Rod Dixon

OPEN SOURCE SOFTWARE LAW

CD-ROM
INCLUDED

Open Source Software Law

DISCLAIMER OF WARRANTY

For a complete listing of the *Artech House Telecommunications Library*,
turn to the back of this book.

Open Source Software Law

Rod Dixon

Artech House
Boston • London
www.artechhouse.com

Library of Congress Cataloging-in-Publication Data

A catalog record for this book is available from the U.S. Library of Congress.

British Library Cataloguing in Publication Data

A catalog record for this book is available from the British Library.

Cover design by Yekaterina Ratner

685 Canton Street
Norwood, MA 02062

International Standard Book Number: 1-58053-719-7

A Library of Congress Catalog Card number is available from the Library of Congress.

10 9 8 7 6 5 4 3 2 1

To my brother Christopher

Contents

Preface

Free software and open source software is freely licensed, not sold; that statement alone might cause those who yelp and rail against free software to stop reading, but it is undisputable that many software vendors still prefer to do business only by collecting substantial fees from software licenses, and many have earned billions in revenues doing just that—selling software, hiding computer source code, and suing anyone believed to have stolen the source code that the vendor originally copied from someone else.

Software development is often a messy state of affairs, and one might conclude that a movement that promotes the development of free software, where the source code is open, the software is free, and others are encouraged to take the code and do with it as they desire, is doomed to failure. Indeed, the opponents of open source have probably wished for exactly that unfortunate fate. If you doubt that, reconsider while rereading some of the bold headlines in your daily newspaper concerning Hollywood or almost any entertainment industry colossal force from the latter half of the 1990s or more recently; in doing so, you will not fail to notice the explosive level of litigation that has been initiated against open source proponents. Certainly, you have read about the alleged hackers who supported Napster or those who dared to view a movie recorded on DVD by using a computer that did not have an operating system produced by Microsoft. These hackers and open source proponents were pursued in courts from New York to California; not even the First Amendment could beat back the onslaught of Hollywood's mastrodontic entertainment industry.

Of course, open source is about much more than whether Napster's online database of digital sound recordings constituted a useful peer-to-peer or file-sharing tool or whether the motion picture industry should be permitted to further tighten its grip around the throats of movie viewers who watch DVDs on computers. Indeed, open source reengages a debate about the fundamental question of whether the public should continue to maintain support for wide-ranging, government-backed, monopoly protections of intellectual property in the digital domain; from the perspective of those who fought to stop Napster, that question was answered with a "why not?" But perhaps the public interest is better served by asking, "why?" This book does not provide an answer, but it does provide a perspective and a focus upon one important consideration in the increasingly popular digital domain: the nature of the legal regime supporting the open source and free software community. This book offers one view of how the distribution of free software and open source software is accomplished, and sets out the legal starting point upon which that goal is achieved.

The bedrock of the legal strength of the open source community is its innovative software licenses, including the well-known GNU General Public License (GPL). (GNU is a recursive acronym meaning "GNU's Not Unix.") These public licenses serve the objectives of the open source community in a number of ways. A principle purpose of this book, therefore, is to put in plain words how open source software licensing accomplishes the goals of the open source and free software community. As a practical matter, this book offers guidance on how open source license templates may be used by new adherents of open source. With the aid of this book, you may avoid wasting valuable time creating software licenses entirely from zilch or inefficiently copying on-line licenses that serve ends far removed from your own needs.

Not to belabor the point about software licenses, but you might think that open source license drafting is simple because the goals are straightforward. If so, a word of caution is warranted for those who may want to draft their own open source license: open source software licenses are simple, but not simplistic. For example, the popular public software license GNU GPL is a free software license that is the essence of simplicity, if legal instruments may be viewed that way, but its robust aspects hide how extensively the GPL covers complex licensing matters. The GPL may be viewed as simply a copyright license for most licensees who simply want access to open source or free software within the bounds that constitute copyright; namely, those end-users who merely want to copy, use, or modify GPL'ed software benefit a great deal from the GPL's simplicity in that the end-user's conduct is deemed permissible under the GPL without any further action by or restriction on the end-user.

If, however, an end-user or licensee wants to redistribute a GPL'ed work, the GPL imposes a restriction that the licensee must or is considered to have consented to as a matter of contractual obligation; namely, licensees must comply with copyleft, which is an intricate software licensing concept that is explained in Chapter 2.

The GPL is one of the most popular open source licenses, but new entrants to the open source community often have needs that are not met by the terms of the GPL, despite its popularity and comprehensiveness. Accordingly, a broad range of free and open source license templates are presented in the Appendix that can be edited for use in preparing your own license during negotiation and planning.

The task of *drafting* a software license (or what some refer to as rolling your own license) is not simple for a number of reasons, including the fact that, essentially, a software license is a unique form of contracting aimed at governing the relations in a complex technology transaction. Regardless of whether you roll your own license or have a license drafter begin the task, you must make an effort to understand the underlying technology involved in the transaction to properly effectuate your objectives or the objectives of the client, developer, or copyright holder. Therefore, the license templates contained in this book are mere starting points on the journey to success. Of course, nothing in this book replaces the legal advice that you will need before final execution of your open source software license; even so, this book should provide ample guidance on the journey of open source licensing. For those not fixated on the how-to of open source licensing, this book may serve as a backdrop to further explorations of the open source and free software communities.

Conventions Used in This Book

Throughout this book, I use the term *open source community* to refer to both the free software and open source communities. In Chapter 2, I introduce what I view as the pertinent distinctions between open source software and free software, and review the unique philosophies that distinguish these two important factions of the large community of free and open source software adherents. While some may prefer the caption *free software* instead of *open source*, I have selected open source as the anchor for this book because the theme of open source is consistent with one of the objectives of the book, namely, to present the legal regime that supports a wide-ranging selection of various business models engaged, in some manner, in nonproprietary software development.

More precisely, free software licensing tends to be less forgiving of alternative approaches to software distribution than those that border along the edges of proprietary software development, and is less tolerant of the development of open source codebase that is less likely to evolve as only open source software. Although this book embraces open source software development because the software development model is more permissive as to what might be perceived of as marginally open source development or on the edge of closed software distribution—in other words, a wider base of models of production—it is intended to cover the full spectrum of free software licensing issues that may arise among different developers or participants in the free software and open source software community.

The software licenses included in this book are open source software and free software license templates. I have chosen to use excerpts as license templates to avoid or diminish duplication of license terms. One word of caution for those who intend to use this book as helpful guidance in using preexisting licenses to draft a license: Be careful to view existing licenses as license templates or ideas that you can build upon to create your own license. If, instead, you decide to copy an existing license, you should determine whether the license itself carries copyright terms. An increasing number of authors of software licenses seem to be asserting copyright interests in the license. Strictly speaking, if so, you may use the ideas within the license text, but not its expression.

The aforementioned caution notwithstanding, although the text of the run-of-the-mill software license (whether the license covers an open source software product or otherwise) is certainly expressive in some qualitative sense, the expressive quality does not tell us very much about the copyrightability of a software license, which I suspect is minimal. I think the application of the originality requirement of copyright law to software licenses is not a simple matter, but in most cases it would result in thin copyright protection, if any, for those works.

As a practical matter, lawyers and license drafters are acutely aware that no one wants to be too original in drafting a legal instrument like a license or contract. Instead, a careful lawyer drafts licenses and contracts by using legal terms of art, common phrases (expressed with a string of words that have ostensibly merged with the underlying ideas) (merger doctrine), and reliably interpreted clauses. In this light, the text created is likely to receive only thin protection under copyright. Consequently, one should find few obstacles to using the ideas within preexisting open source software licenses, but should also get permission when necessary.

The last concern I raise on licensing is that the licenses included in this book are presented as examples of important licensing issues that have arisen from popular or interesting open source licenses in actual use, along with my commentary on what the license terms mean and what issues may be implicated by those terms. The inclusion of a license template or the omission of others does not carry a stamp of approval or rejection. The list of established open source licenses seems to grow weekly with the same abounding energy that flows through a movement that is on the edge of bringing about systemic change through the growth and development of an industry already known for its fast pace and expansion. There are many open source licenses currently in use, and a review of all of them is far beyond the scope of this book. The book contains guidelines that should be helpful in drafting a license in much the same way a contract-form book is used by lawyers. One should be aware of the relatively novel use of open source software licenses when drafting any open source software license; notably, no court has ruled on their validity or lack thereof, and while there is no concern that typical open source licenses will be judicially enforced, open source licensing is undoubtedly on the bleeding edge of cyberlaw.

Acknowledgments

I want to express my sincere thanks to Lawrence Rosen, chief counsel of the Open Source Initiative (OSI), who encouraged me to embark upon this project and even began the journey with me. Many thanks to all of the active participants of OSI's license-discuss listserv, who, from the winter of 1999 until the present, have shared ideas while reviewing pending open source licenses in a thoughtful, informative, and highly constructive environment. In addition to my own contributions, I learned a great deal from the members of license-discuss. I would also like to thank the editors at Artech House Publishers, Mark Walsh and especially Barbara Lovenvirth, whose supportive encouragement and noteworthy assistance during the writing and development process for this book have been inspiring and very helpful. Finally, there is no doubt that many of my thoughts on open source software have been shaped and affected by a list of individuals far longer than my acknowledgments or citations could represent, but, ultimately, any errors in judgment or sins of bias are my own.

1

Open Source Software

You cannot hack what you cannot see; freedom requires openness.

—A slogan posted on Usenet unsigned

Open Source Means Free

If you have recently reflected upon what it might mean to create *open source* software, the Internet cliché often associated with hackers—that "free software means free speech, not free beer"—may come to mind. Even so, the metaphor leaves you wondering why anyone would feel good about volunteering to write source code or computer software for someone else. After all, even though there must be a number of ways to develop and distribute software, giving it away because it supports something as amorphous or abstract as freedom of expression seems suspiciously similar to getting highly skilled individuals to work for nothing. In other words, there seems to be something odd about open source.

The truth is that there is something that appears twisted about the motivation of those who freely participate in open source projects; twisted, that is, until you understand the philosophy of the open source community and compare the methodology of creating open source software with alternative proprietary models of software development. This book attempts to untwist what at first blush may be a distorted view of the open source community and to provide a fresh perspective and an engaging analysis of the legal framework that exists to support the increasingly popular Internet-based open source and free software community.

To provide a context for understanding the law of open source software, this book explores the formal and legal aspects of two innovative ideas: namely, that software should be offered to users with open access to the source code and that end-users should be freely able to modify, copy, or redistribute the software they have legally acquired. These ideas are significant because they stand well-accepted notions concerning copyright on their head and demonstrate that our long-held understanding about the nature of human motivation may be fundamentally misguided.

Open source software authors want the widest dissemination possible of their information products. For these authors and developers, the traditional view that copyright holders must have access to the greatest economic gain possible in exchange for producing a copyright-protected work is not only outdated, but unfounded. The traditional view that copyright holders must have access to the greatest economic gain must yield to the more precise and superabounding idea that copyright holders really have one goal in mind, namely, maximizing the value of the work through achieving the widest dissemination possible. As an introduction to software licensing in the information age, this book explores these themes along with the general theme connecting computer software to the Internet through the sharp insights of the open source and free software community.

Although open source is rooted in an alternative method of software development, the bedrock principle of the open source community goes beyond the interests of the professional programmer. Open source promotes the idea that when Internet or computer users are given access to the source code of the software they use, the end-users are empowered with the tremendous freedom to create or improve *by and with* their own computing tools. When end-users are empowered to read, redistribute, or modify the source code of their software, not only does the software advance the knowledge base of the public domain, but the users' lives are improved through the empowerment of computing.

Certainly, one can hardly imagine an interesting world filled only with programmers, but it requires only slight imagination to envision a world embedded with computers and computing devices that aid or control our daily routines. Like the binary code used to run computers today, the choice is off or on; the difference is akin to being read to or learning how to read. What choice would you make?

In its most basic sense, this conception of the empowered end-user does not rely upon a simple-minded notion that if we throw source code at a problem, the problem will be solved because we are living in the information age; instead, the concept of the empowered end-user involves a reawakening of the creative spark that people demonstrate when interfacing with puzzling work or difficult tasks. With regard to software programs, for example, only a creative spark of curiosity in an empowered end-user would lead to the end-user's determination that the spell-checker and dictionary contained within a word processing program was purposely abridged to contain only the words that a software vendor wanted users to use in their writing and thinking. Missing from the dictionary were words that the software developer found offensive, disliked, disfavored, or feared might aid users in recalling the names of competitive products. The point is not that we must fear a powerful force who might attempt to control our thoughts and desires—that is the province of science fiction—rather, the point is, more simply stated, that open access to information (such as source code) empowers users to discover offenses against freedom and then change them.

As simply stated as the principle is, its nuances are easy to overlook; open access to information, for example, is not the same as shared access. Quite ironically, perhaps, some forms of shared access to source code constitute the opposite of end-user empowerment. Shared access to information emphasizes the use of prebuilt modules without fundamental access to the source code for the modules. The restricted use of prebuilt or shared code can result in the dumbing-down of programming. In some circumstances, sharing ensures that programmers do not understand the code they are using and, perhaps much worse, the knowledge accessible by programming disappears into the source code unknown or unknowable by the programmer. Shared access to information is not open; it is closed, restricted, secret. Hence, shared access to information benefits only those who participate in the sharing. While the tendency to benefit few, rather than many, would be counterproductive to the goals of the open source community, which considers the aims of empowering all programmers and computer users as primary objectives of the community, it is noteworthy that those who champion the benefits of shared access as a viable substitute to open access also promote support for a legal regime that privileges proprietary source code over open source development models.

That notwithstanding, to some, even the lofty goals of open source and free software exceed the interests of most members of the open source community, and to some extent the concern over the distinction between *freedom* and *openness* goes to the ongoing debate between two of the most visible camps of the open source/free software community, namely, whether freedom or openness better describes the essential feature of the software distribution model used by the community. On a larger scale, the argument that freedom requires openness goes to the very essence of freedom in cyberspace. In other words, open access to source code is an important aspect of the freedom to prevent or undermine the current effort by government and certain intellectual property holders to redesign the technology of the Internet to establish control over the network.

As closed code slowly replaces the open code of cyberspace, the owners of the closed code may not only restrict the unprecedented freedom to innovate on the Internet, but may also reuse their closed code to unilaterally regulate cyberspace or impose governing systems. The freedom to hack or tinker with the code brings an element of democracy to code that imposes a governance system upon cyberspace. In this regard, it is clearly evident that the epigrammatic jingle that, "You cannot hack what you cannot see; freedom requires openness" contains a profound statement concerning much more than a squabble over what name best represents a community's goals.

Classifications of Software Distribution

Software and the process of software distribution may be described and classified in a number of different ways, depending on context and goals. Some software creators, for example, only develop components or snippets of code that provide connections between or among different applications, while others may develop software for a vast enterprise, but never sell software to third parties. Since it is often difficult to determine under what conditions and for what purpose a computer program may have been developed without reading a software license, shorthand references to the wide-ranging models of software distribution have become infused with terminology and classifications that are occasionally bewildering. To avoid adding to the confusion, this chapter begins the assessment of the law of open source software with a discussion of software classifications in a context relevant to open source. The distinctions among the classifications of software are important, because consistent use of the appropriate terms often facilitates or advances our discussions about software.

In some respects, off-the-shelf software, shareware, freeware, vaporware, and open source software might all be considered distinct forms of software distribution or marketing despite the fact that consumers of software may view them similarly. Vaporware, for example, actually refers to the absence of software marketed well before it is produced. On the other hand, shareware and freeware seem similar, since end-users often use the programs without paying for them, but neither can be considered open source software, and shareware developers usually rely upon a licensee's promise to pay for software that is used beyond a preset time period. On the other hand, freeware does not require any payment from the licensee or end-user, but it is not precisely free software, despite the fact that to an end-user the software is acquired in what appears to be an identical manner.

Freeware is provided to end-users at no cost, but free software provides more benefits than simply delivering a no-cost product—indeed, for the end-user, there may be circumstances where the monetary cost of acquiring free software exceeds the cost of freeware. How is it that free software may cost more than freeware, but fundamentally be "freer" than freeware? Providing an answer to this question is one of the objectives of this book.

In the context of software classifications, it is noteworthy that the many divergent models of software development and distribution did not arise haphazardly or to confuse consumers or end-users; instead, the diversity of software production is a reflection of the progressive ingenuity that percolates throughout the information technology industry—particularly in the United States. Many commentators estimate that since 1980 the American computer software industry has grown faster than the rest of the American economy and in the 1990s was considered the engine of growth for the entire economy. Today, at least half a million Americans are employed in the computer software industry and, despite occasional downturns, the industry continues to produce software that likely accounts for over 70% of the world market in software distribution.

