not be absolutely necessary to ensure adequate intellectual property protection for some types of software."[56]

Like authors of works traditionally covered by copyright, programmers could license the copyright in their software to publishers or distributors via agreements similar to those used in the motion picture or print industries.[57] Traditional licensing provided content developers the flexibility to grant exclusive rights to a single distributor or nonexclusive rights to several distributors. Exclusive licenses represented a complete divestment of the rights granted, whereas nonexclusive licenses reserved to the grantor the power and ability to

[52] *See, e.g.*, Note, *Copyright Protection of Computer Program Object Code*, 96 HARV. L. REV. 1723, 1723-24 (1983) (arguing that, with the recent growth of the computer industry, copyright should be extended to cover object code as well as source code).

[53] *See* Pub. L. No. 96-517, § 10, 94 Stat. 3015, 3028 (codified as amended at 17 U.S.C. §§ 101, 117 (1994)) (defining "computer program" in section 101 and authorizing the owner of a copy to make copies necessary to utilize the program or archive it for safekeeping in section 117); Evan Finkel, *Copyright Protection for Computer Software in the Nineties*, 7 SANTA CLARA COMPUTER & HIGH TECH. L.J. 201, 204 (1991) (describing changes in copyright law to cover software). Ten years later, Congress modified the statute to prevent software rental. *See* Computer Software Rental Amendments Act of 1990, Pub. L. No. 101-650, § 802, 104 Stat. 5134 (codified as amended at 17 U.S.C. § 109(b) (1994)).

[54] For example, the Universal Copyright Convention provides that a work created in a member country receives copyright protection in the other member countries. *See* Universal Copyright Convention, Sept. 6, 1952, art. II, 6 U.S.T. 2731, 216 U.N.T.S. 132, revised July 24, 1971, 25 U.S.T. 1341, 1345 ("Published works of nationals of any Contracting State . . . shall enjoy in each other Contracting State the same protection as that other state accords to works of its nationals . . . ."). If a *sui generis* regime for software had been enacted, new international treaties would have to have been negotiated to obtain this reciprocity.

[55] *See* 17 U.S.C. § 202 (1994). Section 202 states in its entirety:

> Ownership of a copyright, or of any of the exclusive rights under a copyright, is distinct from ownership of any material object in which the work is embodied. Transfer of ownership of any material object, including the copy or phonorecord in which the work is first fixed, does not of itself convey any rights in the copyrighted work embodied in the object; nor, in the absence of an agreement, does transfer of ownership of a copyright or of any exclusive rights under a copyright convey property rights in any material object.

*Id.*

[56] Ronald E. Myrick & Penelope Smith Wilson, *Licensing Rights to Software*, *in* TECHNOLOGY LICENSING AND LITIGATION, *supra* note 37, at 503. Developers' software packages may nonetheless include license agreements that attempt to protect the authors' work with contract as well as copyright. *See id.* at 504-05.

[57] *See, e.g.*, Gilliam v. American Broad. Cos., 538 F.2d 14 (2d Cir. 1976) (licensing the right to broadcast a television program); County of Ventura v. Blackburn, 362 F.2d 515 (9th Cir. 1966) (licensing the right to duplicate and sell copies of a map); Considine v. Penguin U.S.A., 24 U.S.P.Q.2d (BNA) 1947 (S.D.N.Y. 1992) (licensing the right to publish a literary work in book form).

grant similar rights to others. The 1976 Copyright Act recognizes this difference between exclusive and nonexclusive licenses.[58] Moreover, the Act makes little if any distinction between assignments and exclusive licenses: "For most purposes, the [1976] Copyright Act treats an exclusive licensee like any other copyright owner."[59]

Nonexclusive licenses are analogous to real property licenses[60] and useful when the owner of a copy wants to use the copy in a manner that would otherwise infringe the copyright owner's exclusive rights. For example, the owner of a copy of a songbook might acquire a nonexclusive license to exercise the songwriter's public performance right in the songwriter's musical compositions.[61] Because minimal rights are transferred, nonexclusive licenses need not be recorded in the Copyright Office and nonexclusive licensees do not have the right to sue infringers.[62] In addition, "absent authority from its licensor, a nonexclusive licensee cannot transfer its license or sublicense under it."[63]

[58] *See* 1 PAUL GOLDSTEIN, COPYRIGHT § 4.4.1.1, at 4:50 (2d ed. 1996). The legal implications of exclusive and nonexclusive copyright licenses changed with amendments to the Copyright Act. The 1909 Copyright Act made a broad distinction between assignments and licenses. *See id.* § 4.4.1.2.

[59] *Id.* § 4.4.1.1, at 4:50. Exclusive licenses must be recorded and exclusive licensees are given the right to sue copyright infringers. *See id.* § 4.5.1.1(a).

[60] *See* Schuyler M. Moore, *Entertainment Bankruptcies: The Copyright Act Meets the Bankruptcy Code*, 48 BUS. LAW. 567, 569 (1993) ("[T]he licensee under a non-exclusive license does not own any property interest in the licensed rights.").

[61] If the owner does not provide the license explicitly, a court might find the nonexclusive license implied in the sale of the work. *See* Johnson v. Jones, 885 F. Supp. 1008, 1014 (E.D. Mich. 1995) ("[A]ll of the circumstances surrounding the negotiations made between the parties must be considered to determine if and to what extent an implied license was granted.").

When Congress brought software under the regime of copyright, it was faced with an interesting inconsistency between digital works and copyright. The right to reproduce copies of the work under section 106(1) of the 1976 Copyright Act is required for the possessor of a copy to use software covered by copyright. When a computer displays a digital work, the screen is necessarily a reproduction of the code that resides in the computer's memory, and the computer's memory necessarily contains a reproduction of the information stored on the computer's disk. *See* MAI Sys. Corp. v. Peak Computer, Inc., 991 F.2d 511, 518 (9th Cir. 1993) ("The law also supports the conclusion that Peak's loading of copyrighted software into RAM [memory] creates a 'copy' of that software in violation of the Copyright Act."). Once reproduction of software copies was restricted by copyright law, a specific exception to the author's exclusive rights in reproduction was necessary to allow computer program owners to make the copies necessary to use programs and the archive copies for backup purposes. *See* 17 U.S.C. § 117 (1994) (providing an exemption for the owners of a copy of the copyrighted work). The section 117 exceptions are very narrow in that they apply only to owners of computer programs. If a user is not the *owner* of a program, or if the work cannot be classified as a *computer program*, defined in section 101 as "a set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result," then section 117 does not apply. 17 U.S.C. §§ 101, 117. This also applies to other digital works that are covered by copyright but do not have a specific statutory exemption for use on a computer. For example, the owner of a copy of a digital work that is not a computer program, such as a song or an image, may not have a statutory exception to the license requirement and thus permission to make copies necessary to use the work.

[62] *See* 1 GOLDSTEIN, *supra* note 58, § 4.4.1.1, at 4:50. However, some courts have voided nonexclusive license agreements because they did not meet the statute of frauds requirement of a writing. *See, e.g.*, Freedman v. Select Info. Sys., 221 U.S.P.Q. (BNA) 848 (N.D. Cal. 1983); Meyers v. Waverly Fabrics, 479 N.E.2d 236, 238 (N.Y. 1985).

[63] 1 GOLDSTEIN, *supra* note 58, § 4.4.1.1, at 4:50.