Notwithstanding the ingenuity and innovation that seems inherent in information technology, there is a downside to diversity; namely, the labels or signals that are meant to inform software end-users of the specific type of software being purchased and the conditions that attach have shades of meaning so thin it is often too difficult to slice the correct meaning from the label or classification. It is increasingly common for software developers to use imprecise terminology to describe software classifications or marketing strategies. Words do not have a consistent or clear meaning, and occasionally technical terms gain a twisted, but popular, marketing meaning, which is so distorted from its original meaning that the term is rendered meaningless in its technical sense [1].

This lack of precision in the use of terms has had spillover in the open source movement, and those who have attempted to hijack open source practices have had considerable success in confusing the use of labels and terms in the popular media. Quite remarkably, there is still a great deal of confusion and disagreement over proper uses for the label or classification *open source*, which arose as a label because many viewed the term *free software* as misleading, imprecise, and, to some extent, weighted with controversial inferences for those who are in the business of making money from software. It is quite a paradox that the classification of software within the information technology industry reflects growing imprecision, since industry insiders seem to spend considerable effort struggling over classifications.

Interestingly enough, time spent reflecting upon classifications is anchored in the deep roots of Western civilization, at least as far back as ancient Greece. Aristotle was a philosopher who was so fascinated with the clarifying force of classifications that he developed a system of classifications for the subjects he studied and taught; he divided themes by arranging them under topics and categories, examined arguments by cataloging them by their persuasive force, and sorted entire fields of study by ordering them under arts and sciences. Aristotle's method is still with us. We still rely upon Aristotelian classifications in the fields of science and art and other subject matter, including, for example, disciplines studied throughout the curricula of higher education.

Although it is clear that the open source and free software community is about freedom and openness, the community is still developing its own lexicon, identity, and, most importantly, its classifications. Therefore, if you are new to the open source genre, be patient with the community's lexicon and syntax—some concepts have evolving meanings or unclear uses; open source is still a relatively young movement which has had its most significant growth coincide with the increasing use of the world's growing networked infrastructure—the Internet. Occasionally, questions arise concerning whether the community is properly denoted by the open source community or the free software community or whether it makes a difference how the community is identified or classified. Unfortunately, a search for the answer has not generated a bright-line classification. The community seems too dynamic for static classification or universal agreement on identity.

Throughout the book, however, basic issues are reviewed and analyzed concerning where the movement is today regarding its response to what constitutes open source or free software; the aforementioned notwithstanding, it is important not to become closely tied to a particular classification set. The objectives of this book are to describe the open source community: to understand the concepts of open source and free software, and to unearth the principle aspects of the community's software licensing scheme. Consequently, the issues that are out of the range of the book's objectives will not be fully addressed, notwithstanding that some of those issues may raise interesting fundamental questions.

Perhaps, by default, popular technology and business media outlets have accepted the self-selected term *open source* as a preferred point of reference for the open source/free software community. Although the open source/free software community has factions within its own group with differing viewpoints on some of the objectives of the open source community, the free software/open source software community throughout, regardless of name or classification, stands for the same innovative views of software development and distribution: that software should be offered to users with open access to the source code and that end-users should be freely able to modify, copy or redistribute the software under open source business models.

These two aims are innovatory because they offer new senses for what the nature of human motivation might be in the context of computer software. Why do programmers program, if not for the economic incentive promoted by copyright? Aside from the reality that open source software authors want the widest dissemination of their information product, open source authors believe in the objectives of community. It is in this regard that open source developers have leveraged the Internet into a vast source of potential of open source project participants.

Of course, software is widely viewed as intellectual property because that is how we treat it under our legal regime, not as a result of some inherent or salient quality of software rendering it more like property than anything else—a point made clear in Chapter 4. In fact, the connection between copyright and property is easily and often overstated: notwithstanding the analogic similarity between intellectual property concepts and common law property, neither copyright nor, more generally, intellectual property are derived from legal theories historically rooted in the law of property. Still, our views on the level of human motivation required to create works or to invent useful objects is tied to the idea that such motivation would not arise if the property value in such creations could not be captured by those doing the creating or inventing. When the U.S. Congress first decided to include software among those items in which the motivation to create is greatly enhanced by the reward system embraced by copyright law, software became anchored to the subject matter of intellectual property. In the next section of this chapter, we will briefly survey software classifications and the assignment of rights under the intellectual property system. A more detailed discussion of copyright issues is introduced in subsequent chapters. The subject matter of intellectual property law includes the rights and obligations related to copyright, patent, trade secret, unfair competition, and trademark law, with each playing a different, but interrelated role within our system of intellectual property law.

Facing increasing competitive pressures from Microsoft's distribution of its Web browser, Internet Explorer, in January 1998, Netscape announced it would stop distributing its Web browser software solely as a proprietary product. Netscape decided that its best chance of recapturing its strong competitive status as a Web browser developer was in developing and distributing its browser as an open source product. Today, the development of Netscape's Web browser as an open source product is coordinated and managed at Mozilla.org, as will be discussed further in Chapter 6. Mozilla.org operates as a collective of hundreds of developers who volunteer their labor and are granted access to the entire browser codebase for any use, including distributing the resulting product under the terms of a public license.

Netscape's adoption of open source development marked one of the first times that a company audaciously released its software as an open source product after selling the same product solely as a proprietary product. Two influential members of the hacker culture, Eric Raymond and Bruce Perens, followed up on Netscape's announcement by urging other software developers to join the open source software development model as well. Emphasizing the pragmatic grounds of reliability, cost, and strategic business risk, Raymond and Perens encouraged proprietary developers to follow Netscape into the open source community. In part, Raymond and Perens had hoped to overcome some of the damaging opinions of free software held by proprietary software developers by bootstrapping the favorable media attention brought to open source by Netscape's adoption of the model. They also recognized the publicity surrounding Netscape as an opportunity to secure a favorable label, *open source*, for the community of developers working on free software, which could be used as a strategic brand or signal for the community's more commercially oriented focus.

Raymond and Perens further assisted in securing the future of open source by establishing the Open Source Initiative (OSI). OSI is a nonprofit association dedicated to managing and promoting a standard set of criteria of distribution terms that an open source software license should conform to in order to obtain the benefit of using OSI's certification trademark or having a license listed on OSI's Web site as an approved open source software license. The criteria used by OSI are referred to as the Open Source Definition (OSD) [2]. The OSD is posted on OSI's Web site at www.opensource.org. The benefit of distributing software under OSI's certification trademark or having a license listed on OSI's list of approved licenses arises from the clear and widely recognized signal that software distributed under an approved license is a genuine open source product.

Why adopt a certification trademark program? Regrettably, the term *open source* was not trademarked prior to its widespread popularity as a descriptive term referring to itself (generally, such terms cannot be trademarked). Subsequent to the term's widespread popularity, some software developers began using *open source* as a marketing term in order to take a free ride on the increasing popularity of the open source movement. To thwart significant erosion of the open source community's ability to distinguish its projects from those who deceptively adopted the term *open source* as a marketing ploy, the community understood that there was a critical need to develop a reliable way of informing others whether a piece of software really is open source; the answer to this need was the adoption of the OSD and the establishment of a certification mark program. This section of the chapter focuses upon how OSI uses the OSD to manage what ostensibly spells out the essential qualities of open source software.

OSI relies upon a number of volunteers and volunteer organizations that consider themselves open source adherents rather than necessarily free software adherents. The distinction generally has to do with the degree to which the open source community may tolerate departures in licensing terms and accept divergent software development models from software creators in order to broadly encourage business entities to participate in distributing open source software. Generally, free software proponents have less tolerance for software development projects that allow participants to use open source software for closed source or nonfree software development projects than the more tolerant open source membership.

Even so, the groups have closer ties than some press accounts indicate, and there is a desire to provide users on both the Free Software Foundation (FSF) Web site (www.fsf.org) and the OSI Web site (www.opensource.org) with information on license templates. In addition to FSF and OSI, there are a number of additional groups involved in open source and free software licensing, including Debian (www.debian.org) and Software in the Public Interest (www.spi-inc.org).

OSI publishes the OSD. OSI manages the OSI Certified Open Source Software certification mark and program, which enables open source software users to be certain that the software they are using is legitimately using the label *open source*. The open source community exemplifies a cutting-edge software development initiative that has advanced from the edges of cyberspace to the forefront of the Internet's drive toward commerce.

As an increasing number of businesses adopt the open source model of software development and come to rely upon its template of software licenses to achieve their business objectives, Internet users will become increasingly aware of the fact that what accompanies the knowledge of source code in software is the power and control of the software used to conduct electronic commerce, send e-mail messages, or surf Web sites. As Internet and computer users discover that genuine control over privacy, governance, and freedom of expression may be obtained only through direct access to source code, the value of open source software will increase. But imitators of open source software will continue to distribute unreliable software. Hence, an important goal of OSI is to maintain end-user awareness of genuine open source software in light of the fact that the competitive pressures of the future of software development might make this goal increasingly difficult to achieve.

OSD

The OSD is viewed as a type of manifesto of the open source community. Its primary drafter is Bruce Perens, who is a cofounder of OSI. The OSD is not an open source general public license or even a template for an open source public license. Instead, the OSD also helps ensure that consistent criteria are applied when OSI evaluates whether to approve a submitted license to be added to the publicly available list of approved open source licenses posted on http://www.opensource.org. Hence, the OSD has become a kind of bible of open source licensing. Although the OSD was not drafted specifically for the purpose it is used for today, it is widely regarded as a helpful guideline on the expectations of OSI in its review and approval of open source licenses. Free software adherents may want to compare projects and licenses listed on www.fsf.org to determine the unique aspects of software projects guided by the FSF.

The OSD is a type of social contract among programmers and users of open source software. The OSD sets forth principles and policy guidelines that assist open source participants in drafting new licenses that are consistent with the goals of open source software development and distribution. Since the OSD is intended to be a generalized document, all of its principles need not be implemented by every open source license. Although the specific terms of the OSD need not appear in any given open source license, it is critical that open source licensees remain mindful that their own distribution license should not directly conflict with the spirit or words of the OSD. In this respect, the OSD is most useful to those who must draft a new open source license as a result of concluding that none of the existing or approved open source licenses provide a suitable license template for their particular software development objectives.

Since there are considerable benefits that accompany the ability to accurately label or describe a software program as open source software, the OSD, along with the determination to allow participation in the OSI certification mark program, provides a meaningful signal to software users as to whether software is legitimately open source software or merely software sold as a marketing gesture intended to confuse software and Internet users. Consequently, the OSD is an important tool that may aid new open source participants in the drafting of a new open source license that implements, with careful legal precision, the nuanced guidelines that enable disparate business models to easily fit within the goals and objectives of open source projects.

The OSD currently has 10 articles [3]. The first article is captioned "Free Redistribution," and it provides that an open source "license shall not restrict any party from selling or giving away the software as a component of an aggregate software distribution containing programs from several different sources." In addition, under article 1, an open source license cannot "require a royalty or other fee for such sale." The requirement that open source software be freely distributed covers all software distribution transactions to the general public (including mass-market consumer transactions), as well as restricted distributions except those that are internal, in-house, or within a single entity having a defined or limited membership. Article 1 is a "know-your-rights" clause, meaning that free software is free only if the licensee or end-user knows that it is free. Therefore, the license should conspicuously express that the software is open source.

An aggregate software distribution refers to the common practice of distributing multiple software applications on a single CD-ROM or in a single package from many different sources. Many open source operating system packages are distributed in this manner, and doing so is entirely consistent with open source objectives. Article 1 primarily guides the licensor in drafting the grant clause or the distribution provision of the open source license. The open source software distribution must remain as free software. The software is always distributed with a license provision that grants end-users or licensees the right to receive a source code version of the software and the permission to modify the software or freely use portions of the source code in a newly derived open source program.

Notably, nothing in article 1, nor the text of any open source software license approved by OSI, suggests that free distribution of software should be accomplished by expanding the licensor's copyright interest beyond traditional copyright. Consequently, although open source vendors may distribute software freely and without cost to the licensee by providing paid-for services or hardware, doing so must be accomplished without constituting copyright misuse. One possible misuse is the licensing of computer software so that only the licensor can maintain or service its products. In addition, article 1 makes it inconsistent with open source software licensing practices for licensors to adopt subsequent licenses that unilaterally alter the express agreement of the parties. For example, an open source software license binding a licensee on January 2, 2003 to a term that allows code forking cannot be unilaterally replaced with version 2.0 of the same license template that is created on February 1, 2003 and prohibits code forking and is simply posted to the licensor's Web site. The February license term prohibiting code forking is not enforceable against the licensee as a valid open source license unless the licensee consents to the new license. The terms cannot be unilaterally changed after the fact.

In article 2, captioned "Source Code," the OSD provides that an open source software program "must include source code, and must allow distribution in source code as well as compiled form." In the event that a computer program is not distributed with source code, "a well-publicized means of obtaining the source code for no more than a reasonable" cost of production must be provided to end-users. Under article 2, source code is assumed to be the preferred form an end-user or programmer would use to modify a software program.

Article 2 encapsulates what is ostensibly a rule of open source software law, namely, that access to the source code is what makes open source open. Although there is some debate as to what really constitutes source code, these conceptual issues are beside the point. Access to human-readable code that is intended to fulfill the dual purposes of expressing what the software does as well as providing computational instruction that will ultimately result in a running program if executed properly constitutes code that should be well documented. Source code that is exceptionally difficult to read either because it is not documented or is cryptically expressed is considered to be obfuscated source code that does not comply with the access-to-source-code requirement.

In article 3, "Derived Works," the OSD requires that open source licenses contain provisions that allow licensees or end-users to modify the software program or to create a derivative work based on the program, as well as permit the end-user or licensee to distribute the modified program (or derivative work) "under the same terms as the license of the original software." Section 106 of the Copyright Act enumerates a copyright owner's *exclusive rights* in copyrighted works. Section 106 authorizes the right of the copyright holder to reproduce its copyrighted works and prepare derivative works. Consistent with these copyright interests, the licensor necessarily licenses to licensees the right to reproduce; otherwise, a licensee would not be able to lawfully redistribute copies of open source software; the rights to make copies and publicly distribute copies of a work ostensibly fold into overlapping copyright interests with regard to digital works.

Notably, minor modifications to software do not produce a *derivative work*. A derivative work is “a work based upon one or more preexisting works” that is “an original work of authorship,” and thus copyrightable itself [4]. Not every alteration to a copyrighted work results in a derivative work; the variation must be *original* and *substantial*. Courts have made clear that unauthorized use of a copyrighted work is not necessarily infringing unless it conflicts with one of the specific exclusive rights conferred by the copyright statute [5]. In other words, there is no provision or principle of copyright law that equates the Copyright Act’s list of exclusive rights with an absolute right to control a copyright-protected work beyond the limits or outside the scope of the Copyright Act in the name of copyright.

Article 3 further distinguishes open source software from proprietary software licensing. It is highly unlikely that article 3 will ever become a standard practice used by any proprietary software developer when distributing proprietary software. Article 3 means that end-users and licensees need not rely upon fair use, additional licensing terms or outright infringement to gain access to the software program’s expression and ideas, and freely build upon it. In other words, article 3 requires that open source users not only get an opportunity to see the source code, but also obtain the opportunity and right to use it.

Articles 4, 5, and 6, “Integrity of the Author’s Source Code,” “No Discrimination Against Persons or Groups,” and “No Discrimination Against Fields of Endeavor,” respectively, are aimed at business practices by similarly permitting open access to source code, but imposing restrictions on burdensome license provisions intended to limit distribution of modified software to protect an author’s integrity or to lock out certain groups or business forms. For example, an open source license may neither preclude distribution of all forms of derivative works by licensees (even if the basis is to protect the integrity of the codebase), nor preclude educational institutions from distributing derivative works, but permit all other groups to do so.

Article 7, “Distribution of License,” reinforces the open source practice of using a single license. In this regard, the rights granted under an open source license “must apply to all to whom the program is redistributed without the need for execution of an additional license by those parties.” Article 7 ensures that open source licenses are effective without additional steps to execute the license; notably, licensors are precluded from adding steps for the licensee to follow before the license binds both parties. In the words of Bruce Perens, "you should not have to sign and mail in a license to use or distribute a piece of Open Source software. Instead, the license should apply to you automatically." More importantly, article 7 should be read to emphasize that open source licenses will be enforceable if they provide adequate notice to open source participants of their terms and give those participants a reasonable opportunity to review the terms and the right to reject them.

Articles 8 and 9, "License Must Not Be Specific to a Product" and "The License Must Not Restrict Other Software," respectively, are aimed at foreclosing license traps for end-users. The open source license cannot contain a provision wherein the rights attached to the program depend upon the program's inclusion in a particular software distribution bundle, nor may an open source license place restrictions on other software that is distributed along with the licensed software. For example, the license must not insist that all other programs distributed on the same medium must be open source software. Article 10 contains a recently proposed addition to the OSD that is intended to allow potential licensees to express consent to license terms through a number of methods that might be described as "click wrap" interfaces.

The 10 articles of the OSD provide OSI with its primary set of guidelines and principles for determining whether a given software license constitutes a genuine open source software license. If the distribution terms of the license comply with the terms of the OSD without violating its spirit, then the license is likely to be approved as an open source license.

Endnotes

[1] Moore's Law is a notable example. Moore's law is supposed to be a postulation that the speed of a microprocessor will increase at a predictable rate of time. Throughout the 1990s, microprocessor manufacturers like Intel Corporation brought new processors to market on an approximate 18-month schedule, which met the time frame of Moore's Law with remarkable marketing consistency.

[2] The OSD is set forth in Appendix 1A, at the end of this chapter.

[3] Please see Appendix 1A.

[4] 17 U.S.C. 101; 17 U.S.C. 102(a).

[5] *Sony Corp. of Am. v. Universal City Studios, Inc.*, 464 U.S. 417, 447 (1984).

The Open Source Definition—Appendix 1A

The Current Open Source Definition (OSD)

(VERSION 1.9)

Open source doesn't just mean access to the source code. The distribution terms of open-source software must comply with the following criteria:

1. Free Redistribution

The license shall not restrict any party from selling or giving away the software as a component of an aggregate software distribution containing programs from several different sources. The license shall not require a royalty or other fee for such sale.

2. Source Code

The program must include source code, and must allow distribution in source code as well as compiled form. Where some form of a product is not distributed with source code, there must be a well-publicized means of obtaining the source code for no more than a reasonable reproduction cost–preferably, downloading via the Internet without charge. The source code must be the preferred form in which a programmer would modify the program. Deliberately obfuscated source code is not allowed. Intermediate forms such as the output of a preprocessor or translator are not allowed.

3. Derived Works

The license must allow modifications and derived works, and must allow them to be distributed under the same terms as the license of the original software.

4. Integrity of The Author's Source Code

The license may restrict source-code from being distributed in modified form *only* if the license allows the distribution of "patch files" with the source code for the purpose of modifying the program at build time. The license must explicitly permit distribution of software built from modified source code. The license may require derived works to carry a different name or version number from the original software.

5. No Discrimination Against Persons or Groups

The license must not discriminate against any person or group of persons.

6. No Discrimination Against Fields of Endeavor

The license must not restrict anyone from making use of the program in a specific field of endeavor. For example, it may not restrict the program from being used in a business, or from being used for genetic research.

7. Distribution of License

The rights attached to the program must apply to all to whom the program is redistributed without the need for execution of an additional license by those parties.

8. License Must Not Be Specific to a Product

The rights attached to the program must not depend on the program's being part of a particular software distribution. If the program is extracted from that distribution and used or distributed within the terms of the program's license, all parties to whom the program is redistributed should have the same rights as those that are granted in conjunction with the original software distribution.

9. The License Must Not Restrict Other Software

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

10. The License must be technology-neutral

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

Note: *Article 10 is a recent addition to the OSD, and is intended to ensure that open source software licenses that require end-users or licensees to express an assent to terms through "click-wrap" mechanisms must allow for the possibility that (a) redistribution of the software will take place over non-Web channels that do not support click-wrapping of the download, and that (b) the covered code (or re-used portions of covered code) may run in a non-GUI environment that cannot support popup dialogues.*

2

Free Software and the GNU GPL

The free software/open source mantra: "reveal the code!"

Open your source code to the Internet's programming community and let them improve it.

Imagine life as a fantasy for the moment; wouldn't life be magnificent if it were like the file-sharing program called Napster? You could get your clothes from a place called Clapster by sharing clothing with others. You might be moved to describe life as if it were actually the cliché "what is yours is mine." Sure, the clothing from Clapster might not be perfectly suitable for your tastes, but what clothing is? And, of course, it would cost nothing.

Similarly, you could get your car from Crapster, where cars are shared. A kind of even exchange of cars—you take mine and I will take this other guy's car. Sure, the car wouldn't run exactly as if it had been purchased from the dealer, but borrowed cars aren't supposed to run like new. In fact, the car may not run at all, but that's ok: it's from Crapster so you can borrow another one. More seriously, despite what we may imagine could be a magnificent life of genuine sharing, when digital audio and video became the target of file-sharing on Napster, the music recording and motion picture industries did not think much of the idea. Yet within the open source community, sharing is a way of life, and it is fostered and supported in a manner that is actually better than what we might imagine life to be like in a world of Napsters.

This chapter focuses primarily on the nuts and bolts and practical realities of open source software development and licensing. While some have argued that a proliferation of software licenses in the open source and free software community will risk harm to the social norms supported and established by community members, it is difficult, if not impossible, to draft a single license template that covers all imagined development models. Moreover, lawyers must consider the individual needs of a given client and draft accordingly. Notably, rather unlike the software licenses written by lawyers who work for most proprietary software developers, open source licenses are primarily targeted toward individual end-users and small developers or programmers. A good license uses clear, straightforward language that is readable by the target audience. Developers who roll their own license must carefully consider their own objectives and draft accordingly.

Even so, license proliferation should be avoided. If the adoption of an existing, reliable open source license template suitably meets the needs of the licensor, then there may be no need to draft an entirely new, untested open source license. If a license drafter decides to use an existing open source or free software license as a template, the drafter should carefully edit a template when appropriate. The following material will provide a starting point toward making wise judgments regarding which license templates (and which license provisions) are most suitable for a given need.

Quite unlike the Napster controversy, downloading files from an open source project should never feel like stealing because no one is claiming rights to stop you from sharing. In the open source community, software is shared, not sold. The open source community emphasizes social conditions where openness is valued more than substantive exclusiveness. In this manner, the social norms valued in the open source community are parallel to those valued in some areas of academia, wherein researchers and scholars promote identity exclusiveness by distinguishing themselves by developing reputations as experts, but eschew substantive exclusiveness by freely contributing their academic work (or sharing their "code") through scholarly publications and conferences.

Academic works are shared so that the community of scholars may build on the work or rework the research to enhance its value or fix its flaws. Many scholars have built or enhanced their reputations by freely sharing their work product with others; the scholar's work is thereby identified and recognized by peers. Quite ironically, proprietary software development tends to undercut identity exclusiveness and load heavy importance on substantive exclusiveness, wherein the exclusiveness of intellectual property rights and secrecy of code creation are paramount. In this manner, the software authors gain little reputation-enhancing benefit from their programming because software code is often developed in secret and hidden from end-users, the employer builds its reputation by taking authorship and ownership of the copyright for itself, and disclosure agreements block open discussion about the development process.

Open source software authors contribute their work by sharing their source code as an open source or free software project initiated on a Web site designated as the project's main body of work. Once software authors download the source code from the Web site and begin extending the source code by improving it or adding features to the software program, an open source project is under way and improvements to the project's source code are associated with the respective programmers.

Often, word spreads about a program on electronic mail lists, word of mouth, exposure on a popular Web site, and the success of spreading news about an open source project largely depends on the popularity of a particular program. Popular open source projects may have hundreds of software developers and programmers participating in the project by downloading the software and uploading their modifications of the software. No one pays for the software because it is freely copied, distributed, and modified. These social norms are enforced by practice or custom and, most importantly, the terms of the open source software license.

The open source software development model seems to have borrowed a great deal from the academic community, but one should not overemphasize how these two groups overlap. Some commentators on open source development seem to oversell the importance of open source members obtaining reputation-enhancing benefits from their participation in open source projects. It is noteworthy that identity exclusiveness or reputation is more likely to directly motivate the work of academics than it is open source developers. Academics may be rewarded with tenure or media attention for enhancing their reputations, while open source developers frequently are motivated by goals closely associated with a business objective or a goal of improving the efficiency by which a task may be completed.

As suggested earlier, the open source community operates by its own energy and force; hence, the community has charged itself with the development of open source concepts that sometimes confuse end-users. The names GNU Public License, GNU GPL, and GPL refer to the same license—the General Public License. Occasionally, the abbreviated GPL is used as a reference to the concept of open source licenses in recognition of the fact that there are many open source licenses, of which the GNU Public License is, perhaps, the most well known. (In this case, GNU, or "Guh-New," is an acronym that means "GNU's Not Unix.") Notably, most critics of open source seem to focus their attack on the open source community on the GNU GPL. The GPL is probably one of the most controversial public licenses used by the open source community.

The open source community split with the free software community in 1998 over a series of disagreements regarding goals and ideology. Today, most popular media publications consistently refer to the open source community regardless of whether the points raised more appropriately apply to the free software faction of the community. To a large extent, the distinctions between the two groups are not pertinent, since the community, taken together, represents a coherent frontal assault on the values of proprietary software builders. Both groups have a common interest in opening up information products to more people than would be permissible under prevailing intellectual property standards; to wit, the groups have more shared values and goals than differences.

The open source community, generally, focuses upon increasing the participation of commercial software developers in actual projects, rather than upon clarifying ideological matters concerning the direction or ultimate goals of the movement. While the free software movement strives to bring commercial projects into the movement as well, its focus includes changing the ideological landscape of the software industry. Open source, on the other hand, reverses the method of the free software movement by encouraging participation without ideological commitment, with the somewhat idealistic hope, perhaps, that participation in the movement ultimately will make commercial developers more receptive to the movement's ideology.

There are factions within the community and voices that speak with unique positions or views of software development, but the objective of this book is to highlight and illuminate basic, uniform precepts of open source software licensing, not to take positions on esoteric or theoretical distinctions on the margins of open source philosophy. Consequently, this book will not traverse the nuanced arguments that delineate further distinctions between the factions within the free software/open source community. The success of open source is well established in cyberspace. Indeed, its current success should signify that open source is a viable model of electronic commerce that could extend far beyond the software industry. In this respect, it is particularly appropriate to forecast a technological vision in which an open source philosophy will provide the core principles for examining the keystone of a dynamic shift away from conventional business models and toward open, nonproprietary business models over the next decade in electronic commerce.

At bottom, open source has one major guideline: if you share your source code with the world's open source programming community and let everyone try their hands at improving it, the benefits are astounding for the end-user community as well as for other software authors. "Source code is like manure. Spread it around and things grow. If you horde it, it just smells bad" [1]. This principle identifies a global community that is successfully challenging the contemporary proprietary model of software development with a model in which openness is considered a virtue rather than a blunder.

Licensing To Meet Philosophical Objectives

Open source programming eschews the use of software development to strategically control markets without regard to the production of superior software applications. Instead, the open source model produces a superior product from the software programming assistance of potentially hundreds or thousands of programmers. In this manner, open source reinforces market competition by precluding a lock-in to a proprietary technology controlled and dominated by a single source. It is not surprising that some proprietary software builders view open source as a critical threat to their ability to sustain software development models that exalt secrecy over openness, control over freedom, and property rights over the public domain; those models dominated the vitally important software development cycle during the 1990s.

To thwart the growth of open source, some critics contend that the philosophical objectives of open source should be viewed with suspicion and even contempt. Not surprisingly, a few critics opine that open source ideas could undermine the very foundation of the capitalist ideal if open source spreads into areas outside of software development. Some critics have argued that open source software development is premised on a conceptually flawed philosophy and, hence, does not constitute a viable alternative framework to the prevailing legal regime governing the intellectual property protection of software. Some of the most thoughtful arguments raised against open source seem to be framed around the view that open source software development raises the economic risks associated with software creation and carelessly threatens the economic health of many information technology companies by increasing the likelihood that established proprietary software developers would be hurt by a legal regime that forced them to open their source code, divulge their trade secrets, and remove copy-protection code from the open source software.

Following upon that argument is the position that quite to the contrary of the urging from proponents of open source, proprietary software developers like Microsoft and Oracle—sometimes called cathedrals of software development—produce software that has all of the pertinent advantages of open source software development without any of the hazards. Some have taken the argument further by arguing that cathedrals like Microsoft or Oracle can ostensibly become "bazaars" behind the closed walls of the cathedral and provide the same type of software development model. Notwithstanding that some of these arguments raise plausible positions and are delightfully sentimental—especially in the manner in which the arguments attempt to resuscitate the eroding proprietary software development model—they are not moored in the essence of the philosophy of open source.

Even so, the link between open source software licensing and traditional copyright law should not be understated; perhaps the real ingenuity of the use of open source software licensing is how the licenses are used to both reinforce the prevailing legal regime of copyright and to undermine it. Still, open source best demonstrates that the prevailing assumptions about the relationship between copyright and software are no longer well-founded, and the time may have come to remove the legal lumber—its root and branches—that supports the overprotection of software by copyright. The Internet itself provides abundant proof that a wide-ranging network of programmers can develop robust, popular, and reliable software by ensuring that the development of software is customarily accompanied by the distribution of or access to source code in a manner that is open and freely available to the public.

As the world we live in becomes more technocentric, all technology end-users have a vested interest in controlling technology in a more fundamental way than closed code and closed systems allow. Open code empowers end-users to understand and control technology rather than be controlled by technology or the technology makers. Should end-users become programmers? No. Still, there is little basis for doubting the fact that end-users in a technocentric world must become techno-savvy.

Indeed, it might be that open source *is* radical. And those who would rather make a grand marketing blitz each year announcing that the cathedral is about to offer the masses a new version of This95 or a new version of That2000, rather than produce superior products that are supported and wanted by computer users, may be justified in their assertions that open source is a threat to their business. Open source, of course, is not just about technology; it is about free books, free art, and free information (as we are often told, free, as in freedom).

In the context of software development, however, open source is a business model. Its philosophical precepts are an outgrowth of a document radical in its inception, but fundamental in the values it supports, namely, the United States Constitution. In Article I, section 8, clause 8 of the Constitution, the founding fathers provided the U.S. Congress with the power to grant exclusive rights to authors and inventors for the purpose of promoting the progress of science and art. For open source proponents, copyright begins at the constitutional grant, because the purpose of copyright is more evident in the clear words of the constitution than it is in the rather long-winded Copyright Act. In this regard, open source reengages the debate concerning what grants are necessary to promote the progress of computer science.

Copyright and Copyleft

Conceptually, *copyleft* is a legal construct—developed by the FSF—that grants licensees or end-users rights of reuse, modification, and reproduction of a work or its derivatives as long as those same rights are passed onto others when the work is reproduced or redistributed. In other words, in exchange for the nonexclusive right of redistributing the open source software, licensees must attach the same license terms to their distribution as that found in the GNU GPL, including the terms that grant end-users the nonexclusive right to modify, copy, and further redistribute the work with access to the source code. This is a rather unusual and perhaps novel [2] approach to software licensing.

The approach turns copyleft into both a controversial and an extraordinarily significant software license term. Copyleft is significant because it is a powerful tool that enables an open source project to grow in the number of end-users who assist in developing the software while also minimizing the risk that rogue programmers or opportunistic end-users will copy the source code of a successful open source project, close up the source code, and claim it as their own; copyleft renders that conduct unlawful under the terms of the software license. In this manner, copyleft provides significant assurance to open source developers and end-users that an open source project will remain an open source project.

The primary risk that an open source project protected by a copyleft provision in the general public license could deplete the volunteer efforts of end-users and programmers who assist in developing the software stems from the project owner's copyright, not from an outsider attempting to gain an advantage from the fact that the source code is freely available. Like any copyright holder, the project leader of an open source project may exercise an exclusive right to control the work by issuing a new version of a license to cover a preexisting open source product. If the new version of the license were a proprietary software license, rather than an open source or free software license, then the project owner would inflict a serious blow to the goals of open source. Even so, the adverse effects of altering a software development project from open source to closed source or publicly redistributing the open source software under the alternative terms of a proprietary software license may be minimized by a technique called defensive *code forking*. Code forking occurs when the codebase of an original open source project splits into two or multiple projects that may begin to compete with each other. Similarly, abusive code forking occurs when an open source project's codebase is split off by the original copyright holder or licensor as a separate proprietary software application wherein the right-holder benefits from the free labor of others by selling the proprietary version. To date, only one project, the X-windows project, was altered in this manner *after* receiving the benefit of open source labor. In that instance, the project owner was convinced to withdraw the new software license by a few vociferous members of the open source community. It cannot be doubted that aside from the legal claims that might arise from such a circumstance, a project owner is likely to commit this selfish atrocity only once.