### D. *Shrinkwrap License Agreements*

As the commercial software market expanded and the negotiation of individual software licenses became less practical, developers began using "shrinkwrap license agreements" in an attempt to delineate the use rights afforded their retail consumers. Shrinkwrap license agreements earned their name because their terms are usually visible through the shrinkwrap plastic surrounding a box of computer disks. Software developers assert that a contract is formed based on the terms of the shrinkwrap license when the purchaser breaks the shrinkwrap plastic and removes the software.[64] Although the developer does not actually obtain the user's signature to the contract, the developer claims that breaking the shrinkwrap seal on the software constitutes acceptance of the developer's terms.[65] To avoid the formation of a contract, the user is instructed to return the software for a refund, with the shrinkwrap unbroken.[66] These shrinkwrap licenses may contain provisions that extend the software developer's proprietary rights in the software,[67] define the scope of the warranty provided with the software,[68] limit the purchaser's rights to use the software,[69] and provide contract rights that may supplement or replace other intellectual property rights.[70]

Some software developers attempt to use shrinkwrap license agreements to establish the legal fiction of retaining ownership in the physical copy of the software by stating in the shrinkwrap license that the software transfers as a rental rather than as a sale. The developer then retains all property rights, both intellectual and tangible, and only grants a nonexclusive license to the user. If the software developer retains ownership in the user's copy of the software, then 17 U.S.C. § 117, which gives the *owner* of a software program the right to make copies necessary to run the program, does not apply to the user.[71] If the

[64] *See* Neukom & Gomulkiewicz, *supra* note 37, at 777 ("The end user indicates 'consent' to the terms of the license by opening the software disk package or the sealed product container rather than by signing a written document."); *see also* Lemley, *supra* note 40, at 1241 ("[T]he software vendor is attempting to create what some have referred to as a 'reverse unilateral contract.'").

[65] *See* Neukom & Gomulkiewicz, *supra* note 37, at 777.

[66] *See id.* ("The end user may reject the terms of the license by returning the software product to the publisher for a full refund.").

[67] Contract terms might include a provision requiring that the contents of the program be kept secret so that the software would qualify as a trade secret. *See* Lemley, *supra* note 40, at 1242 ("Generally, proprietary rights provisions asserted that the information contained in the accompanying computer software was proprietary to the vendor and could not be copied or disclosed without the vendor's permission.").

[68] Software license agreements typically try to limit any and all warranties. *See id.* at 1245. Although warranties are not the focus of this paper, they constitute an important part of the software license agreement.

[69] For example, a developer may use license agreements to preclude reverse engineering or define the specific use of the software. *See* Neukom & Gomulkiewicz, *supra* note 37, at 778; Lemley, *supra* note 40, at 1247.

[70] In the event the vendor's asserted intellectual property rights are invalid, contractual rights may still be enforceable.

[71] Only the owner has that right under section 117:

> [I]t is not an infringement for the owner of a copy of a computer program to make or authorize the making of another copy or adaptation of that computer program provided: (1) that such a new copy or adaptation is created as an essential step in the utilization of the computer program . . . .

software is sold to the user, then subsequent transfer to a third party is specifically permitted under the "first sale" doctrine of copyright law, but if the software is only licensed, then the software developer may prevent the user from transferring ownership in the copy to a third party.[72] By licensing the software instead of selling it, the developer moves the transaction from the statutory realm of copyright law to the "free-market" domain of contract, thus limiting the user's rights under the Copyright Act.[73]

The validity of shrinkwrap license agreements has been explored in only a few appellate court cases.[74] Courts have generally refused to enforce shrinkwrap licenses where the user has not manifested the requisite intent to accept its terms or the agreement is preempted by federal law. A court might also refuse to enforce a shrinkwrap license if its terms are unconscionable, it constitutes a contract of adhesion, or it otherwise violates public policy.[75]

The Third Circuit held in *Step-Saver Data Systems v. Wyse Technology Inc.*[76] that a shrinkwrap contract was formed, if at all, when money and software changed hands.[77] According to the court, the terms of the contract include only those terms agreed on at the time of sale; omitted terms are determined by the Uniform Commercial Code ("UCC"). Since the shrinkwrap license at issue was not visible to the purchaser until after the contract was formed, the shrinkwrap license was deemed a subsequent attempt to change the contract terms. Such an attempt to modify the terms of the contract was held not to be binding because both parties did not agree to the new terms.[78]

A contract might be preempted if it includes rights that are covered by federal copyright law. The federal copyright statute specifically preempts state

17 U.S.C. § 117 (1994).

[72] *See* Neukom & Gomulkiewicz, *supra* note 37, at 778. Neukom & Gomulkiewicz explain:

> [O]nce a copy of a copyrighted work has been sold, the copyright holder's rights in that particular copy are exhausted, and the copy may be freely resold, leased or loaned . . . [but i]f title to the particular copy of the program is not transferred to the user, the user may only transfer the software to others as described by the terms of the license agreement.

*Id.*; *see, e.g.*, Microsoft Corp. v. Harmony Computers & Elecs., Inc., 846 F. Supp. 208, 212-13 (E.D.N.Y. 1994) (holding that licensee was not entitled to sell a copy of purchased software absent a transfer of title).

[73] *See* Neukom & Gomulkiewicz, *supra* note 37, at 777 (giving reasons publishers license rather than sell copies).

[74] *See* ProCD, Inc. v. Zeidenberg, 86 F.3d 1447 (7th Cir. 1996) (holding that the shrinkwrap contract was part of the agreement and that the contract was not preempted); Step-Saver Data Sys., Inc. v. Wyse Tech., 939 F.2d 91 (3d Cir. 1991) (holding that shrinkwrap contract was not part of the agreement); *see also* National Car Rental Sys. v. Computer Assocs. Int'l, 991 F.2d 426 (8th Cir. 1993) (upholding negotiated license agreement and holding that the terms of the license agreement did not assert any rights equivalent to federal copyright); Arizona Retail Sys. v. Software Link, Inc., 831 F. Supp. 759 (D. Ariz. 1993) (distinguishing an initial shrinkwrap transaction, in which the user had an opportunity to try out the software, from later transactions made with no discussion of the shrinkwrap terms before purchase); David L. Hayes, *Shrinkwrap License Agreements: New Light on a Vexing Problem*, 15 HASTINGS COMM. & ENT. L.J. 653 (1993) (discussing the validity of shrinkwrap license agreements); Lemley, *supra* note 40, at 1244 n.23 (discussing the legal status of shrinkwrap licenses).

[75] *See Step-Saver*, 939 F.2d at 99-100.

[76] 939 F.2d 91 (3d Cir. 1991).

[77] *See id.* at 100.

[78] *See id.* at 98-99.

law rights that are equivalent to copyright.[79] Whether a breach of contract claim is equivalent to copyright is largely an open question, but it has been addressed in the Seventh Circuit.[80] In *ProCD v. Zeidenberg*,[81] Judge Easterbrook held that a shrinkwrap license was not preempted because "a simple two-party contract is not 'equivalent to any of the exclusive rights within the general scope of copyright' and therefore may be enforced."[82] The contract was an "extra element" that made the contract claim different from a copyright claim.

In summary, a shrinkwrap license agreement can serve two purposes. First, as a copyright license, it effects the transfer of rights from the copyright owner to the user. A shrinkwrap license will usually be enforceable in accordance with its terms, except to the extent limited by statute, common law, or the conduct of the parties. Second, a shrinkwrap license can serve as a contract. Whether the contract will be enforced depends on contract principles and the applicable provisions of the UCC, as well as the doctrine of preemption.[83] Courts tend to enforce shrinkwrap licenses when they meet traditional contract formalities and do not violate federal intellectual property policy.[84]

The shrinkwrap license agreement cases have important implications for the electronic distribution of software and other content over the Internet, since the

[79] The Supreme Court has said, "When state law touches upon the area of [patent or copyright statutes], it is 'familiar doctrine' that the federal policy 'may not be set at naught, or its benefits denied' by the state law." Sears, Roebuck & Co. v. Stiffel Co., 376 U.S. 225, 229 (1964) (citations omitted). A state law claim is specifically preempted by the federal copyright statute if (1) the subject matter of the state law claim falls within the subject matter of the copyright laws as set forth in 17 U.S.C. §§ 102–103, and (2) the state law creates rights which are equivalent to any of the exclusive rights granted to the copyright holder by section 106. *See* 17 U.S.C. § 301 (1994); Vault Corp. v. Quaid Software Ltd., 847 F.2d 255, 270 (5th Cir. 1988) (holding a portion of the Louisiana License Act preempted by federal copyright law); Del Madera Properties v. Rhodes & Gardner, Inc., 820 F.2d 973, 976-77 (9th Cir. 1987) (holding a claim under state law preempted by federal copyright law); Ehat v. Tanner, 780 F.2d 876, 878 (10th Cir. 1985) (holding state law preempted by federal copyright law); Harper & Row Publishers, Inc. v. Nation Enter., 723 F.2d 195, 199-200 (2d Cir. 1983) (holding state law preempted by federal copyright law), *rev'd on other grounds*, 471 U.S. 539 (1985).

[80] *See* ProCD, 86 F.3d at 1453-54 (holding license agreement not preempted). Professor Goldstein writes, "Contract law is a good example of a state law that will be immune from [section 301(a)] preemption under the extra element test. . . . [C]ontract law requires the plaintiff to prove the existence of a bargained-for exchange—something it need not prove in a cause of action for copyright infringement." 3 GOLDSTEIN, *supra* note 58, § 15.2.1.2, at 15:11; *see also* National Car Rental Sys. v. Computer Assocs. Int'l, 991 F.2d 426, 430-31 (8th Cir. 1993) (holding license agreement did not assert any rights equivalent to federal copyright); Taquino v. Teledine Monarch Rubber, 893 F.2d 1488, 1501 (5th Cir. 1990) (holding contract not preempted). However, some courts have found contract claims to be preempted despite the extra element of a contract. *See* American Movie Classics Co. v. Turner Entertainment Co., 922 F. Supp. 926, 931-32 (S.D.N.Y. 1996) (finding no "qualitative difference" between breach of contract claim and copyright claim); Wolff v. Institute of Elec. & Elecs. Eng'rs, Inc., 768 F. Supp. 66, 69 (S.D.N.Y. 1991) (holding contract claim preempted by Copyright Act); Acorn Structures, Inc. v. Swantz, 657 F. Supp. 70, 75 (W.D. Va. 1987) (holding breach of contract claim covering an idea preempted by Copyright Act).

[81] 86 F.3d 1447 (7th Cir. 1996).

[82] *ProCD*, 86 F.3d at 1454.

[83] Copyright preemption doctrine is codified at 17 U.S.C. § 301. *See* note 79 *supra*.

[84] Plans to rewrite the UCC include the addition of section 2-2203, which would make shrinkwrap licenses enforceable under state statute. *See* Lemley, *supra* note 40, at 1259-63; *see also* John B. Kennedy & Shoshana R. Davids, *Web-Site Agreements Do Not Wrap Up IP Rights*, NAT'L L.J., Oct. 23, 1995, at C3.

same licensing principles that apply in a retail store or over the telephone also apply online.[85]

## E. *Shareware License Agreements*

Shareware is software that programmers distribute openly and widely for computer users to try before they buy.[86] A shareware software developer can distribute his software by uploading it onto electronic bulletin boards and Internet archive sites.[87] The software may then be downloaded by prospective users.[88] Typically, the shareware developer grants the user permission to use the software free of charge during a trial period.[89] If the user decides to keep the software, she pays the developer and "registers" the software.[90] If she decides not to pay for the software, she is expected to delete it from her computer.[91] As an incentive for payment, the developer may provide registered users with a version of the software that includes additional features or a printed copy of the manual.[92] In essence, shareware is an arrangement based on trust that enables programmers to distribute software directly to users without the distribution overhead associated with customary retail sales channels.[93]

[85] *See* Kennedy & Davids, *supra* note 84, at C3 (describing the use of "web-wrap" agreements to cover World Wide Web pages).

[86] Robert Wallace, one of Microsoft's original software developers, is generally credited with coining the term "shareware." *See* Foley, *supra* note 18, at 32; Jerry Pournelle, *Computing at Chaos Manor: Traveling Computers*, BYTE, July 1986, at 325, 342 (describing Wallace's PC-Write program as "the original 'shareware'"). Jim Knopf (a.k.a. Jim Button) and Andrew Fluegelman wrote the first widely distributed personal computer programs that requested voluntary payment. *See* Craig T. Turkington, *A Comparison of Shareware and Other Software Licensing: Role of the New Copyright Office Shareware Registry*, 1 U. BALT. INTELL. PROP. L.J. 76, 78 n.12 (1992). Knopf and Fluegelman originally used the term "freeware" to describe this distribution system. *See* Jim Knopf, *The Shareware Story* (visited Apr. 24, 1997) <http://www.halcyon.com/knopf/history.htm>. "Shareware" is now the most commonly used word to describe this distribution method. *See id.*

[87] It was possible to distribute shareware before the development of computer networks, but the methods available at the time involved the exchange of physical media at greater cost.

[88] Shareware may also be obtained from independent distributors who sell disks and CD-ROMs containing compilations of shareware software. Some computer equipment manufacturers also "bundle" shareware with their products.

[89] *See* Association of Shareware Professionals, *Shareware FAQ* (visited Apr. 24, 1997) <http://www.asp-shareware.org/sharewar.html> [hereinafter *Shareware FAQ*] ("You pay for [the shareware program] at the end of a trial period (typically 30 days) by sending the author a fee he or she has established for the program.").

[90] *See id.*

[91] *See id.*

[92] *See* Pamela Samuelson, Randall Davis, Mitchell D. Kapor, & J.H. Reichman, *A Manifesto Concerning the Legal Protection of Computer Programs*, 94 COLUM. L. REV. 2308, 2377 (1994). They state that shareware

> [r]egistration may be attractive because it offers benefits such as periodic code updates, a printed manual, or at times, a more fully functional version of the program, but the basic spirit is of sharing interesting, enjoyable, or useful code, primarily for the sake of enjoying good craftsmanship rather than profit.

*Id.*; *see also*, *Shareware FAQ*, *supra* note 89 (explaining that, when users register shareware, they typically receive the same things they receive when they pay for other software).