Other defenses against potentially abusive practices by open source project leaders are built into the open source model. For example, no end-user would contribute effort to any future project with an abusive project owner. More importantly, there is, at least, the theoretical possibility that as more contribute to an open source project's original codebase, the project owner's ability to unilaterally code fork the project without infringing the copyrights of the contributors diminishes significantly. In actual practice, contributors have reason to cede control over the project to the project owner for defensive code forking purposes and for the sake of the open source project, which seems to have a greater chance of enormous success if the project is managed well, although not necessarily authoritatively.

An interesting example of this is the GNU/Linux kernel, managed by its founder, Linus Torvalds, which is a project managed remarkably well by Torvalds. GNU/Linux is an extremely successful open source project. Despite the fact that the project owner's contribution to the current codebase might be considered sufficiently diminished as a result of the significant contributions of others to have weakened his control over the project, Torvalds still leads the project and is viewed as important to its continued success.

On the other hand, copyleft provisions are controversial for three primary reasons: the provision appears in licenses that lack privity of contract, the provision has a viral tendency, and the provision favors a business and software development model that restricts the end-user's ability to code fork and/or incorporate proprietary source code in a software application.

Some legal commentators question whether a licensing model that appears to lack privity of contract is enforceable against end-users who have no direct successive relationship with the copyright holder of the open source project. Section 6 of the GNU GPL overcomes the privity-of-contract issue, to the extent that privity is still relevant in contract law, by providing that the original licensor, regardless of who actually provides the software to the end-user, licenses each distribution of an open source software program.

Even so, with one notable exception, there is little reason to challenge an open source license on grounds of lack of privity of contract, since the open source license grants freedoms to end-users that, absent the license, would violate copyright if the end-users are shown to have engaged in those activities. In the one instance where end-users might successfully argue that they were not bound by the terms of the license because of a lack of privity of contract, they would also likely raise a fair use defense that could prevail if it were based on the end-users' creation of a transforming derivative work.

This claim may be one of the most fundamentally sound criticisms of copyleft, although the argument tends to be more theoretical than an actually persistent phenomenon. Certainly, a popular and important open source project protected by copyleft could engulf a great deal of source code and, in effect, appear to resemble an anticompetitive business practice accomplished through licensing conditions. Copyleft has two viral qualities: the potential mixing of free or open code with nonfree or proprietary code infects the nonfree code by bringing it within copyleft if the code is publicly distributed and copyleft devours a copyright interest owned by the author of a transformative derivative work. For example, copyleft tends to swallow up all types of derivative works, including those that may be so transformative that they come within the Supreme Court's conception of fair use.

Under copyright law, users are allowed to create derivative works that are so distinct from the preexisting work that they are held to be transformative. Copyleft appears to deny these users the opportunity to distribute a transformative work under terms entirely distinct from the open source license. This ironic twist of fate of copyleft, however, is highly unlikely under current software development cycles and the life span of typical software applications. More importantly, the recognition that source code can be used to manipulate or enhance market power in an important software product category may illustrate how copyright law has failed to adequately calibrate the proper scope of copyright protection for computer source code, rather than indicating a potential area of abuse of software licensing practices.

Code Forking

Within the open source community, adherents of the FSF disagree with those of the OSI concerning whether code forking of open source projects should be widely permitted. To a significant extent, the acceptability of code-forking stems from the boldness of the copyleft concept. Copyleft precludes the distribution of open source software with proprietary source code and it halts the conversion of free software to nonfree software.

In some respects, whether copyleft provisions are lawfully enforceable or may effectively prevent pernicious code forking is still an unanswered legal question. An ironic impact arising from the use of a copyleft provision in a general public license is that it seems to have a tendency to weaken an author's freedom to develop software almost as much as it may enhance some of those freedoms.

More importantly, copyleft may occasionally have the same adverse impact upon the public domain as does proprietary software licensing; namely, both copyleft-protected open source software and copyright-protected proprietary software tends to keep more source code out of the public domain during its useful life span than would otherwise be the case. For example, copyleft has a viral node that is activated when open source software protected by copyleft is "mixed" with non–open source software.

Since the GNU GPL requires the redistribution of the software under copyleft, the open source software distresses the non–open source software by imposing the conditions of the GNU GPL. In this instance, proprietary code becomes open source software—an outcome many would not find displeasing—but in other circumstances it may be more apparent that the open source project has the undesirable capability to implacably take in all that it touches.

This result is likely to be distressing to those who promote the growth of works in the public domain, where none of the limitations of copyright are found. (An open source software license is a copyright license; hence, open source software, although freely distributed under the terms of a license, is not in the public domain.)

Even so, as mentioned earlier, copyleft provisions help ensure that open source software projects remain open source projects. Most adherents of open source are likely to value the assurances that derive from copyleft, but many also have pragmatic aims to promote open source within the software industry by allowing proprietary-software developers or any end-user the freedom to distribute software developed as a derivative of an open source project without the restraint of copyleft.

In other words, many OSI adherents, unlike some FSF adherents, favor the use of open source licenses that do not restrict the end-user's choice to prepare a derivative work that ultimately is distributed as a nonfree, closed, proprietary software application. In this manner, code forking is not viewed as entirely undesirable conduct. Instead, permissible code forking is considered as a cost of ensuring that end-users have the freedom to choose a preferred software development model as well as a practice that generally benefits the open source community by bringing prominent software developers into the open source community, including those who might fail to develop or lead open source projects if their source code had to be within the reach of copyleft.

What Is Free Software?

When referring to software as *free software*, *free* is not a reference to the cost of software. Instead, it refers to the freedom to change the source code and redistribute it as is or as a derivative software program. Since it is entirely permissible to charge a price for the cost of software distribution, to many it is confusing to use the label free software community to describe software that may actually cost something to acquire it. Richard Stallman attempts to clarify the confusion by referring to a conceptual distinction summarized as free software means free speech, not free beer.

No doubt those who are not entirely comfortable with the free-speech justification for free software do not find it compelling that a reason for supporting free expression seems to hinge upon the same odd notion that source code is free speech. Instead, free software more plausibly refers to equitable notions of when or how aspects of technologies (e.g., source code or exclusive rights) should be shared in fulfillment of an ultimate goal to produce greater access to works. In other words, I would urge that the *free* in free software actually fulfills an ultimate purpose of copyright and does so through rewriting what should be considered fair use when copyright applies to a technological work.

The most famous free software project is probably Linux (or GNU/Linux), a version of the UNIX operating system. Its proponents distributed a version of the code on the Internet several years ago, and hundreds of programmers have added their own refinements. The result is claimed to be a much faster, less crash-prone operating system than any of the Microsoft Windows operating systems.

Interestingly enough, the specifications for the software protocol that controls the flow of information for the World Wide Web, Hypertext Transfer Protocol (HTTP), and the specifications that allow Web site authors to create Web sites, Hypertext Markup Language (HTML), were developed as open source technologies by the father of the World Wide Web, Tim Burners-Lee. On January 22, 1998, Netscape Communications Corporation made the source code for its popular Web browser software available for free licensing on the Internet. Then, as explained more fully in Chapter 3, Netscape joined the open source community in an attempt to harness the creative power of thousands of programmers on the Internet.

The open source community has been a successful way to stimulate the creative energies of the cyberspace-based programming community. It has inspired unprecedented levels of innovation in software development. Netscape manages its open source project by using a distribution license called the Mozilla Public License (MPL), which allows source code modification and redistribution and provides for free availability of source code versions, but has no copyleft provision. Hence, Netscape represents one of the variations within the open source community.

Open source is in some limited respects similar to the freeware tradition of distributing software on the Internet. Freeware describes software that is distributed to use it at no cost while the software application is under development. In other words, users were granted free use of a software program in exchange for comments about whether the software performed according to expectations. Once a program had been sufficiently improved, many programmers would abandon earlier versions of the software and begin selling the more refined product. In this respect, open source significantly extends the freeware development process far beyond cheap labor and smart marketing by allowing others to freely develop new programs using original source code.

What Is Free About Free Software?

Although the appropriate scope for the copyright protection of computer software has been subject to extensive commentary in scholarly and popular media, the boundaries between what is within the scope of copyright protection for software remains fuzzy and subject to semantics; this is particularly true in the context of copyright infringement actions involving computer source code.

Even so, the global importance of the Internet and the increasing vitality of the software industry warrant examination of the question: whether there really is a principled basis for delineating distinctions between copyrightable and uncopyrightable expression in software. Today, overhanging the answer to that question is often the refrain that software is speech, or source code constitutes free expression. Indeed, the open source and free software community promotes free access to source code on the basis of accepting the principle that what should be free about software is not its cost, but its expression.

In the borderless virtual space of cyberspace, the shift from mere idea to the communication of an idea occurs automatically almost as a transparent instinctive response. Yet the conceptual distinction between ideas and the communication or expression of ideas is fundamental in copyright doctrine. Since the proverbial moment the law of copyright first recognized that computer programs could be subject to copyright protection, courts have struggled with setting the boundaries of what aspects of a computer program are copyrightable. Clearly, both the Copyright Act and the First Amendment prohibit the application of copyright protection to ideas, but applying that constitutional and statutory doctrine to actual allegations of copyright infringement is neither simple, nor precise. It is a well-established principle of fair use of copyright-protected computer programs that reverse engineering of those programs to obtain access to the underlying ideas embedded within the software's source code is a fair use of the software.

The United States Supreme Court has determined that First Amendment freedoms of speech include the collective interest in protecting an individual's right to freely express almost any idea known to man. Copyright law directly affects the free expression of ideas because the U.S. Constitution secures for "limited times" to copyright holders "the exclusive Right to their respective Writings and Discoveries." The copyright statute gives copyright owners a variety of exclusive rights: the rights to make copies of their works, to create derivative works, to distribute the works, and to publicly perform or display them. In other words, the law of copyright grants to authors the right to control, restrict, or thwart public access to their expressive product. Copyright law protects original authorship, but sets up a dichotomy between the protected work and the idea within the work. Any original work of authorship that exists in tangible form is copyrightable. Copyright protection, however, extends only to the particular expression of the ideas contained within the work, not to the ideas themselves.

Despite the compelling language of the Constitution's copyright provision, it is apparent that the founding fathers may have only intended to permit Congress to protect copyright holders' rights to their original expression. In the clash of competing constitutional provisions and almost strictly as a conceptual matter, the First Amendment trumps Article I, Section 8, Clause 8 in its significant limitation upon the scope and function of the law of copyright. Although there is historical support that the early eighteenth-century European system of copyright was a tool of censorship that was primarily used to restrict the flow of knowledge and information in a manner that benefited those viewed favorably by the state, the founding fathers took a different view of copyright by providing for Congress's power to authorize copyright grants in Article I of the U.S. Constitution. Notably, rather than adopt the view of their European predecessors, the language of the Constitution supports the judgment that the founding fathers must have viewed copyright as an important constituent element of democracy and civilized culture.

The paradox is that the public can only benefit if it has access to a work. Access is restricted, at least for a limited time, by granting authors property rights to their work, for only by restricting access can authors charge users and earn a profit. The artifacts of cyberspace are largely intellectual property, and the owners of the intellectual property have a right to control how their property is communicated. In this respect, despite the open and public nature of cyberspace, it is the province of an inherent and basic tension flowing from the goal of access to information between those that communicate and those that own the artifacts of communication. Of course, reliance on copyright law is not the only way authors may reliably restrict access to their work. They may, for instance, license use of their computer software. Or they may rely upon technological barriers—often referred to as *digital copyright management systems*—to prevent unfettered access to their work. Notably, some authors use a variety of factors to restrict public access to a given work.

Although the precise question has not been addressed, for the few courts that have addressed what is free about source code, most have conflated the inquiry into a single inquest rather than acknowledge that two questions are before the court, namely, whether source code is speech and what level of First Amendment constitutional protection should source code be afforded if it is speech. There is a good reason why a court might conflate these distinct questions; namely, courts are no better suited than anyone else to answer the former, while it is their constitutional province to answer the latter. In the normative sense, to answer the question of whether source code is speech is to determine what *is* speech, which is no simple matter. (It is true of many areas of the law that the route toward a unifying principle is aligned with dead ends and circular trails; apparently, the jurisprudence of the First Amendment's conception of freedom of expression is no exception. Even so, abstract notions deserve rigorous coherent scrutiny.)

In this regard, to say that source code is speech is merely to say that there is a connection between the goal of computing or using machines and the appearance of a natural language. Most courts addressing the question of whether source code is speech seem to quite correctly recognize the duality of the essence of source code: "it says what is does." Even so, source code exists to run machines more efficiently because of the vast advantages that cross-platform use of software has over hardware. The ability to use software to efficiently perform tasks once solely performed by expensive, inefficient machines, along with the industry's recognition of the benefits of standardization, forms the primary basis for the extraordinary growth of the software industry. Hence, the inherent connection between source code and machines cannot be discounted simply because source code serves an important secondary purpose. On the other hand, what precisely is free about free software seems properly rooted within the doctrinal concept of *fair use,* rather than directly tied to the values associated with freedom of expression. Fair use is a copyright doctrine that invokes an equitable rule of reason. Although there are those who would favor a strict adherence to the statutory fact-based inquiry now viewed as the fair use test under copyright law, I think the technological form of software calls forth the application of a fair use doctrine with substantial vitality.

Source code *cannot* be speech simply because it is potentially or often mediated by human interaction—programmers communicate to each other through source code. The success of speech recognition and character recognition software will soon enable programmers and anyone else to instruct machines directly through a type of verbal source code or list of commands closely approximating natural language. The problem is, of course, that verbalization, like the source code that precedes it, is not human speech when it is not mediated by human interaction.

Although there are a number of programs that constitute examples of open source projects, some of which will be discussed in this chapter, there is considerable debate as to which programs are in substantial compliance with the terms of the GNU GPL, which could be described as the constitution of open source. The greater a program's public license departs from the terms of the GNU GPL, the more likely that the program's license will restrict rather than broaden the freedoms associated with open source code distribution and copyleft.

Software licensed under the GNU GPL or a license adopting the GPL as its template is, of course, copyrighted software. Hence, the conditions provided for under section 2 of the GPL are imposed by the licensor or copyright holder, and for some purposes, it makes a difference who that is. Since the prevailing view of copyright classifies copyright holders as *owners* of copyright, copyright holders are also usually the owner of the work embodying the copyright interest as well.

Even so, in a GNU project, free software adherents primarily see themselves not as owners of something one can use at their pleasure, but, ethically, as custodians of the work on behalf of the public. In this manner, genuine free software occupies a space that closely approximates the public domain—perhaps even more so than typical open source projects that are not governed by licenses covered by the GNU GPL or based upon the GPL's template. Of course, the practical distinction between the view that the public should be treated *as if* the public owned your project and the public freely acting in a manner supported by lawful "public ownership" in some instances could be vast and one should make no mistake with regard to open source software and the public domain, the public is not legally the owner of an open or free software project. Tim Berners-Lee, the author of the World Wide Web, makes note of this distinction in his book, *Weaving the Web*, when he notes that the fact that the GPL is a copyright license, regardless of its otherwise appealing framework, led him to dedicating his Web protocol to the public domain after initially considering using the GPL.

Generally, the terms of an open source license do not have a pertinent application to the licensee until or unless the licensee seeks to publicly distribute the original work or a modified version of the original work (e.g., a derivate work). In redistributing the work, the licensee must comply with the terms of the software license. The distribution term of critical importance under the GPL is the copyleft provision under section 2, which requires the same license to be used in all subsequent redistributions, and the source code provision under section 3, which provides that the licensee is granted the right to redistribute the work publicly if the licensee provides access to the source code.

The nonexclusive rights that open source provides end-users, resellers, and third-party software developers are maintained by the use of an open source public license; most notably, as was mentioned before, that license is frequently the GNU GPL. Although it is unclear whether the GNU GPL would be considered a type of shrink-wrap, click-wrap (or click-through), or browser-wrap license, when used in its most common formulation, which is as an electronic license appearing on a Web site, the GNU GPL binds end-users who consent to its provisions by downloading an open source program. The GNU GPL license achieves three goals: it designates ownership of copyright for the given open source project; it grants consenting end-users the nonexclusive right to modify, copy, and redistribute the source code in the derivative or copied program; and it sets other distribution terms affecting the rights of subsequent end-users, such as imposing a copyleft requirement.

In 1999, the FSF renamed the GNU Library GPL by substituting *Lesser* for *Library* to discourage open source and free software developers from unnecessarily using the license when the GNU GPL would better serve developer's purposes. The GNU Lesser GPL (LGPL) is a license that is less restrictive than the GNU GPL in that it allows the public distribution of open source or free software programs with software or libraries designed to be linked to other programs dynamically or at run-time.