[93] Possible grounds for enforcing shareware distribution arrangements are discussed in this note only to examine the legal basis for enforcement and not because such enforcement in court is anticipated. Users are on the honor system to pay, and it is currently impractical to enforce a shareware agreement against all users. Research revealed no cases attempting to enforce a shareware agreement in court. People do not register shareware because of the legal requirement, but rather because of good-

From a legal standpoint, the shareware transaction can be divided into two stages. In the first stage, the user downloads the software onto her computer and uses it during the trial period.[94] Here the question is whether the developer can enforce payment for or deletion of the software after the trial period ends.[95] The transition from the first stage to the second stage occurs when the user registers the software. In the second stage, the analysis of the license agreement's terms is similar to the analysis of other commercial software licenses. In the following two subsections, the shareware developer's copyright and contract rights are examined.

### 1. *The user downloads the software.*

When a user obtains shareware from a bulletin board or over the Internet, the program is usually accompanied by a license limiting the developer's right to copy and distribute the work. Since a license is simply a form of permission, the developer may place conditions or restrictions on that permission. If the licensee uses the software and later copies the work in violation of the conditions, then the grantor may have an action for copyright infringement, breach of contract, or both.[96]

One copyright-based argument for the enforceability of shareware registration requirements assumes that the copy is only authorized as long as it exists within the terms of the license. Any copying outside the limited terms of the license constitutes copyright infringement. A court could find that a user who downloads shareware for permanent use, not for temporary evaluation, copies

will, product support, and access to the updated or more functional versions that come with registration. As John Perry Barlow writes:

> [P]eople seem to eventually buy the software they really use. Once a program becomes central to your work, you want the latest version of it, the best support, the actual manuals, all privileges attached to ownership. Such practical considerations will, in the absence of working law, become more and more important in getting paid for what might easily be obtained for nothing.

Barlow, *supra* note 6, at 128.

[94] *See Shareware FAQ, supra* note 89.

[95] Actually, the question is of more than academic interest. At least one successful shareware vendor deliberately allowed individuals to use its software, but attempted to extract payment from corporations when the individuals brought the software into the office. *See* Laurie Delater Weeks, *McAfee, Maker of Antiviral Software, Plans Offering to Build on Its Success*, WALL ST. J., Sept. 21, 1992, at A4. McAfee allowed people to share the software in the hope that

> [a]t least some of the people . . . will find it useful enough to share it with office co-workers. Big employers, which routinely check for usage of unlicensed software, will discover McAfee's product and voluntarily pay his comparatively low license fees, or increasingly, do so when contacted by independent agents for McAfee Associates.

*Id.*

[96] The Second Circuit has found that "use of a work that exceeded the license granted by the proprietor of the copyright" would constitute copyright infringement. Gilliam v. American Broad. Cos., 538 F.2d 14, 21 (2d Cir. 1976). Professor Nimmer agrees:

> If the nature of a licensee's violation consists of a failure to satisfy a condition to the license (as distinguished from a breach of a covenant), it follows that the rights dependent upon satisfaction of such condition have not been effectively licensed, and therefore, any use by the licensee is without authority from the licensor and may therefore, constitute an infringement of copyright.

3 MELVILLE B. NIMMER & DAVID NIMMER, NIMMER ON COPYRIGHT § 10.15, at 10-121 (1995) (footnotes omitted).

outside the bounds of the shareware license, since the user has infringed the programmer's copyright by making an unauthorized copy. Additionally, a court could find that the shareware user is not the owner of the copy and therefore does not have the right to use the software without the copyright owner's explicit permission.[97]

A contract-based argument also exists for allowing shareware developers to enforce the shareware distribution system and prevent users from downloading shareware without paying. A contract exists if there is an offer, acceptance, and consideration. If a user receives the software knowing that the offer was limited to inspection and evaluation and the user then accepts the offer by downloading the software, there is a meeting of the minds on the proposed use restrictions.[98] Shareware is so widespread among the computer community that, when a user downloads shareware from a web site or bulletin board, one could argue that the user already knows, or should know, the terms of the license, with or without actual notice.

If a notice appears when the shareware program first runs, indicating that the software is provided only for a limited evaluation term, that notice along with continued use by the user could evidence the formation of a contract on the terms appearing in the notice. Even if the user did not know about the terms before she downloaded the software, by continuing to use the software, the user agrees to the license terms and must either pay for the software or delete it at the end of the trial period. Once again, there is a meeting of the minds, and the exchange of software for a promise either to pay for it or to delete it represents consideration. In a typical shareware scenario, where a user downloads a trial version of a program from a bulletin board or an Internet web site, no money changes hands before the user is given a chance to evaluate the software and the license agreement. Therefore, the user cannot use the Third Circuit's argument in *Step-Saver* that the terms of the agreement constitute an unaccepted modification of a contract previously formed at the time of purchase.[99]

### 2. *The user purchases the software.*

When the user registers shareware with the developer, the user expressly agrees to the terms of the license agreement that came with the software. The user simultaneously downloads or otherwise obtains the software and the terms

[97] *See* note 61 *supra* (discussing 17 U.S.C. § 117). Section 117 allows the *owner* of a copy of a computer program to make the copies necessary to run the program. *See* 17 U.S.C. § 117 (1994). Since this section does not apply to nonowners, nonowner possessors of a copy apparently need the copyright owner's explicit permission to run the program.

[98] Both the Wisconsin District Court and the Seventh Circuit panel that reversed it would probably agree with this assessment. *See* ProCD, Inc. v. Zeidenberg, 908 F. Supp. 640, 655 (W.D. Wis. 1996) ("I conclude that because defendants did not have the opportunity to bargain or object to the proposed user agreement or even review it before purchase and they did not assent to the terms explicitly after they learned of them, they are not bound by the user agreement."), *rev'd*, 86 F.3d 1447, 1452 (7th Cir. 1996) ("A vendor, as master of the offer, may invite acceptance by conduct, and may propose limitations on the kind of conduct that constitutes acceptance. A buyer may accept by performing the acts the vendor proposes to treat as acceptance.").

[99] *See* text accompanying notes 78-79 *supra*.

of the license agreement. If the license terms are spelled out in the program files and on the screen when the program runs, then the user is aware of the existence and terms of the agreement as soon as she reads the files or executes the program.

When the user sends a check or other form of payment to the shareware developer, she purchases the software and assents to the terms of the license agreement. This second stage of the shareware software purchase more closely resembles a bargained-for agreement than the commercial shrinkwrap licenses discussed earlier.[100] Unlike the shrinkwrap scenario, the shareware user is given a trial period to evaluate the software and read the terms of the license agreement. If the user does not want the shareware once the trial period expires, she does not have to return anything to the point of purchase. She is merely required to delete the software from her computer.

## II. Copyleft

While he was still an undergraduate physics major at Harvard in 1971, Richard Stallman joined the Artificial Intelligence laboratory at the Massachusetts Institute of Technology ("MIT"). At that time, the software industry was in its infancy. While working at MIT, Stallman became personally affected by the expansion of proprietary rights in software and saw companies increasingly unwilling to provide source code with their products.[101]

[100] A shareware transaction resembles the initial transaction in *Arizona Retail Systems v. Software Link, Inc.*, 831 F. Supp. 759 (D. Ariz. 1993), in which the software developer sent an evaluation copy of the software in a box that had the license terms pasted on its front. Unregistered shareware is an evaluation copy of the software. Also, the Copyright Office has a special registry for shareware in which the terms of the license agreement are made available to the public. *See* Judicial Improvements Act of 1990, Pub. L. No. 101-650, § 805, 104 Stat. 5089, 5136 (1990) (codified as note following 17 U.S.C. § 205); *see also Legislation: President Signs Bill on Software Rental, Architectural Works and Artist's Rights*, 41 Pat. Trademark & Copyrights J. (BNA) 123 (1990).