The FSF defines *libraries* as "specially designated software packages." These software packages, whether free or not, are often modules of source code that are intended for use by third-party application developers. In this manner, developers save time from writing the same code each time the author wants to call a certain object. Instead, the author may write code that calls the library dynamically (only during run-time or upon program execution by the user) or statically (at compile time).

To bolster the popularity of the GNU/Linux kernel, open source and free software developers adopted the practice of using the GNU LGPL for libraries so that proprietary software developers would have access to the libraries and would be encouraged to develop programs that relied upon the library without fear that their proprietary software would have to be distributed under the terms of an open source license. The terms of the GNU GPL, if applied to a library, precluded proprietary developers from having access to the library for the development of proprietary software. Whether GNU GPL actually disallows dynamic linking with a shared library as well as static linking is greatly disputed. The FSF seems to say yes and directs users to the GNU LGPL, but others point out that the GNU GPL was drafted before dynamic linking became routine and, therefore, it is doubtful that it is directed toward dynamic linking. Indeed, according to the FSF, the programming paradigm in use is not the critical factor in determining what rules of compliance apply to the license. Regardless of whether the creation of a derivative work in this context involves the straightforward modification of a work licensed under the GNU GPL or whether the issue concerns the more nuanced issue involving static or dynamic linking, if the resulting combined program or derivative work must be combined with the source code governed by the GPL in order to run, section 2(b) of the GPL controls, wherein the new combined work must be distributed under the terms of the GPL. If the LGPL, rather than the GPL, were used to distribute the original work, then the combined or derivative work could be distributed under any license. The GNU LGPL was updated and revised in 1999, and the GNU GPL is currently under review by the FSF. Consequently, it is likely that the next version of the GNU GPL will clarify its coverage on linking.

Even so, some view the GNU GPL and the GNU LGPL as unclear on whether the GNU GPL actually precludes all types of mixing of free software libraries with nonfree software libraries. In particular, there does seem to be genuine confusion within the open source community as to whether FSF really intended the GNU GPL to preclude dynamic linking and, if so, what impact that may have for open source developers who have adopted the GNU GPL to distribute their own programs, but have mistakenly interpreted the GNU GPL only to preclude static linking.

In applying section 117(a)(1), whether dynamic linking to a shared library somehow creates a derivative work is a question of copyright law, not contract. Since, in this regard, the GNU GPL is intended to apply to derivative works, the GNU GPL may not apply at all in most cases of dynamic linking of shared libraries. It is highly doubtful that the dynamic linking by a proprietary application to a GPL'ed library would normally result in a derivative work of the original licensor.

The linking would be permissible under section 117 as a permissible copy of the library, and the dynamic linking would not come within the GNU GPL, unless it could be shown that the linking modified the library in a manner that amounts to the creation of a statutory derivative work. Although this issue is discussed frequently within the open source community, the fear of impermissible dynamic linking is more conspicuous than the reality of creating undesirable derivative works.

One answer, which only partially sidesteps the question presented, might be that in this regard the GNU GPL should be read to be consistent with copyright law, since the goals of copyright law and the open source community, in this regard, overlap. Section 117(a) of the Copyright Act [3] limits the right of the copyright holder with regard to computer programs, and is relevant to the dynamic linking of shared libraries. Given the general purpose of the GNU GPL, we can make two assumptions about its provisions: (1) it does not restrict a right of the user that is provided for by the Copyright Act under section 117(a)(1), since to do so would undercut public access to works, and (2) to be faithful to the open source distinctions between the GNU GPL and the GNU LGPL, care must be undertaken to note that this issue usually arises under circumstances where the licensor is the author of the shared library. In that regard, the author's intent is for proprietary applications to be permitted to link to the library. Hence, the author uses the GNU LGPL.

Only where a library implements an important capability that is not generally available to proprietary software developers are open source developers encouraged to license their library under the terms of the GNU LGPL. The FSF, in particular, encourages this narrowly tailored approach to using the GNU LGPL because of perceived strategic benefits, such as the range of useful modules to serve as building blocks for new open source or free software that would arise primarily from the stockpiling of GPL-covered libraries that have no parallel available to proprietary software.

Section 7 of the GNU LGPL authorizes the distribution of open source or free software code along with nonfree software code. In the pertinent part, section 7 provides:

> 7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

We use this license for certain libraries in order to permit linking those libraries into nonfree programs. When a program is linked with a library, statically or dynamically, the combined derivative work distributed under the GNU GPL must be a free or open source program. On the other hand, the combined work may be distributed under the GNU LGPL, where a free or open source program is linked to code in a nonfree or proprietary program.

The initial clause of section 5 appears to exclude from the scope of its provisions any work that "uses the Library," but that is not, itself, a derivative of the library. If, however, a derivative work is at issue, then section 6 sets forth conditions that, if complied with, would permit distribution "under terms of your choice," as long as those terms include a provision permitting modification of the licensor's work by the licensee for the licensee's "own use" including reverse engineering for debugging.

Section 7 [4] of the LGPL introduces a level of confusion regarding the application of the term *derivative work* to software since the LGPL appears to use the terms "derivative work" and "modification" synonymously, despite the fact that the Copyright Act uses exclusive meanings for these terms. As a practical matter, a licensee who develops a work based upon some type of modification or adaptation of an open source work is concerned with whether the distribution of the modification must comply with the open source license. Many critics of open source licensing have argued that the derivative work issue, coupled with the GPL's copyleft provision, has an undesirable viral effect; namely, the reach of the GPL is so broad that its license tends to spill over and control the development of any work that touches a work originally licensed under the GPL.

To blunt the sting of those arguments, the FSF created the LGPL. The viral effect of the GPL was particularly troublesome with regard to the use of libraries (objects) or files containing source code that can be called by external programs without the need to rewrite the code for the function or class contained in the library. The LGPL is not favored by the FSF because it undoes a portion of what the copyleft provision protects by allowing free software to mix with (or link to) software or code that is not free. Still, the most viable threat to the survival of open source and free software is the concept of *trusted computing*, Trusted computers are computers that can be trusted by media companies not to run software that users can modify, so that the content can be delivered without fear that software modified by users will exercise fair use rights that media companies do not want to allow.

As open source grows more competitive, the need for the LGPL should diminish considerably, but today it serves a critical function in encouraging commercial software developers to participate in free and open source software development. Richard Stallman explained the effect of section 7 of the LGPL in the following manner:

> If an application 'A' uses a library 'B' in what might be described as an essential way, then, irrespective of the physical mechanism of linkage (static/dynamic/run-time/compile-time/CORBA) [5], I would expect 'A' to be considered as a derived work of 'A'. Especially if 'A' is distributed together with 'B', and especially if 'A' won't function without 'B'. In other words, as far as the drafters of the GPL and the LGPL are concerned, it is not likely that a software developer intending to use some aspect of open source software will also be able to escape the reach of typical open source software licenses by the use of clever programming techniques. Instead, the developer is advised to use the license providing the greatest degree of freedom from the viral effects of some open source licenses, and in most cases, the LGPL offers that type of freedom.
>
> So, for Sun's Java program, for example, if you GPL your java library, and a commercial company distributes a java program using it, then I would expect the GPL to apply—even though the technicalities of linking differ from the C case. … So I think it is meaningful to release a Java program either under the GPL or the LGPL, and the consequences are basically the same as for a C program: if you use the LGPL for your Java program, it can be called by non-free programs, but if you use the GPL, it cannot be.

With specific regard to CORBA, the use of the GPL should not cause a viral effect if it is the license covering the original work. The circumstances are practically identical to those regarding many Java programs; it is often common practice with Java to have programs link together dynamically. Even so, the FSF has concluded that although it is likely that the GPL would be disruptive under Java or CORBA when applied to most linking questions, no categorical conclusion can be dispositive of the issue. In the view of FSF, "any definitive criterion based solely on the mechanism of communication is likely to give silly and arbitrary results." Hence, complex linking questions may have to be resolved on a case-by-case basis until a court establishes a sound legal principle that sensibly applies to the technical concern.

Although it is clear that the copyleft restriction in the GPL exists as a condition for using free software in order to prevent a licensee from attempting malicious code forking or from successfully closing up free software and converting it to proprietary format, it is not as apparent how this goal is achieved in the context of dynamic linking to objects, since the copyleft restriction in that context seems to restrict nothing more than a standard activity of software authors. In this respect, critics of copyleft have complained that the GPL not only may serve the ends of the free software movement—since the application of copyleft ostensibly cuts off undesirable free-riding by proprietary software builders—but may serve such ends to the detriment of those that join the movement by restricting the free choice of programmers to use efficient programming techniques through linking to the available objects in an operating system or platform.

The LGPL, like most open source software licenses, contains an "as is" warranty disclaimer. "As is" licensing disclaimers are authorized under the Uniform Computer Information Transactions Act (UCITA). In addition, the federal warranty law (Magnuson-Moss) only governs a written warranty for consumer, mass-market goods, which, arguably, may include software distribution when the seller provides a written warranty. Generally, if a written warranty is provided, the seller cannot eliminate any implied warranty under the federal law. You might conclude that "as is" licensing is not exactly consumer-friendly, but one might also view it as part of the trade-off for the freedom granted by the licensor.

The litigation against America Online (AOL) regarding software bugs in the on-line service's software is a notable example of the use of disclaimers, and may have relevance to the "as is" disclaimers often used in open source licenses. There are two interrelated cases involving AOL in the litigation: AOL's users are suing the company in Florida and AOL is suing its insurer in Virginia. [See *In re America Online Inc., Version 5.0 Software Litigation, No. 00-1341-MD-GOLD* (S.D.FL.)]. AOL users sued AOL claiming that the on-line service's software version 5.0 caused users' computers to crash.

For whatever reason, AOL's insurer refused to pay any liability on successful claims because software-based computer crashes do not constitute tangible property damage, which is the only type of damages the insurer agreed to cover. (This must have shocked AOL, of course.) The Virginia court reportedly agreed and opined that where software causes a computer to crash, the crash, the loss of computer use, and the accompanying destruction of data are intangible losses. Denominating the losses as "intangible" rather than "tangible" is significant in states that follow the common law rule. Is this a bad decision? At first blush, it might seem odd to urge on the one hand that source code is a cauldron of ideas often hacked together by programmers in a manner that strips the programming of originality, while urging, on the other hand, that courts are misguided when they conclude that source code is speech. Of course, the point, precisely stated, is that in a given context, the use of source code may serve a communicative basis, but that determination neither requires that courts bar consumer claims of economic loss related to computer use (i.e., under tort theories); nor, hold that in pertinent states, the claimant may recover only for losses based on contract theory. Instead, courts need to develop a rigorous analysis of software use that contrasts software speech from software functionality.

The case in Florida is still pending, but the Virginia court opined that the consumers' claims are barred because only intangible losses were alleged (i.e., no contract theory claims). Although there is reason to view the Virginia court's decision with some caution, it does provide some perspective on this disclaimer issue. First, some states may cabin software off in an intangible loss category. at least under circumstances similar to software-caused computer crashes. Second, in these states, claimants will not be successful in tort-based claims, and contract-based claims will be subject to contract terms, which would include the disclaimer provision in the license. This looks like a particularly good development for open source developers whose disclaimers are appropriately drafted.

As noted already, open source software licensing is not a no-risk proposition; not only is the business model novel, but the very basis of the software development model is rooted in a higher-than-usual degree of risk of litigation over intellectual property rights. Not long ago, for example, the open source movement was abuzz in gossip about imminent breaking news that a proprietary software vendor, Santa Cruz Operation (SCO), a proprietary software distributor based in Utah, was about to join the open source movement by distributing Linux software. SCO's defection from proprietary software development, like Netscape's, was viewed as a considerable accomplishment for open source. As anticipated, SCO joined the open source community in March of 2000 and announced that Linux would be a supported server platform for its flagship application-hosting system. SCO also announced that the company's professional services division would be providing services supporting the deployment of Linux. The enchantment over SCO's defection evaporated three short years later, in the spring of 2003, when SCO and other proprietary software builders launched the most aggressive campaign to date *against* open source software development; namely, they sued. SCO's claim struck at the heart of the open source community by claiming that Linux—a highly successfully open source product—contained stolen source code.

Prior to the lawsuit, Linux had been achieving increasing success in eroding proprietary software market share in the enterprise operating system software space. SCO's direct strike at the open source community included a claim that the open source development model is premised upon theft of valuable intellectual property in the form of source code. In effect, SCO claimed ownership of copyright and trademark interests in a version of an operating system called UNIX, for which it claims the Linux operating software system infringes copyright.

For a notably short time period, SCO operated genuinely as an open source software distributor called Caldera, Inc., which was founded in 1994. In

1998, Caldera Systems, Inc., was created to develop Linux-based business solutions. In 2001, Caldera acquired the assets of SCO, and shortly thereafter adopted the name: The SCO Group. SCO filed suit against the deep pockets of IBM, one of the most visible supporters of open source development and a major competitor of Microsoft.

What is the real point of the SCO lawsuit? This may not ever be conclusively established, but some commentators have asserted that SCO's lawsuit is a thinly veiled attempt to raise shareholder value of SCO or to pressure IBM into an acquisition. If true, SCO's strategy is not only irresponsible and beset with incredible risk, but is also illustrative of the highly cynical attitude toward software development that has overtaken some well-known proprietary software vendors: in effect, these companies seem willing to risk disrupting the future prospects of an entire industry on the basis of self-interest and a rather dubious and parochial copyright claim. Commentators ascribe ill-will to SCO's motives for a number of reasons, including two noteworthy grounds: (1) it appears highly unlikely that SCO distributed Linux software—with open access to its source code—for at least three years before discovering UNIX in Linux mixed within the source code, and (2) Novell Corporation has maintained an unrebutted open and notorious claim that it owns the intellectual property rights to the original UNIX operating system software for which it has not sought enforcement against Linux. Still, SCO claims it is the owner of the UNIX operating system and that its intellectual property rights date back to 1969, when the UNIX system was created at Bell Laboratories.

SCO's suit against IBM seeks more than $1 billion in damages, and alleges that IBM made concentrated efforts to improperly destroy the economic value of UNIX, particularly UNIX on Intel, to benefit IBM's new Linux services business. According to SCO, IBM deliberately undermined the market value of UNIX as an enterprise operating system despite IBM being privy to SCO trade secrets. SCO is alleging misappropriation of trade secrets, unfair competition, and breach of contract by IBM.

In May 2003, SCO began distributing a "letter to Linux users," which expressed a number of unfavorable opinions about open source, including that although closed source software is built by carefully selected and screened teams of programmers working to build proprietary, secure software, the open source Linux operating system "has been built from contributions by numerous unrelated and unknown software developers, each contributing a small section of code. There is no mechanism inherent in the Linux development process to assure that intellectual property rights, confidentiality or security are protected. The Linux process does not prevent inclusion of code that has been stolen outright, or developed by improper use of proprietary methods and concepts."

Of course, SCO may have reason to enhance the risks associated with open source software development, but its case against IBM also highlights the palpable threat that open source development represents for proprietary software developers.

Endnotes

[1] Anonymous poster, Slashdot.org, July 16, 2000.

[2] Interestingly, the federal government's Federal Acquisition Regulations (FAR) secures for the government as an end-user and a licensor of computer software the same rights as those supported by the open source community, except for the right of public distribution. Section 52.227-19 of the FAR, Commercial Computer Software—Restricted Rights, provides, in the pertinent part:

(a) As used in this clause, "restricted computer software" means any computer program, computer data base, or documentation thereof, that has been developed at private expense and either is a trade secret, is commercial or financial and confidential or privileged, or is published and copyrighted.

(b) Notwithstanding any provisions to the contrary contained in any Contractor's standard commercial license or lease agreement pertaining to any restricted computer software delivered under this purchase order/contract, and irrespective of whether any such agreement has been proposed prior to or after issuance of this purchase order/contract or of the fact that such agreement may be affixed to or accompany the restricted computer software upon delivery, vendor agrees that the Government shall have the rights that are set forth in paragraph (c) of this clause to use, duplicate or disclose any restricted computer software delivered under this purchase order/contract. The terms and provisions of this contract, including any commercial lease or license agreement, shall be subject to paragraph (c) of this clause and shall comply with federal laws and the FAR.

(c)(1) The restricted computer software delivered under this contract may not be used, reproduced or disclosed by the Government except as provided in subparagraph (c)(2) of this clause or as expressly stated otherwise in this contract.

(2) The restricted computer software may be—

(i) Used or copied for use in or with the computer or computers for which it was acquired, including use at any Government installation to which such computer or computers may be transferred;

(ii) Used or copied for use in or with backup computer if any computer for which it was acquired is inoperative;

(iii) Reproduced for safekeeping (archives) or backup purposes;

(iv) Modified, adapted, or combined with other computer software, provided that the modified, combined, or adapted portions of the derivative software incorporating any of the delivered, restricted computer software shall be subject to same restrictions set forth in this purchase order/contract.