[101] Stallman told Simson L. Garfinkel, writing for MIT's *Technology Review*, a story about the first two laser printers in the MIT Artificial Intelligence lab:

> The laser printers of the mid-1970's were the size of today's compact cars. When Xerox gave the AI lab a Xerox Graphics Printer, the only place for it was in the lab's ninth-floor machine room. Researchers connected the printer to the local area network that the lab was developing, and soon anybody in the building could print a 100-page document by typing in a few commands.
>
> That worked fine, except that sometimes the printer would run out of paper or jam, and dozens of other jobs would pile up. Other times there would simply be a lot of people wanting to print long documents, and the person who needed to print a single page would have to run up and down the stairs or babysit the printer until that page appeared. But since the programmers at the lab had the source code to the program that ran the printer, they could add features that solved these problems. Soon the printer was helping the lab run smoothly. "It would send you a message when your document had been printed," recalls Stallman. "It would send you a message if you had anything queued and there was a paper jam."
>
> All this changed in 1978, when Xerox replaced the machine with a new laser printer called a "Dover" but wouldn't share the printer's source code with the lab. "We wanted to put those features into the Dover program, but we couldn't," Stallman says. Xerox wouldn't put the features into the program either. "So we had to suffer with paper jams that nobody knew about."

Simson L. Garfinkel, *Programs to the People*, Tech. Rev., Feb./Mar. 1991, at 53, 54.

Stallman opposes the application of intellectual property and contractual rules to software on ethical grounds.[102] He believes that the commercialization of software creates conflict among programmers: "The fundamental act of friendship among programmers is the sharing of programs; marketing arrangements now typically used essentially forbid programmers to treat others as friends."[103] In 1984, Stallman resigned from MIT so that the University would have no legal claim on his software, but MIT allowed him to keep his office and use the Institute's laboratory equipment because the software he developed was useful. Stallman was quoted in the electronics engineering magazine *EDN*:

> I decided that it wasn't worth continuing in the software field without being able to cooperate with people and to write and improve whatever program you want to improve . . . . I decided that I would make a new software-sharing community even if I had to write all the software myself.[104]

Stallman wants programmers to be able to share programs with each other. Under the most common commercial model of software distribution, where the object code is provided but the source code is not, programmers cannot improve the software they buy. Stallman, however, always provides the source code along with the object code, thereby allowing users to make changes.

The Free Software Foundation[105] is a nonprofit organization founded by Stallman in 1985 to support free software development by the GNU Project.[106] As described in the *GNU Manifesto*,[107] Stallman believes programmers should get paid for the work they do, not for the exploitation of proprietary rights. This philosophy radically departs from existing proprietary rights models, such as copyright and patent, in which software developers generate income by selling copies of their software. Under existing proprietary rights models, users are prohibited from copying or modifying the software without the copyright owner's consent.

Stallman believes that preventing people from sharing information with friends constitutes antisocial behavior.[108] "[C]opying useful, enlightening or entertaining information for a friend makes the world happier and better off; it benefits the friend and inherently hurts no one. It is a constructive activity that

[102] *See* John Foley, *The High Priest of Software*, INFORMATIONWEEK, Aug. 14, 1995, at 36, 36 (quoting Richard Stallman: "I would like to see proprietary software disappear because people think the idea is so disgusting, they wouldn't have anything to do with it.").

[103] *GNU Manifesto*, *supra* note 9.

[104] Jay Fraser, *Keeper of the Faith: Richard Stallman Is Leading a Crusade to Preserve Your Programming Freedom*, EDN, Oct. 1, 1990, at 174, 176 (quoting Richard Stallman). Stallman supported himself by doing a small amount of consulting each year. In 1990, he received a John D. and Catherine T. MacArthur Foundation "genius fellowship," an award given to gifted individuals to allow them to devote themselves to their pursuits. *Id.* at 178.

[105] In the context of the Free Software Foundation, "free" refers to the liberty to copy and modify software, not to price. "To understand the concept, you should think of 'free speech,' not 'free beer.'" Free Software Foundation, *What Is Free Software?* (visited Apr. 8, 1997) <http://www.gnu.ai.mit.edu/philosophy/free-sw.html>.

[106] *See* Declaration of Richard M. Stallman, Bernstein v. United States Dept. of State, 945 F. Supp. 1279 (N.D. Cal. 1996) (No. C-95-0582 MHP) <http://www.eff.org/pub/Privacy/ITAR_export/Bernstein_case/Legal/960726_filing/stallman.decl>.

[107] *See GNU Manifesto*, *supra* note 9.

[108] *See id.*

strengthens social bonds."[109] He thinks programmers should get paid for providing convenient access to software, as well as customizing, servicing, repairing, and supporting software, but should not be compensated for the proprietary rights in their code. Programmers may be paid less under this structure, but it will create a more collaborative atmosphere.[110] As *Computer Weekly* reported:

> The Free Software Foundation does not provide technical support directly, although it publishes a directory of consultants. "We see programmers as providing a service, much as doctors and lawyers do now; both medical and legal knowledge are freely re-distributable entities for which the practitioners charge a distribution and service fee," says a spokesman.[111]

The goal of the GNU Project is to create a permission-free, compatible replacement for the UNIX operating system, as well as other application programs and utilities.[112] Several hundred programs are currently available for public use in the GNU library.[113] The software is extremely popular and widely acknowledged to be of very high quality.[114] Many programmers develop the software collaboratively, improving features and creating new tools.[115] "Free," in the context of the Free Software Foundation's name, is defined as permission free, not free of charge.[116] Although the GNU software is available free of charge over the Internet, the Free Software Foundation does

[109] Stallman, *supra* note 6, at 294.

[110] As a good example of such a collaborative atmosphere, a group of computer companies pooled funds to support the maintenance of the GNU C compiler ("gcc"). *See GNU Manifesto*, *supra* note 9, at n.5. Companies often find it in their own interest to support a freely available compiler that each company can modify for use with its products.

[111] Rachel Frampton, *Would You Put It in Your Machine?*, COMPUTER WKLY., Sept. 23, 1993, at 24, 24. This concept is similar to John Perry Barlow's idea that musicians should get paid for their live performances and the albums they sell, but should allow free recording of their concerts:

> True, I don't get any royalties on the millions of copies of my songs which have been extracted from concerts, but I see no reason to complain. The fact is, no one but the Grateful Dead can perform a Grateful Dead song, so if you want the experience and not its thin projection, you have to buy a ticket from us. In other words, our intellectual property protection derives from our being the only real-time source of it.

Barlow, *supra* note 6, at 126. In Barlow's model, the real work that musicians do is the live performance. When fans make copies of their work, the artists become more popular. In the case of the Grateful Dead, this popularity allowed them to sell out large amphitheaters. *See id.*

[112] For an explanation of the name GNU, see note 9 *supra*. "We hope to supply, eventually, everything useful that normally comes with a Unix system, and more." *GNU Manifesto*, *supra* note 9.

[113] The list of available programs includes compilers or interpreters for at least seven programming languages (Ada, C/C++, Fortran, Lisp, Logo, Postscript, and Smalltalk), as well as games (Aerial Combat, Go, NetHack, and Shogi) and application programs (emacs editor, oleo spreadsheet, and gdb software debugger). *See* Free Software Foundation, *GNU Program/Package Index* (visited Apr. 8, 1997) <http://www.gnu.ai.mit.edu/prep/program.index.html>.

[114] *See* Jack Woehr, *Getting to Know GNU*, EMBEDDED SYS. PROGRAMMING, Feb. 1994, at 34, 37 (explaining that programmers writing software for recently developed hardware may not need to obtain the GNU software directly from the Free Software Foundation's archive because "[t]he popularity among the programming elite of the GNU toolchain makes the inclusion of these tools incumbent upon any hardware vendor intending to provide free development tools accompanying an initial hardware release").

[115] *See* Jack Woehr, *What's GNU*, EMBEDDED SYS. PROGRAMMING, Jan. 1994, at 70, 72 ("[A]s the usage of GNU tools spread, so did the participation of the users themselves. They voluntarily expended thousands of person-hours in improving extant tools and coding new ones for free distribution.").

[116] *See* note 105 *supra*.

charge for GNU CD-ROMs. The distribution model also allows distributors to charge for modified versions of a program. But according to the terms of the license agreement, these modified versions must be distributed with the source code and unencumbered by any other restrictions.[117]

### A. *The GNU General Public License*

Richard Stallman wants to share GNU software with the public. He wants to allow everybody to use the software, modify it, and give it to their friends. He wants to make sure that the public will always have the benefit of the GNU software and that anyone who develops a work based on it also gives the public permission to use, modify, and redistribute the new version.

At first glance, dedication to the public seems like it would be easy to accomplish by simply abandoning copyright and donating the work to the public domain. However, when a work is in the public domain, others can establish their own proprietary rights in new versions to which they contribute. For example, when an old song falls into the public domain after its copyright has expired, a performer can own a copyright in a new arrangement based on the old song.[118] Similarly, when a computer programmer modifies software that is in the public domain, that programmer owns a copyright in the changes he made, even though the underlying work is still in the public domain.[119]

Alternatively, a programmer can incorporate a patented process into a public domain program, thereby preventing others from using the modified program. Thus, whether he uses copyright or patent, a programmer can take a work from the public domain, modify it, and establish his own proprietary rights in the modified version.

Ironically, then, in order to share its software with the public while preventing others from establishing their own proprietary rights in derivative versions,

[117] *See* text accompanying notes 120-136 *infra.*

[118] *See* Jennifer L. Hall, *Blues and the Public Domain—No More Dues to Pay?*, 42 J. COPYRIGHT SOC'Y U.S.A. 215, 218 (1995) ("The performing rights societies ASCAP and BMI will credit and pay royalties for new arrangements of [public domain] works.").

[119] When a programmer modifies software developed by someone else, he usually creates a derivative work. For copyright purposes, a derivative work is one which "substantially borrows the expression of ideas from an existing work." Apple Computer, Inc. v. Microsoft Corp., 759 F. Supp. 1444, 1454 (N.D. Cal. 1991) (citing M. NIMMER, 1 THE LAW OF COPYRIGHT § 3.01 (1990)), *on reconsideration*, 779 F. Supp. 133 (N.D. Cal. 1991), *aff'd*, 35 F.3d 1435 (9th Cir. 1994), *cert. denied*, 115 S. Ct. 1176 (1995). According to the Copyright Act, a derivative work based on an underlying work in which copyright has not expired must be made with permission from the owner of the copyright in the underlying work. Copyright in the derivative work does not extend to any preexisting material used in the derivative work:

> [Copyright extends] only to the material contributed by the author of [the derivative] work, as distinguished from the preexisting material employed in the work. . . . 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 preexisting material.

17 U.S.C. § 103(b) (1994). The rights in a derivative work are limited by the rights of the copyright owner in the underlying work, so the owner of the copyright in the underlying work still has rights in her creative expression as it appears in the derivative work.

the GNU Project needed to establish strong proprietary rights in its works.[120] By licensing its software to the public with terms that prevent others from establishing their own rights in the software, Stallman created an arrangement he calls "copyleft."[121] The GNU GPL gives users permission to copy, modify, and distribute GNU software conditioned on the user's agreement to license all derivative versions under the same terms.[122] Further, users must agree (1) not to establish proprietary rights in the software;[123] (2) to provide the source code to anyone to whom they give the object code;[124] (3) to include in the software notice of the applicability of the GNU GPL;[125] and (4) to accept the software without warranties of any kind.[126]

In accepting the terms of the GNU GPL, a user agrees to license works based on GNU software under the terms of the GNU GPL.[127] This agreement prevents a programmer from establishing copyright or patent rights in the software. For example, the GNU GPL states that any relevant patent must be "licensed for everyone's free use or not licensed at all."[128] Thus, if a user cannot distribute derivative software covered by the GNU GPL without a patent, then the user may not distribute the software at all.[129]

Under the GNU GPL, each distributor must ensure that every recipient of the object code has access to the source code.[130] This requirement can be satisfied by actually distributing the source code with the object code or providing information regarding where the source code may be obtained.[131] Access to the source code allows the recipient to understand how the software operates, change it, and further build upon it.

The source files and the program containing the licensed software must provide notice of copyright, the GNU GPL, and the disclaimer of warranty.[132] If the software is modified and then distributed, the modified software must clearly indicate that it is not the original so that programming errors will not reflect on the original authors' reputations.[133] Finally, the modified program

[120] Or as Richard Stallman said in an interview, "You could also see it as using the legal system that software hoarders have set up against them. I'm using [their legal system] to protect the public from them." Betz & Edwards, *supra* note 12.

[121] Stallman uses "copyleft" as a general term that refers to any kind of copying conditions that permit some form of redistribution, but do not allow any further restrictions to be added during redistribution. "Copyleft says that anyone who redistributes the software, with or without changes, must pass along the freedom to further copy and change it. . . . Copyleft is a general concept; there are many ways to fill in the details." Free Software Foundation, *What Is Copyleft?* (visited Apr. 2, 1997) <http://www.gnu.ai.mit.edu/copyleft/copyleft.html>.

[122] *See GNU GPL, supra* note 14, § 2(b).

[123] *See id.* §§ 2(b), 7.

[124] *See id.* § 3.

[125] *See id.* § 2(c).

[126] *See id.* § 11.

[127] *See id.* § 2(b).

[128] *Id.* at Preamble.

[129] *See id.* § 7.

[130] *See id.* § 3. The source code for a work is defined as the "preferred form of the work for making modifications to it." *Id.*

[131] *See id.*

[132] *See id.* § 1.

[133] *See id.* at Preamble, § 2(a).

must provide the above notices on the computer screen during program execution.[134]

Because the software is licensed free of charge, the GNU Project does not provide a warranty.[135] A distributor, however, is welcome to provide a warranty, for a fee, as a value-added service. The GNU Project wants to encourage programmers to donate their programming time, but legal liability could detract from this goal, since personal liability for defects might discourage programmers from participating.[136]

### B. *Enforceability of the GNU GPL*

The enforceability of the GNU GPL has not been litigated.[137] However, enforceability of the GNU GPL can be evaluated by analogy to shrinkwrap and shareware license agreements. As discussed above,[138] shrinkwrap licenses provide rights under contract and copyright law, focusing on the use of a single copy of the software. Permission to make a backup copy or keep duplicate copies on computers at work and at home might be granted, but redistribution is usually prohibited. Furthermore, some developers attempt to prevent reverse engineering and modification of the software. Shareware licenses, on the other hand, usually permit redistribution. Thus, a shareware license more closely resembles a license between a software developer and a distributor in that permission is granted to make copies and distribute them to others. The GNU GPL differs from shrinkwrap licenses because it includes permission to modify and redistribute, and it differs from both the shrinkwrap and shareware transactions because no money changes hands. Although the law regarding mass-market transactions in intangible goods has not yet solidified and the contract requirement of consideration disfavors nonmonetary transactions, the GNU GPL should be enforceable to the same extent as shrinkwrap license agreements.