[3] (a) Making of Additional Copy or Adaptation by Owner of Copy. Notwithstanding the provisions of section 106, it 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 in conjunction with a machine and that it is used in no other manner, or

(2) that such new copy or adaptation is for archival purposes only and that all archival copies are destroyed in the event that continued possession of the computer program should cease to be rightful.

[4] The licenses discussed throughout this book are excerpted in the Appendix at the end of the book.

[5] CORBA is the acronym for Common Object Request Broker Architecture, an open, vendor-independent architecture and infrastructure that computer applications use to work together over networks.

3

Drafting Open Source Licenses

Regardless of whether it is free speech or free beer that is at stake, some open source licensors draw the line of what is not "free" at free-riding commercial uses.

In its most fundamental respect, the purpose of an open source software license is not unlike the purpose of any type of copyright license; namely, the purpose of issuing a license is to allow the copyright owner to rely upon a legal means of conveying or transferring a nonexclusive copyright interest in the software to another party while retaining lawful ownership of the copyright. In other words, the license is used as a means for granting a set of permissions to a licensee. Of course, a software license agreement may contain a number of provisions that impose restrictions as well as grant permissions, but the fundamental purpose of issuing a software license is to signal the copyright holder's authorization of the licensee's use of a copyright-protected work in the manner permitted by the license, and constituting a use within the legal scope of copyright law. If the licensee's conduct is legally permitted without the license, then the copyright license may serve little or no purpose under copyright law. Hence, the purpose of issuing a software license should include the desire to grant a licensee permission to act in some accord relevant to copyright law. A software copyright license may govern the right to reproduce and distribute copies of the software, the right of public performance and display (if applicable), and the right to prepare derivative versions of the software program. Although these rights are broad, they do not encompass every imaginable use of a copyright-protected work. If your purpose in drafting a software copyright license is restricted to concerns about noncompete obligations of those who work on development projects with you or only involves trade secret matters, then the use of a copyright software license warrants substantial reconsideration. You may not have the right legal tool for your objective.

Although the number or type of provisions that could be included in a software copyright license agreement is plausibly unlimited, several factors counsel against issuing verbose or extensively drafted licenses, including common-sense factors such as the fact that a concise and clearly written license is more likely to be read by the licensee. In that regard, there are likely to be seven to ten provisions covered by most software licenses, and each provision might form an entire paragraph or section of the license agreement: (1) the Grant clause, (2) the license term period, (3) a provision regarding compensation or, perhaps, consideration, (4) a warranties and obligations clause, (5) a notice of infringement provision, (6) a termination provision, (7) indemnity, and (8) a severability provision. Some drafters may prefer to add a couple more sections, including a jurisdiction and dispute settlement clause and an assignability clause. These provisions will be discussed in appropriate detail throughout the remaining chapters.

The easiest way to prepare an open source license is to adapt an existing open source license for your own use. As noted in Chapter 2, the OSI maintains a list of approved open source licenses on its Web site at www.opensource.org, and this list provides an excellent starting point in drafting a recognizable open source license. A license from that list may serve as a useful template for adapting a license for the unique needs of nearly any open source project. Yet, a major shortcoming in relying on license templates is the tendency to accept the template and use it as is. Although most open source software licenses approved by OSI are precisely written to meet the obligations and objectives of open source software development projects, software creators should steer clear of the tendency to port a template license as is. If the license is not annotated, it may be too easy to miss the significance of a license term appearing innocuous or pertinent to all open source licenses.

Even the terms of the most common open source license template, the GNU GPL, are constantly debated and occasionally redrafted to avoid unanticipated defects in meaning or interpretation; other popular or common open source licenses, such as the original Artistic License, should rarely, if ever, be used as templates, because their terms are either unclear or controversial [1]. Of course, when drafting an open source license, there are benefits to avoiding the time-wasting reinvention-of-the-wheel phenomenon; hence, the use of a license template is a good way to begin drafting a license, since a good template is intended principally to provide a guideline for implementing a license with terms that are adapted to your needs.

Open source software licenses are primarily drafted by the licensor without the encumbrance of negotiations. As such, license drafting requires the licensor to answer the basic questions in the text of the license, including who, what, where, when, and how. Failure to answer one of these questions means someone else will—and if it is a court, courts follow well-settled rules that could result in the interpretation of an ambiguous term in your license under a meaning far from that which you intended as the drafter of the license. To draft an effective software license, the lawyer and the software developer must use clear terms with accepted meanings. In other words, license drafting is not the time to be creative, fresh, and original in written expression.

BSD License

An example of how the open source community promotes the use of open source software development among distinct business models is the development effort that originally adopted what has come to be known as the BSD License [2]. This open source license template has been used by at least two widely recognized open source projects, the X11 Project, noted in the next chapter, and the MIT License Open Source Project. These license templates are unique in that they are what is best described as *unrestricted licenses* [3]. In other words, the BSD License template allows open source developers to adopt a license that permits either the restrictive copyleft open source licensing or the permissive or no-copyleft licensing under subsequent public distribution. Given this latitude in participation in an open source project, one might expect the BSD License or one of its iterations to be widely adopted, but this is not so when compared with the use of the BSD License with the more restrictive open source license: the GNU GPL.

In fact, the most restrictive open source license, the GNU GPL, as discussed in the previous chapter, still remains the indisputably most popular, most openly adopted open source/free software license to date. Many open source software projects use the GNU GPL to distribute software. There are several reasons for this: (1) the GNU GPL's popularity exceeds the understanding of the license terms, (2) earlier versions of the BSD License were somewhat controversial, and (3) within the open source community, a number of participants are anxious about code forking and free-riding, which are widely permissible by unrestricted open source licenses. Of course, the strength or popularity of unrestricted open source licenses is likely to be higher than it appears. As a practical reality and, perhaps, due in part to a lack of awareness that permissible unrestricted open source licensing exists, it is difficult to accurately document the number of businesses and developers seeking unrestricted use of open source software—inevitably, some will use open source software in any manner that they can, including simply taking the source code [4] and using it in similar proprietary projects or embedding open source within larger proprietary software projects [5].

Even so, the permitted use of unrestricted open source licenses is popular among a significant number of open source developers, especially large software companies like IBM or Apple. For adherents of unrestricted licensing, the risk that some developers will use their source code by closing it off and hiding their own enhancements in a proprietary product (code forking and free-riding) is largely counterbalanced by a greater concern that the restrictions [6] imposed by the terms of the GNU GPL essentially give competitors an unfair equal access to the copyright holder's own development efforts without granting a privileged role or competitive edge to the original copyright holder. Regardless of the soundness of this position, the fact that some developers assert sound business reasons for participating in open source projects that permit them to take their enhancements down the road toward proprietary development at any moment is likely to mean that unrestricted open source licensing will remain an important tool in the open source community.

Unrestricted access to source code is actually anchored in the academic model used by scientists and scholars, who often impose no restrictions upon the use of scholarly works by others. In academia, research projects and the resulting scholarship are essentially open source projects without the luggage of open source labels. Research is funded and then produced so other scholars may examine the findings, improve upon the body of knowledge, or fork it in other directions. Indeed, it is not surprising that research scholarship predating the field of computer science, but having to do with computing technologies or networking, often led to the creation of a number of computing tools that, along with research documentation, were made freely available [7].

Before getting into the specifics of the BSD License template, a little background on BSD is helpful in understanding the peculiarities of the BSD software. What looks like the harbinger of the recent UNIX-Linux litigation between SCO and IBM, as discussed in Chapter 2, involved disputes over the UNIX operating system software between AT&T and the University of California at Berkeley, which presumably was not resolved until Novell stepped in to purchase AT&T's UNIX software unit (an action that, you will recall, Novell claims is pertinent to the resolution of the current SCO-IBM litigation).

When AT&T managed the development of the UNIX operating system, it did so in a decentralized manner, which enabled (or resulted in) the development of several forks of UNIX, including one by the University of California at Berkeley. Most commentators view AT&T's loose development style (AT&T provided no support, no bug fixes, and no credit to those creating their own version of UNIX) to be due to legal concerns connected with antitrust litigation with the U.S. government and to a failure to recognize the potential commercial value of the software. It is true that AT&T may not have been known how actively it could pursue its interests in the nascent computer industry without a consent decree. Commentators seem to accept that there is a general connection between the effects of the antitrust litigation and AT&T's reluctance to take on a more centralized role in managing the development of early versions of UNIX. Notably, AT&T's lawyers were probably far too involved in software development decisions, including the (likely) mistaken decision to allow educational institutions to freely distribute copies of UNIX, which ultimately resulted in the development of unique versions of UNIX, when AT&T denied educational institutions free support or bug fixes for the software. This misguided practice forced UNIX end-users to share ideas, information, programs, bug fixes, and hardware fixes with one another.

By 1993, two versions of the UNIX operating system had begun to be widely and freely distributed, FreeBSD and NetBSD, under the BSD License. The BSD software is primarily used as a Web server operating system. It is distributed under an open source license, which allows licensees open access to its source code and authorization to modify the source code without the restriction of being required to provide the changes or modifications to the copyright holder or the open source project [8]. This form of unrestricted use of a project's source code may have resulted more from the past custom and practice of the use of the UNIX operating system than with the freedoms exalted by open source and free software adherents. Clearly, the unrestricted uses resulted in inevitable code forks. What seems most apparent, however, is that unrestricted open source development projects require project leaders who have sharp project management skills; these skills are necessary to prevent the inevitable free-riding from undermining the desire of programmers to participate in the project and to ensure achievement of the project goals.

BSD License Template

"Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met." The preceding is a provision of the BSD License template that constitutes the grant rights clause. This is the most important provision of any open source license, and it should reflect that importance by being clearly expressed and precisely written.

The BSD grant rights provision grants to the licensee the right to publicly distribute the copyright-protected work or any modification of it. It also grants the licensee the right to *use* the work.

Ideally, the grant clause should adopt a term that is clearer than the term *use* or that expresses what uses are covered. In other words, the grant rights clause should expressly and precisely state what copyright interests other than distribution are being granted. If the copyright holder intends to grant a right outside the scope of copyright, then a term or phrase reflecting that intention is needed to avoid the potential that the licensee may underutilize the work. Aside from possibly meaning to include rights outside the scope of copyright, reliance upon the term *use* does not sufficiently designate with specificity and certainty which interests within the scope of copyright are being granted to the licensee. For instance, does *use* mean display? Although the copyright holder may understand or believe that in the context of software, *use* is likely to include the right to copy and display a work, the belief may be legally unfounded or unknown to the licensee. At any rate, a reliable rule of thumb for license drafting in almost any context is that it is particularly worthy of the effort it takes not to leave your licensee guessing about the meaning of the terms in the grants clause.

The grant clause should also follow the customary approach of copyright licenses by explicitly indicating that the grant offered to the licensee is a nonexclusive one. Generally, it should be fairly apparent to most software users that open source licenses are not intended to transfer interests exclusively to a single licensee, but the fact that the license grants a bundle of copyright interests to the licensee warrants an explicit disclosure of the nonexclusivity of the license.

The BSD License template also provides that "redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution." The two preceding provisions contain conditions that apply to source code and binary code distributions, respectively. The conditions are (1) pass on the warranty disclaimer to subsequent licensees, and (2) ensure that the copyright notice is retained.

Generally, these two provisions, along with the absence of a copyleft provision anywhere in the license, amount to what really makes the BSD License unique. In this regard, it is apparent that the BSD License is an open source license that is intended to fulfill two objectives: to offer licensees the opportunity to participate in typical open source development efforts and to offer software developers who might not otherwise participate in the open source community an opportunity to use the work of open source as well as contribute to open source without being precluded from also using the same work in a closed or proprietary software development business venture.

The extent to which the BSD License fulfills its twin objectives is unclear, since to some extent they have cross-purposes that require exceptionally careful management by the original copyright holder. One notable indication that the objectives of the BSD License can be successfully accomplished is the open source software development project for which the BSD License was given its name, namely, the University of California at Berkeley, which achieved astonishing success with its popular version of the UNIX operating system software.

Since there is no clearly expressed copyleft provision in the BSD License template, this template is useful when the licensor wants to draft a license that does not require immediate access to source code by subsequent licensees. In addition, where the license permits access to source code by subsequent licensees, code forking is also accommodated for distributions of modified works. Hence, any software project managed under the BSD License could result in several versions of the same work being publicly distributed.

To adherents of the FSF or to those who admire the GNU GPL, the relative openness of the BSD open source license might seem oddly fatalistic to open source objectives, but one should keep in mind that one of the clear successes of the open source community, the Linux kernel and its concomitant operating system, might not have sustained the successes it achieved if Linus Torvalds had not decided to manage the development process of Linux with an eye toward the same objectives of the BSD License. In this regard, it is important to be mindful of an overriding objective of the open source community, namely, to support a software development framework that includes a number of alternative business models attractive enough to others that they find a compelling need to join the community in building open source software.

The BSD License template includes a provision regarding the right of attribution. Specifically, it provides that "neither the name of the <ORGANIZATION> nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission." The preceding provision, which is included in what is known as the current version of the BSD License template, is the only provision of this license that differs from a related and popular open source license template: the MIT License template.

The MIT License lacks this clause, but it should be apparent why an entity might include the provision. Entities like the University of California at Berkeley (and MIT as well) understandably want to preclude anyone from deriving unintended benefit from the institution's enormous good will by simply distributing a version of open source software under the pertinent license.

Even so, the provision is likely to be unnecessary for most open source developers. Instead, the provision could be replaced with a disclaimer as to implied license to use the licensor's logos or trademarks without prior written permission. For example: "Except as otherwise provided for in this license or consistent with noncommercial purposes, the name_or logo_may not be used in any manner in connection with this software product or any software product derived from this software product without prior written permission."

The final provision of the BSD License template is the disclaimer clause, a boilerplate that is intentionally printed in all capital letters. "THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS AS IS." The MIT License template is substantially similar to the BSD License.

Restricting Commercial Uses

In direct contrast to the permissive, unrestricted uses of the BSD License template, which is viewed as a significant tool for recruiting commercial interests to join the open source community, there are open source licenses that have quite different objectives, and some of these licenses contain clauses that attempt to restrict commercial uses. Here, we examine what software license provisions are relevant when the open source licensor seeks to restrict the distribution of its software for commercial purposes. For some within the open source community, the implementation of such a restriction is considered highly undesirable, but for others, the participation within the open source community is significantly related to the desire to distribute open source software with a commercial use restriction.

The goal of some software developers is to develop and freely distribute modules, libraries, or interfaces that more efficiently enable third-party applications (or so-called middleware) to interact with each other, a computer's hardware, or an operating system. In doing so, these developers may want to limit the use of their modules to noncommercial uses [9]. For adherents of unrestricted licensing, the risk that some developers will copy the source code of an important module or library, improve it, and hide the enhancements in a proprietary product is counterbalanced by the greater concern of ensuring open access to source code and cultivating an environment for developing open technology standards [10]. As a backdrop to understanding what issues arise when an open source developer drafts a license that restricts a range of uses, including commercial uses, the next section begins by exploring some of the basic issues that arise when a use restriction imposed on licensees is intended to cover modules or libraries that have been created as part of an application or operating system: occasionally the technical aspects of some use restrictions or open source licensing conditions appear so arcane that it becomes difficult to determine whose rights are really affected.

Although functions performed by software were frequently hardwired into the electrical circuits of a computer to be performed by the hardware, modern computing has essentially rendered many computing devices useless without software. There are two types of software typically found on a desktop computer: operating system software and application software. Operating system software often provides several functions, including access to an application programming interface (API), which makes it relatively simple to develop application software by reducing the amount of code that must be written or rewritten.

Operating system software also aids computer users to benefit from the useful life of a computer by simplifying the exercise of certain hardware functions, such as printing or keyboard entry. In this respect, the operating system acts as a collection of programs that manages the computer's internal functions and acts as an interface between the computer and the computer user. Application software, on the other hand, consists of one or more computer programs that fulfill a specific function for the user, like word processing. Even the contour of the Linux operating system is difficult to define. Linux is licensed under the GNU GPL and referred to as either Linux or GNU/Linux. What actually constitutes Linux or the GNU/Linux is open to slight debate or considerable confusion. Over the years, many useful packages that are, perhaps, not essential to the operating system software have been added to certain versions of GNU/Linux, but excluded from others. Most would agree that the essential aspects of the GNU/Linux operating system include the kernel, a windowing user interface, compilation tools, Bash, the C library, Emacs, X11, TeX and Texinfo, Ghostscript, and a host of other network utilities. Consequently, it is worth noting at the outset that the task for the licensor who wants to draft a license with limitations regarding certain modules or libraries that may easily function as part of an operating system specification or as part of a designated application is often likely to be difficult. Limitations or use restrictions that refer to both application software and operating system software modules should be identified specifically in the license. Section 3 of the GNU LGPL provides an example of a carefully drafted provision with regard to the use of certain libraries linking either at compile-time or run-time. Even though the GNU LGPL permits risky linking of open source applications with proprietary libraries, the license attempts to minimize the risk of failure at run-time by clear references to the critical importance of maintaining functionality, despite the emphasis on linking two distinct applications.