The restrictions that the GNU GPL places on users are essentially conditions precedent to a nonexclusive copyright license. As discussed with regard to shareware, if a copyright licensee uses a work in violation of the license provisions, then the grantor may have an action for copyright infringement, breach of contract, or both.[139] Similarly, the GNU Project should have valid copyright infringement and breach of contract claims against users who do not adhere to the terms of the agreement.

Courts have held that an author can pursue claims of copyright infringement if a licensee makes use of the author's work in a manner that is outside

[134] *See id.* § 2(c).

[135] *See id.* at Preamble, §§ 11-12.

[136] *See id.* at Preamble, § 11.

[137] *See* Email from Richard Stallman to Ira V. Heffan (Feb. 20, 1996) (on file with the *Stanford Law Review*) ("We have sent letters demanding compliance, several times a year I'd estimate. We have never had to sue.").

[138] *See* text accompanying notes 64–85 *supra.*

[139] *See* note 96 *supra.*

the scope of the license.[140] The terms of the GNU GPL are structured as conditions precedent to a copyright license,[141] so a user only obtains permission to modify and distribute the work if she agrees to the terms of the GNU GPL. Any breach of these terms should result in liability for copyright infringement.

The GNU Project can also assert contract claims against users who do not abide by the GNU GPL.[142] As with shrinkwrap licenses, the GNU GPL is enforceable as a contract only if it constitutes a valid contract that is not preempted by federal law.[143] In the case of GNU software, some users obtain the software free of charge over the Internet, while others pay for a CD-ROM containing the software. In either case, the relevant requirements for a valid contract are mutual agreement, legal consideration, and subject matter not preempted by federal law.

If the user does not pay for the software, the agreement accompanying the software might be viewed as the transmission of an offer,[144] where the owner of the software, in this case the Free Software Foundation, is the offeror and the user is the offeree. The user downloads or otherwise obtains the offer. Notice of the GNU GPL appears at the top of every source code file and on the screen every time a program runs,[145] so any use of the software clearly provides notice to the user of the existence and terms of the GNU GPL. The user can use the software indefinitely, decide if she likes it, and evaluate the terms of the GNU GPL. If the user agrees to the terms of the GNU GPL, she accepts the contract and is granted the limited right to modify and distribute the software. If she does not agree to the terms, she does not have the right to make or distribute derivative versions of the software.

Under a shareware license scenario, the user's consideration is the money paid for registering the software. Where GNU software is acquired free of charge, the requirement of consideration will not be met by monetary payment.[146] However, the creation, modification, or destruction of a legal relation on the part of the offeree is valid consideration.[147] The license provided by the

[140] *See* Costello Pub. Co. v. Rotelle, 670 F.2d 1035, 1045 (D.C. Cir. 1981); Gilliam v. American Broad. Cos., 538 F.2d 14, 20-21 (2d Cir. 1976); Considine v. Penguin U.S.A., 24 U.S.P.Q.2d (BNA) 1947, 1949-50 (S.D.N.Y. 1992); Microsoft Corp. v. Very Competitive Computer Prods., 671 F. Supp. 1250, 1257 n.4 (N.D. Cal. 1987).

[141] The GNU GPL reads: "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." *GNU GPL, supra* note 14, § 5.

[142] *See* National Car Rental Sys. v. Computer Assocs. Int'l, 991 F.2d 426, 431-32 (8th Cir. 1993) (holding that violation of license agreement could constitute a breach of contract).

[143] *See* notes 79-82 *supra* and accompanying text.

[144] *See* text accompanying note 96 *supra.*

[145] *See GNU GPL, supra* note 14, §§ 1-3.

[146] Although many academics do not think actual consideration is necessary for enforcement of a contract, some courts still continue to require it. *See* Mark B. Wessman, *Should We Fire the Gatekeeper? An Examination of the Doctrine of Consideration*, 48 U. MIAMI L. REV. 45, 46-48 (1993) (declaring that, even though the doctrine of consideration is "out of fashion" among academics as a gatekeeper of contract claims, "someone forgot to tell large numbers of judges" that "the old gatekeeper has been fired").

[147] The Restatement of Contracts states that "[t]o constitute consideration, a performance or a return promise must be bargained for" and that "[t]he performance may consist of . . . a forbearance, or

GNU GPL is a legal relation, as is the offeree's promise to license any derivative versions. Therefore, the GNU GPL seems to be supported by adequate consideration.[148]

Finally, courts question the validity of license agreements that supplement rights otherwise available under federal statutes.[149] The GNU GPL does not supplement the rights available to GNU programmers under federal law. The GNU GPL specifies the terms and conditions for obtaining permission to use the Free Software Foundation's exclusive rights.[150] It does not restrict use of the software in a manner beyond what is permitted under the Copyright Act, and it does not include any additional terms equivalent to federal claims.[151] Therefore, any claim under the GNU GPL should not be preempted by federal law.

## Conclusion

Copyleft was created as a weapon against copyright. But there are reasons besides a complete disagreement with proprietary rights for ensuring public use of a work without abandoning it to the public domain. The GNU GPL encourages the development of collaborative works by ensuring that they will always be available to the public. It can be applied to other works to provide an island of collaboration and public access in a sea of proprietary rights.

The copyleft mechanism is useful for other works besides software. As traditional works are transformed into digital works, they are increasingly easy to copy, modify, and distribute. Digital technology is useful for speakers who have a message they would like to make available to a large audience at little cost. The digital technologies are ideal media for people who are not *selling* content, but are only creating and *providing* it. Authors no longer need publishers to disseminate their thoughts and opinions. Many people will want to develop works collaboratively and allow those that come later to add or change them, thereby standing "on ye shoulders of Giants" instead of on their toes.[152] But they will also want to ensure that their work and all future versions thereof

. . . the creation, modification or destruction of a legal relation." Restatement (Second) of Contracts § 71 (1979); *see also* 1 Samuel Williston, A Treatise on the Law of Contracts §§ 99-103 (Richard A. Lord ed., 4th ed. 1990).

[148] GNU Project software is available at no charge from various sites on the Internet. The software is also available as a CD-ROM from the Free Software Foundation and other vendors. *See* Free Software Foundation, *Software* (visited Apr. 2, 1997) <http://www.gnu.ai.mit.edu/software/software.html>. If a user purchases the software from the Free Software Foundation, the transaction starts to look more like shareware. If the user is not aware of the license when money changes hands, a court following *Step-Saver* might not enforce the terms of the license on the ground that the license resembles a contract of adhesion. If the terms of the GNU license were made available to purchasers before the time of purchase, then the agreement would include those terms.

[149] *See* note 79 *supra*.

[150] The GNU GPL states: "Activities other than copying, distribution and modification are not covered by this License; they are outside its scope." *GNU GPL*, *supra* note 14, § 0.

[151] *See id.* ("The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program).").

[152] Lotus Dev. Corp. v. Paperback Software Int'l, 740 F. Supp. 37, 77 n.3 (D. Mass. 1990) (quoting Sir Isaac Newton's famous phrase).

remain available for the benefit of others. Therefore, they will want to do more than merely dedicate their work to the public domain.

The GNU GPL is different from other nonprofit software licensing agreements. The National Center for Supercomputing Applications ("NCSA"), for example, allows educational distribution of the software and documentation that it develops and distributes on its web site, but it does not permit commercial use of its software.[153] NCSA accomplishes this by relying entirely on copyright and giving educational institutions the right to reproduce and distribute the software while prohibiting commercial reproduction and distribution.

The Computer Systems Research Group at the University of California at Berkeley distributes its "FreeBSD" version of the UNIX operating system for the Intel-based PC platform along with the source code over the Internet. The Computer Systems Research Group licenses its right to copy, modify, and distribute the FreeBSD software, but requires prominent display of its copyright notice in any software or documentation derived from its works and on all related advertising.[154] Like the GNU GPL, FreeBSD allows modification, but unlike the GNU GPL, the FreeBSD license does not mandate that follow-on works be distributed free of further restrictions.[155]

A simplified version of copyleft would permit users to copy but not modify works. This form of agreement could be useful where one wishes to maintain the integrity of a work by preventing modification while allowing free public copying and distribution. For example, when a work states an author's opinion, the author might wish to encourage distribution yet prohibit future modification.

For works that require frequent modification, such as manuals and other documentation, a license closer to the full GNU GPL makes more sense.[156]

[153] *See* University of Illinois Board of Trustees, *NCSA Copyright Statement* (visited Apr. 2, 1997) <http://www.ncsa.uiuc.edu/General/Copyright/Trade.html>.

[154] *See* FreeBSD, Inc., *The FreeBSD Copyright* (last modified Sept. 24, 1996) <http://www.freebsd.org/license.html>.

[155] *See id.* Richard Stallman foresees application of the GPL to other works besides software: "We [the GNU Project] use copyleft for other things—for example, GNU manuals and my own articles. But we don't use the GPL for them. We use simpler kinds of copyleft." Email from Richard Stallman, *supra* note 137. Stallman further explains:

> The GPL contains a lot of complexity meant to deal with the existence of both source code and object code. That complexity is not needed for copylefting other kinds of works.
>
> If you look at the GNU Emacs manual, you'll see a much simpler kind of copyleft. And on articles, I usually use just this:
>
> Copyright 1996 Richard Stallman
>
> Verbatim copying and distribution is permitted in any medium provided this notice is preserved.

*Id.*

[156] For example, the copyleft notice in the GNU Emacs manual simply states:

> Copyright © 1985, 1986, 1987, 1993, 1994, 1995 Free Software Foundation, Inc. . . .
>
> Permission is granted to make and distribute verbatim copies of this manual provided the copyright notice and this permission notice are preserved on all copies.
>
> Permission is granted to copy and distribute modified versions of this manual under the conditions for verbatim copying, provided also that the sections entitled "The GNU Manifesto", "Distribution" and "GNU General Public License" are included exactly as in the origi-

Such a license would include the nonexclusive right to reproduce and distribute the work, as well as the nonexclusive right to prepare derivative versions, subject to the terms and conditions of the GNU GPL.

Works can be placed under copyleft simply by including a copy of the GNU GPL with the work and giving notice in an appropriate place.[157] For example, the following notice would serve to place a work under copyleft:

> Copyright © 1997 [*Author*]
>
> Permission is granted to make and distribute verbatim copies of this work provided the copyright notice and this permission notice are preserved on all copies.
>
> Permission is granted to copy and distribute modified versions of this work according to the terms of the GNU General Public License, included at [*location of GNU GPL*].

The author should include the GNU GPL with the work and refer to the location of the GNU GPL in the notice.

Revisiting the examples described in the Introduction,[158] copyleft would be useful to artists who want to create collaborative artwork. Such an arrangement would encourage other artists to copy and modify works of art, but it would deny anyone proprietary rights in any of the original or derivative works. Similarly, by using the GNU GPL, groups collaboratively writing about a culture, or about their favorite musicians, could ensure that nobody could prevent others from copying or modifying it.[159] If genealogists used the GNU GPL, they could distribute their family trees on the Internet and be assured that later versions would remain available to themselves and others.[160] Doctors could allow broad use of their test instruments while preventing subsequent contributors from asserting proprietary rights in their improvements.

In each of these cases, the use of copyleft encourages the creation of collaborative works by strangers. Copyleft allows everyone to use and update the works while dedicating their collaborative contributions to the public on the same terms. Finally, copyleft encourages commercial use of the work in ways

> nal, and provided that the entire resulting derived work is distributed under the terms of a permission notice identical to this one.
>
> Permission is granted to copy and distribute translations of this manual into another language, under the above conditions for modified versions, except that the sections entitled "The GNU Manifesto", "Distribution" and "GNU General Public License" may be included in a translation approved by the Free Software Foundation instead of in the original English.

RICHARD M. STALLMAN, GNU EMACS MANUAL: ELEVENTH EDITION: VERSION 19.29 (1995).

[157] *See* 1 GOLDSTEIN, *supra* note 58, § 3.6.2 (describing what constitutes reasonable notice of a claim to copyright).

[158] *See* text accompanying notes 1-5 *supra.*

[159] The GNU GPL would not prevent anyone from selling a copy of a work, but the seller would not have the right to prevent his buyers from further copying. The ability to resell provides an incentive for a seller to add value, such as packaging the work in a particularly useful or convenient manner, but the work still remains free.

[160] Under *ProCD*, the coverage provided by the license could include otherwise uncopyrightable information in a database. *See* ProCD, Inc. v. Zeidenberg, 86 F.3d 1447, 1453-54 (7th Cir. 1996).

that add value, but assures continued public access to the underlying content.[161] Copyleft ensures the collaborative development of these works.

[161] For example, it would be consistent with copyleft for a company to publish the genealogist's data commercially, but under the terms of the GPL, the company could not assert any rights in the data.

## Appendix

## GNU GENERAL PUBLIC LICENSE

Version 2, June 1991
Copyright © 1989, 1991 Free Software Foundation, Inc.
59 Temple Place - Suite 330, Boston, MA 02111-1307, USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

### PREAMBLE

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software—to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

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 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. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

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.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

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 precise terms and conditions for copying, distribution and modification follow.

## TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. 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 Public License. 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. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

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.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

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, when started running for such interactive use in the most ordinary way, 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. (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.)

These requirements apply to the modified work as a whole. 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.

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, mere aggregation of another work not based on the Program with 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.

3. You may copy and distribute the Program (or a work based on it, 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:

> 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 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.)

The source code for a work means 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. However, 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.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. 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 (or any work based on 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.

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 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.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), 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 as a consequence 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 you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

## How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

> *One line to give the program's name and an idea of what it does.*
>
> Copyright © 19yy *name of author*
>
> This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.
>
> This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.
>
> You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

> Gnomovision version 69, Copyright © 19yy name of author

> Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type 'show w'. This is free software, and you are welcome to redistribute it under certain conditions; type 'show c' for details.

The hypothetical commands 'show w' and 'show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than 'show w' and 'show c'; they could even be mouse-clicks or menu items—whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

> Yoyodyne, Inc., hereby disclaims all copyright interest in the program 'Gnomovision'(which makes passes at compilers) written by James Hacker.
>
> *signature of Ty Coon*, 1 April 1989
>
> Ty Coon, President of Vice

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.