An open source license may be used to distribute either application software or operating system software. In determining what license template is more appropriate for a given software distribution objective, it may be critical to determine whether the software is operating system software or application software. Operating system software often involves multiple applications or programs, and restrictions on use might be carried forward for each program or component of the operating system software. Hence, many issues regarding whether a given open source license is compatible with the widely popular GNU GPL arise out of the practice of distributing open source software that integrates multiple licensing conditions from distinct developers.

A great deal of planning is essential in classifying the functionality of the software module or program. A computer program is defined under the copyright statute as "a set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result." Computer programs are written in many different languages, each with their own unique syntax and structure.

Originally, programs were written in machine language. As the machines and operating system software became more complicated, programmers began to write in more sophisticated symbolic languages. While these languages were much easier to code in than machine language, computers could not understand them. Therefore, the program goes through a step called compilation to turn the original source language into a language understood by the computer. The source language is called *source code*, and the language understood by the computer is called *object code*. When drafting a condition or provision concerning linking libraries or, more generally, the state of the code (compile-time versus run-time, for example) the licensor should ensure that the carefully worded condition applies to both copyrightable subject matter as well as matter for which the licensor owns the copyright. Before applying the technical linking issues to legal concepts, it is likely to be more helpful to bridge this issue in a context, namely, as part of the discussion that follows next—commercial use restrictions.

At the heart of open source licensing is the idea that the best software is developed under conditions where software is freely distributed in a manner that permits any licensee to modify the software. With this backdrop, we explore what types of licenses have been used to modify this conception of open source by restricting the licensee from using open source for certain specified commercial uses. There are two popular examples of open source license templates that are used by some open source software developers to impose restrictions on commercial uses: the Aladdin Free Public License (AFPL) and the Q Public License; these licenses are discussed in this chapter.

As with the requirements of the GPL, if a developer plans to distribute or publish any work that contains or is derived from code licensed under Aladdin, then (1) the work must contain prominent notices stating that it is a modified version and providing the dates of the changes made; (2) the work must be licensed as a whole to all third parties under the terms of the license; (3) copyright notices, instruction on how to view the license, and disclaimers of warranties must be displayed each time the work commences operation, if the work reads commands interactively when run; (4) the work must be accompanied by the complete corresponding source code; (5) any distributed written or printed material must include a written copy of the license or a notice that the work is covered by the license and instructions on how to view the license; and (6) no further restrictions other than those enumerated in the license may be imposed on the recipient of the code.

AFPL

No doubt, commercial application of open source software development occurs in a robust and wide-ranging environment. Indeed, despite erroneous reports in some popular media publications, there should be no mistake that open source licensing is primarily targeted toward some form of commercial purpose. Use of unrestricted open source licenses is popular among a significant number of open source developers, especially large software companies like IBM or Apple.

For adherents of unrestricted licensing, the risk that some developers will use their source code by closing it off and hiding their own enhancements in a proprietary product is counterbalanced by the greater concern that the restrictions promoted by the GNU GPL give competitors an unfair equal access to the copyright holder's own development efforts. Regardless of the soundness of either side's positions, unrestricted open source licensing remains an important tool in the open source movement. Even so, there are exceptions where a licensor's objectives include restricting commercial software distribution of their copyright-protected software program.

The restrictions of primary concern are those that accompany the copyleft provision, which essentially keeps the open source project an open source project. The restriction does not mean that the software distributed with a GNU GPL template cannot be directed toward a commercial purpose. Instead, the GNU GPL template precludes lawful attempts to distribute an open source or free software program as a proprietary product. The restrictions against proprietary software distribution, notwithstanding, a range of commerce-oriented business models, are available to the developer distributing software under a software license containing a copyleft provision. An interesting peculiarity of the AFPL is how often it is mistaken as a carbon copy of the GPL. In this case, looks are deceiving because the AFPL provides additional restrictions not found in the GNU GPL.

First, the license explicitly prohibits a commercial organization from accepting money for the software, except in limited circumstances to cover the costs of distribution of the code. Second, licensees may not distribute a free version of the software in a distribution medium containing nonfree software. In the view of the FSF, the AFPL is not a genuine free software or open source license because of its restrictions on charging for distribution.

Many open source adherents have argued for permitting the use of AFPL for dual-purpose open source projects where the project leader desires to maintain open access to the project's source code, but retain or restrict control over commercial uses of the software project. Clearly, the use of the AFPL for open source projects is intended to encourage software developers predisposed to proprietary models to join the open source community; yet this approach is not without controversy.

Projects that are intended to enjoy the free and open participation of an army of open source programmers are not likely to obtain much success if the project adopts the AFPL, since many open source developers are loathe to join projects with commercial control restrictions. On the other hand, large proprietary software developers, like Sun Microsystems (Java and Star Office) or AOL (Netscape) may find remarkable success using open source licensing with projects that set limits or restrictions on third-party commercial uses of their software.

With regard to the AFPL specifically, as long as the licensee complies with the restrictions of the Aladdin license and distributes an unmodified copy of the license with the code (though license modifications may include a description of the licensed work and the law of the country where the work was created), the licensee may freely copy and distribute literal copies, modified copies, or derived versions of the software's source code throughout the world, in any medium.

The Aladdin license explicitly restricts distribution of the program or any derivative works by a commercial organization to a third party if it receives payment in connection with the distribution. The license allows distribution fees as long as the fees are not content-dependent. The restriction against commercial organizations charging for software distribution is somewhat vague. If read narrowly, the AFPL's restriction merely adds a gloss onto the copyleft concept set forth in the GNU GPL by implicitly allowing noncommercial organizations to assess a fee for certain distributions. In general, the GNU GPL itself precludes charges for software distribution under certain conditions, although, of course, licensors of the GNU GPL may charge for other services that accompany the software distribution.

If the AFPL's restriction is read more broadly, however, the AFPL might be subject to serious challenges within the open source community as to whether the license template should be considered purely open source. Indeed, the OSI's highly regarded OSD currently prohibits open source licenses from discriminating against groups or particular business entities under the terms of the license. In this respect, the AFPL's restriction against commercial organizations might violate the OSD. If so, use of the AFPL as a template for an open source license would come with the considerable limitation of not meeting OSI's requirements for participation in its license approval and trademark certification program.

One example of the use of the AFPL template is the distribution of the set of software developed under the Ghostscript name. Ghostscript is the name of a set of software that provides an interpreter for the PostScript language and the Adobe Portable Document Format (PDF—sometimes confused with Acrobat, Adobe's PDF browser and editor product); input modules (utilities); output modules (drivers) for a wide variety of window systems (including X Windows and Microsoft Windows); printer raster file formats; and the Ghostscript library, a set of procedures to implement the graphics and filtering capabilities that are primitive operations in the PostScript language and in PDF.

In simple terms, Ghostscript can read a PostScript or PDF file and display the results on the screen or convert them into a form you can print on a non-PostScript printer. Using Ghostscript, an end-user can view or print an entire document or selected pages regardless of whether the end-user's printer is equipped to process PostScript. Distributing Ghostscript under the AFPL license template (Ghostscript is also distributed under other open source licenses) ensures that licensees are allowed free use, copying, and distribution by end-users, but does not allow commercial distribution.

Unrestricted access to source code is actually anchored in the academic model, as noted earlier. As is generally true of open source projects, notwithstanding that academic research is produced so that other scholars may examine the findings, improve upon the body of knowledge, or fork it in other directions, the existence of contracts, the sources of funding, and the ownership of copyright may explain why some academic research may not be considered immediately in the public domain.

Contract Formation Issues

AFPL License Template

> …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. Nothing other than this license grants you permission to modify or distribute the program or its derivative works. These actions are prohibited by law. If you do not accept these terms and conditions, do not modify or distribute the program.

The provision above seems to indicate that the authors of the AFPL appear to express a copyright interest in the license itself. In this respect, wholesale copying of the AFPL is not advised unless compliance with the terms of the initial paragraph of the AFPL are complied with or permission to reproduce and distribute the license is otherwise granted. Even so, if use of the AFPL as a license template is desired, the ideas embodied within the AFPL may be freely adopted, since copyright only extends to expression, not ideas enabling expression. Applying open source concepts to the text of a license (rather than the software the license applies to) presents an entirely distinct set of circumstances under copyright law; namely, there is considerable doubt whether a typical open source license itself is subject to the benefit of an open source license under copyright law due to the doctrines of merger and the idea/expression dichotomy.

For the purposes of setting forth distinctions between open source licensing and other forms of software licenses, this license template, like other open source licenses, seems to rely upon a less certain form of acceptance of license terms by the end-user or licensee. Ostensibly, a licensee must consent to the terms of the license in order to be bound—as is true of any contract. Since most open source licenses are Web-based (also known as browser-wrap) agreements, an end-user's consent to terms often manifests itself in unique ways. The last provision may appear to establish the terms of consent, but, more precisely, it describes the conditions of the license grant. A provision more conspicuously setting forth the form of the consent expected might avoid a dispute concerning contract formation.

Many browser-wrap agreements structure contract formation around rules where the end-user accepts the terms of the license by clicking and downloading. Is this legally sufficient? What else should be necessary to manifest assent to an on-line agreement? Courts have not rendered authoritative answers, and commercial codes have provided little guidance thus far (states are haltingly moving toward adopting UCITA). Certainly, if clicking is insufficient, there might be serious problems to overcome in some sectors of e-commerce.

Licenses

> Licensor hereby grants you the following rights, provided that you comply with all of the restrictions set forth in this License and provided, further, that you distribute an unmodified copy of this License with the Program:
>
> (a) You may copy and distribute literal (i.e., verbatim) copies of the Program's source code as you receive it throughout the world, in any medium.
>
> (b) You may modify the Program, create works based on the Program and distribute copies of such throughout the world, in any medium.

The grant clause of the AFPL includes a provision allowing the licensee to modify the licensed software and to distribute copies of the modification in compliance with the terms of the license. In this respect, the grant clause is typical of others found in open source software licenses. Within the open source community, there is some debate as to whether the provision permitting modifications to the licensed software, if violated, would be enforceable without distribution of the software. Some have argued that an illicit modification and copy used in-house (hence, without public distribution) should either be deemed fair use or simply would not breach an open source license, which is understood by some to depend upon public distribution as the primary vehicle of enforcement.

Ultimately, the issue may rest upon other terms in the license; however, section 117 of the Copyright Act (17 U.S.C. 117) extends the first-sale doctrine to software by permitting licensees or end-users of software to make modifications (the Copyright Act uses the term *adaptation*) or copies of the licensed software in a computer's memory as a function of using the software, and in doing so, the licensee or end-user does not infringe the licensor's copyright interest.

Consequently, the modification provision itself comes within the licensor's copyright interest, but not under circumstances where section 117 applies.

The licensee without public distribution may breach the modification provision, but such is not likely if the modification or copy is necessary to operate the licensed software. For example, where the licensee's program makes a call to the library file of another program or operating system API, section 117 would render the dynamic linking permissible without explicit authorization from the copyright holder of the linked library file. In this regard, to the extent that an open source license provision explicitly prohibits a section 117 modification without compliance with the additional terms of the license, the open source license may be at odds with the first-sale doctrine. Although a court may choose not to enforce such a provision, what is of more concern would be classifying as an open source license a licensing provision that withdraws end-user privileges rather than enhancing them. The license grant provision should clarify that an end-user or licensee's section 117 privilege is not affected by grant rights clause.

It is very common in negotiated transactions to allocate infringement risk between licensor and licensee or for the licensee to assume all risk of infringement. The sheer number of issued patents, the difficulty of conducting patent searches, and the fact that any given patent can be interpreted dozens of ways make placing the risk on the licensor inequitable in many cases. Often the licensor cannot obtain insurance or will not receive enough income from the license to offset the risk of providing a noninfringement warranty (in many transactions, the licensee will receive much more income through use of the software than the licensor who supplied it). The smaller the software developer or publisher, the more likely the developer or publisher is to resist shouldering the risk of a full-blown noninfringement warranty.

For most off-the-shelf, mass-market software products, the user expects a perpetual license subject only to cancellation for breach. The same expectation is true for licensed informational content that the licensee integrates or combines with other information to create a single product: the licensee does not expect to have to rip the combined product apart at the behest of the licensor.

Many potential adherents of open source come to the community seeking to pursue a software development project that supports a specific platform, a specified noncommercial interest, or some narrowly drawn objective that limits the type of end-users who may be granted the benefits of open source licensing. The OSD discourages this type of narrowly drawn open-source licensing because the open source community views discriminatory restrictions, generally, as a potentially pernicious business model. In other words, open source software ought to be open to any potential licensee who may qualify for licensing, rather than limited to those who are preferred benefactors of a narrowly drawn objective. For example, a software developer may seek open source licensing for a project that provides free and unrestricted access to source code, but for only noncommercial end-users.

In this respect, the software developer may intend to support end-users who use the software for private use and, perhaps, those who use the software for public research purposes. Although it is undeniably understandable why such goals might be important to some software developers, open source licensing does not support the practice, because doing so would promote discriminatory uses of open source software. Licensing provisions containing restricted use purposes cannot be approved as open source licenses. Of course, there are numerous ways to support software development for public research or similar objectives while also developing open source software. What is critically important is that an open source software license extends the privilege of copying, distribution, and modification of the software to all lawful licensees regardless of secondary objectives.

Use Restrictions and Commercial Organizations

Restrictions

> This license is subject to the following restrictions:
>
> Distribution of the Program or any work based on the Program by a commercial organization to any third party is prohibited if any payment is made in connection with such distribution, whether directly (as in payment for a copy of the Program) or indirectly (as in payment for some service related to the Program).

The provision set out above pertains to a user restriction condition imposed on commercial organizations. Whether such restrictions should be viewed as categorically violating the terms or spirit of the OSD depends upon how broadly the term *commercial organization* is defined by the licensor. The final two paragraphs follow typical open source license templates that address conditions whereupon subsequent software distributors must ensure open and public access to the source code and that access not be hindered by excessive fees for distribution media or on-line access. OSI has the duty to ensure that open source software carrying OSI's certification trademark or whose license is listed on OSI's license approval list posted on its Web site complies with the OSD. To accomplish this task, OSI provides an e-mail link on http://www.opensource.org that current or potential open source licensors may use to submit a proposed license. Proposed open source licenses are reviewed by OSI's board on a rolling basis.

One license template that seeks to achieve those objectives is known as the Artistic License. Although this license template is discussed here because of its popularity (or, more precisely, the popularity of the software the license is intended to protect), many of the flaws in the license arise from the fact that its use as a template is far beyond the intended purpose of the drafter. The Artistic License was initially intended to cover only the Perl language project, but many programmers who programmed in Perl also used the Artistic License to protect their own projects despite language in the license itself seeming to prohibit this use.

Today the FSF recommends that the Artistic License continue to be used by programmers who create free software programs using Perl "to promote coherence and uniformity in Perl programming." Complicating matters further, the Artistic License has undergone clarification and revision. What this really means is that there are at least three iterations of the official Artistic License: the Clarified Artistic License, the Artistic License for Perl, and the Artistic License, which is excerpted below.

The cloud or haze over this unusually high level of confusion for a license meant to support moral rights in the integrity of a work should soon be removed as a result of publication of the "Artistic License 2.0." Although version 2.0 has been published, its use is undocumented, and it may undergo further refinement before the authors release it for widespread use; the actual timing of the release of version 2.0 of the license may be set to coincide with a future release of an updated version of Perl. Consequently, the license version discussed here is the Artistic License, which is currently in use and likely to remain a popular license for current Perl projects.

Artistic License

Preamble

> The intent of this document is to state the conditions under which a Package may be copied, such that the Copyright Holder maintains some semblance of artistic control over the development of the package, while giving the users of the package the right to use and distribute the Package in a more-or-less customary fashion, plus the right to make reasonable modifications.

A preamble is a useful provision in a license because it can establish the intent of the licensor in general, less precise language than the main parts of the license, which must be precisely written. Even so, the preamble must be drafted carefully to avoid adding terms or ambiguity to the license. The reference to *artistic control* appears to refer to the copyright holder's intent to protect moral rights, which is not an explicitly protectable interest of copyright in the United States for software.

In this respect, the use of the term *artistic* must be intended loosely to denote that the copyright holder intends to manage the software development project or to retain the authority to do so while providing a generous license grant consistent with open source objectives. The popular Web site scripting language, Perl, was initially distributed under dual licenses, the open source Artistic License and the GNU GPL; the Artistic Public License was used presumably to protect the name and the integrity of the name Perl, and many who desire similar artistic protection still use the Artistic License for that purpose.

Definitions

> "Package" refers to the collection of files distributed by the Copyright Holder, and derivatives of that collection of files created through textual modification.
>
> "Standard Version" refers to such a Package if it has not been modified, or has been modified in accordance with the wishes of the Copyright Holder as specified below.
>
> "Copyright Holder" is whoever is named in the copyright or copyrights for the package.

The definitions section contains an awkwardly worded definition of copyright holder. Since the definition is tautological, it should be avoided. A good definitions section is helpful if it is used to define terms that are contained in the license and that are either unusual terms or concepts or common terms or concepts used in an unusual manner. Terms that are not defined or licenses that do not contain a definitions section are often interpreted by their ordinary or common meaning.

Quite obviously, a principal concern for computer software producers is providing adequate protection for their programs. Generally, a programmer writes software in source-code form, which is in a language understandable by humans. Once the source code is written, the program may be processed through a compiler that produces the object code.

The object code is comprehensible to the computer on which it runs, but not to humans. Because of this unique way computer programs are created, the programs are susceptible to reverse engineering or decompiling. Thus, the jealously guarded secrets of the software producer can be discovered by anybody who is willing or has a financial incentive to go through the process of reverse engineering. As a result, software authors include an express provision prohibiting disassembly or reverse engineering in their licenses.

> (2)…You may distribute the programs of this Package in object code or executable form, provided that you do at least ONE of the following:
>
> …(4)…distribute a Standard Version of the executables and library files, together with instructions (in the manual page or equivalent) on where to get the Standard Version…accompany the distribution with the machine-readable source of the Package with your modifications.

The foregoing provision essentially establishes what modifications to the open source project, when added to the source code, become part of the project's copyright. Notably, this provision makes no mention of modifications constituting an original work created by the licensee, except for public domain source code. Although this exception is or may be quite significant, the licensees are forewarned that their contribution will be subsumed by the project in practically all instances.

In the event that the licensee's modification is not captured by the provision above, then the license does not disturb the licensee's potential or likely copyright interest. It does appear that few modifications will qualify, which may mean that the provision reaches too deeply into the copyright interests of potential contributors, especially where licensees develop modifications from public domain source code that become sufficiently original modifications. In addition, the provision allows those who are not bound by the provision to essentially agree to be bound by its effect, but retain copyright. As a practical matter, such terms seem to present a number of obstacles to subsequent enforcement of copyright by the licensee. Making a modification freely available, for instance, or placing it in the public domain has become a vague concept when used in the abstract and not clearly defined. If adopting the Artistic License template, a licensor should be careful to improve upon the template by defining the unclear terms.

There is one additional reason why the use of terms in the Artistic License template is fuzzy. The general use of the term *modified work* or *modification* in many open source licenses is often unclear. The concept of derivative work does not apply as nicely to software as it does to other forms of intellectual property. Although that may seem to be a justification for actually using the term *modified work* instead of *derivative work*, it best supports using neither term when the licensor is addressing fuzzy software development concepts like the gray area distinctions between static and dynamic linking between or among programs.

Of course, since the copyright law sets out a definition for *derivative work*, licensors should have great reluctance to use *modified work* instead. A difficulty arises, however, if the licensor seeks to control modifications that would not constitute derivative works. In two general instances, this might be the case: (1) where the licensor intends to accept bug fixes or patches that simply do not rise to the level of originality under copyright law to be considered a derivative work, and (2) where the licensor seeks to encourage contributions of modifications that are transformative.

Static and dynamic linking presents a complex problem for open source because software developers are in general disagreement over marginal distinctions between dynamic and static linking, as well as whether such distinctions have copyright significance regarding whether dynamic linking, in particular, may constitute the exploitation of a copyright interest of the work being linked (i.e., if static linking would clearly constitute copyright infringement—unauthorized derivative work, unlawful reproduction—without express permission or fair use, would dynamic linking be so as well?).

Linking issues deserve sufficient attention because the prevailing programming paradigm emphasizes more linking, not less. Even so, drawing out the arguments and posing potential resolution of the debate is far beyond the scope of this book. In one respect, the arguments favoring dynamic linking as subject to fair use seem persuasive, but the arguments concerning the distribution of libraries to sufficiently ensure the success of dynamic linking do not.

At any rate, linking issues should not be assumed resolved by reference to modifications or derivative works, generally. Instead, a clearly expressed and precisely drafted provision directed toward linking issues should be included in the license. Of particular concern for licensors should be the desire to be clear as to how licensees should incorporate modifications that are contributed to the project codebase to avoid linking matters that could bring the project itself within potential infringement liability of copyright holders of linked-to libraries. (See Chapter 2 on the LGPL.)

Finally, the license template uses the term *nonstandard executable* to mean any modified executable, wherein the modification is outside the scope of the license, or any executable added to the project that is intended to replace a preexisting executable included in the standard version. The intent of this provision appears to be to sustain consistently named files in the project. As a practical matter, this provision may restrict the types of changes or modifications that may be added to the project.

The license template ends by including a typical warranty disclaimer often used in open source licenses. As noted, it is not unusual for a software license of any type to disclaim the implied warranties. If so, those disclaimers must be conspicuous. If a version of UCITA sets the default governing rules for the license, it should be noted that UCITA's implied warranty on accuracy of informational content does not cover "published informational content, which could include multimedia encyclopedias, computer databases, or website press releases." Some argue that the exemption is fair for consumers, since it covers published content and furthers the value of free expression by generally encouraging the distribution of publicly available information.

Even so, it should be acknowledged that there are open-ended legal questions, since such exclusions would also include aspects of software interfaces and source code, which could create a great deal of obvious concern for open source users if the source code distributed was routinely inaccurate and this distribution was accorded no legal hardship.

Conversely, a licensor *expressly* warrants that information to be furnished under an agreement will conform to any affirmation of fact or promise, or description made by licensor to licensee in any manner, including in a medium for communication to the public, such as advertising that relates to the information and becomes part of the basis of the bargain. It is not necessary to use words like *warranty* or *guarantee;* however, a mere affirmation of the value of the information or statement purporting to be the licensor's opinion or commendation of the information does not create a warranty. This section does not alter the standards by which an express warranty for published informational content is either created or not created under other law.

Of course, we have already noted that whether current federal law preempts UCITA is far from clear. The question arises whether the U.S. Congress should enact a law explicitly regulating this area and preempting inconsistent state law. Federal law is potentially superior to state law if uniformity is desired. Gaps in state uniformity permit variations that potentially may lead to a race to the bottom in computer information transaction law. Not only might some states fail to adopt the uniform law or adopt a modified version, but some states may have weak laws that favor local sellers to the detriment of out-of-state buyers.

With respect to UCITA, federal mandatory laws may be inefficient. If Congress failed to enact a comprehensive law of computer information transactions, the result might be worse than a lack of uniformity of state law; instead, a poorly anchored federal law could lead to an incoherent patchwork of state *and* federal law.

Commercial Open Source Ventures

The best software is developed under conditions where software is freely distributed in a manner that permits any licensee to modify the software. Here, we explore what types of licenses and business models have been used to slightly modify the conception of open source in the context of commercial open source ventures. Commercial uses of open source licensing covers a wide-ranging variety of business models that can be successfully adapted to participate in open source projects in a straightforward and mutually beneficial manner.

Some of the most common commercial uses of open source licensing include open source vendors who did not develop the software they distribute or employ programmers who create or contribute to open source software; instead, these vendors help promote open source by using it, freely distributing it, marketing it, and supporting it while generating revenue from services or closely associated products at the same time.

Often, commercial open source models use dual-licensing objectives or adopt models that allow the copyright holder of the original project to retain greater control over what modifications are actually incorporated into the official codebase and releases of the software application (e.g., Netscape's initial foray into open source or Sun Microsystems' somewhat clumsy and controversial stance on open source regarding its JAVA programming system).

The most successful commercial open source project is probably a software program generally known as Linux, a version of the UNIX operating system. Linux is widely viewed as a much faster, less crash-prone (or malfunctioning) operating system than Microsoft's Windows operating systems. Its success might be measured by its growth and popularity among end-users (Linux-based operating systems represented 24.6% of all new license shipments of server operating systems in the late 1990s, and is now the most commonly used operating system for Web sites), but other indicators of success include the widespread support of the software among the programming community; a virtual army of programmers have freely enhanced the software system by adding their own refinements to it.

Participants in genuine commercial open source development efforts can generate revenue in a multiplicity of ways bounded only by the practical limits of an entrepreneur's imagination, including making open source products widely available with technical support; serving consumer interests in mass-market, conveniently available shrink-wrapped packaging with manuals through retail distributions; supporting sales of mobile telephone services and other wireless appliances; providing documentation, manuals, books, and other forms of support offerings for software; supporting the sales of hardware or other electronic devices, including desktop computers, personal or handheld devices, consumer robotics, device drivers, and embedded systems; supporting sales of Web services that rely upon open source software; strategic use of open source products as leads to sales for other unrelated services or products; and generating revenues from advertising or marketing services.

From AOL–Time Warner (Netscape) to IBM, Apple, and Sun Microsystems, examples abound of commercial uses of open source projects. The range of software licenses used to accomplish commercial distribution of open source software is similarly wide-ranging, although many commercial-use open source licensors have shown a remarkable preference for a close replica of the GNU GPL license template. Even database vendors have slowly begun to move away from proprietary development models and adopt open source instead. This trend is due, in large part, to the success of open source advocates who have convincingly demonstrated that open source software licensing is not anathema to commercial software distribution. Through its Linux Technology Center, for example, IBM is working within the open source community to improve the Linux operating system and to add the necessary enterprise capabilities that will support the electronic-commerce needs of business-to-business enterprises.

Hagen Software, a successful Internet software distributor and Web hosting company headquartered in Washington, D.C., distributes both proprietary software and open source software supporting electronic commerce. According to Philip Hagen, CEO of Hagen Software, his company, like similarly successful companies, has discovered how open source software may build a framework for a business model that supports services, sales, and other revenue-generating opportunities. Hagen Software, like most Internet businesses, uses Apache, the open source Web server package, and also uses the popular open source programming language Perl to develop Web hosting technologies. In that regard, it is quite apparent that for some commercial software developers, the decision to use open source software is a strategic choice that allows the developer to contribute to the open source community when a suitable project demonstrates that the overall licensing costs are lower and production more efficient than relying upon a proprietary software development model.

A commercial database distributor, Great Bridge, in Norfolk, Virginia, distributes a shrink-wrapped, boxed version, PostgreSQL, a database software package distributed along with vendor services and support offerings. The distribution sells for nearly $50,000, which, of course, is not anywhere near the meaning of *free*. But the database package is open source and the distribution is a very good example of a commercial use of open source software. Some vendors have discovered that licensees often care a great deal about support packages because some system software is simply to complex or costly to run with in-house technical support. Under those circumstances, a vendor can use the opportunity to educate the potential licensee on the advantages of open source software as well as demonstrate the significant cost savings on the development.

Sun Microsystems (Sun), a developer of commercial versions of the Java programming interface, is also associated within the open source community as a developer of an open Java software package. By teaming with the Apache Software Foundation (ASF), maker of the popular Apache Server, Sun is supporting Java supporters by allowing them to submit changes for the Java platform under open source licenses.

Sun Microsystems designed its wireless protocol, the JavaPhone API, to simplify telephony development, using an open platform, Java technology, as a core element of mobile telephony. The API is distributed under dual-licensing arrangements. The open source license includes terms such as the following:

> Sun Microsystems grants you ("Licensee") a non-exclusive, royalty free, license to use, modify and redistribute this software in source and binary code form, provided that i) this copyright notice and license appear on all copies of the software; and ii) Licensee does not utilize the software in a manner which is disparaging to Sun Microsystems.
>
> The software media is distributed on an "As Is" basis, without warranty. Neither the authors, the software developers nor Sun Microsystems make any representation, or warranty, either express or implied, with respect to the software programs, their quality, accuracy, or fitness for a specific purpose. Therefore, neither the authors, the software developers nor Sun Microsystems shall have any liability to you or any other person or entity with respect to any liability, loss, or damage caused or alleged to have been caused directly or indirectly by programs contained on the media. This includes, but is not limited to, interruption of service, loss of data, loss of classroom time, loss of consulting or anticipatory profits, or consequential damages from the use of these programs.

The initial paragraph of this open source license contains Sun's grant clause, which grants the licensee the right to use, modify, and redistribute the Java technology. Included in Sun's grant clause is an integrity rights provision that disapproves of any use of Sun's technology in a manner that may be harmful or disparaging to the licensor's reputation. Sun's support for open source reflects a significant acceptance of the vitality of the open source development model, since Sun's own development model for Java is viewed as distinguished with its own set of unique advantages.

Even so, open source companies looking to certify their products as Java-compatible through the Java Community Process (JCP) that governs Java's standardization had been reluctant to adopt Java or port works to the Java environment primarily because of licensing issues and confidentiality concerns. In response, Sun allows open source adherents to adopt modifications to the Java specification and obtain approval to include those modifications in a given project by submitting changes through the JCP, which may be subsequently licensed under an open source license. Sun contributes directly to open source by dedicating part of its support staff to help open source developers and academic institutions build new features for Java.

In addition, Sun's Star Office desktop software productivity suite is licensed by Sun to licensees at no charge. The package is available as a download from Sun's Web site, and seems to be strategically aimed to compete against other commercial office suites, particularly Microsoft's Office suite. One might quibble that Star Office is not pure open source software or fails to clearly promote interoperability in a manner that would truly provide a competitive choice for consumers of rival office applications, but Sun's distribution has demonstrated that there may be a viable commercial basis to build a complete open source desktop office suite.

This is a significant step for open source, since it was just a few years ago, in the 1990s, when it was widely believed that complex software development projects were beyond the reach of collaborative, but wide-ranging, open source development efforts. Since then, more developers have found confidence in the words of Eric Raymond, an executive board member of the OSI, that "given enough eyeballs, all bugs are shallow." Indeed, the understanding that when an open source project keeps the source code open, one of the many benefits that accrue is that anyone may freely inspect the code and improve it. With this kind of debugging algorithm, instead of keeping complex projects out of open source, the increased level of reliability and adaptability of the software should require systems builders to consider open source first.

No doubt it is easy to confuse commercial software distribution with open source distribution for commercial uses. There are anecdotal reports that some software vendors are intentionally using clever marketing to confuse consumers or erode the meaningfulness of the distinct software development models. Most commercial software licenses prescribe limitations on how the software may be used. These limitations may include restricting the use of the software to certain machines, a specified number of users, internal business purposes, and the like. Since open source turns traditional licensing concepts on their head by licensing the licensee more rather than less freedom, the distinction between an open source distribution permitting commercial use and a traditional commercial software distribution may not be apparent without consulting the actual terms of the license; more directly, one should not rely solely upon the marketing or promotion efforts of vendors to make this determination.

In the late 1990s, Netscape Corporation made a strategic decision to release its browser program, Netscape Navigator, then the most popular Internet browser, as an open source project. Of course, this decision may have had as much to do with competition from Microsoft as it did with a genuine belief by Netscape in open source software development. Netscape drafted its own open license to support at least two important objectives, namely, to cover both the existing licensed software within its program and to allow it to retain certain rights to release proprietary commercial versions of software developed as part of the proposed Netscape open source project. Although as a commercial open source product Netscape is beginning to show signs of life, it is also a case study in some of the difficulties in porting complex proprietary products to open source. For example, Netscape has been slow to overcome its difficulty in recruiting open source developers. This is particularly surprising; the date Netscape joined the open source community is perceived as a critical point in the evolution of the open source movement, and this importance should have accompanied thousands of developers joining the project to ensure its success.

Of course, the popularity of Netscape's Web browser among Internet users in 1998 meant expectations of imminent success as an open source product were exceedingly high for Netscape. More to the point, some of the developers who may have joined the project could have been dissuaded by the lengthy time it took to port a popular commercial proprietary product to open source. Added to programmer impatience with Netscape were at least two issues that slowed the progress of open source development: (1) shortly after Netscape's browser was ported to open source, copyright to the browser and other Netscape software was acquired by AOL, and (2) the role of Netscape's in-house programmers created uncertainty among open source programmers over how the two camps would be managed.

Within the open source community, Netscape's open source licensed is commonly referred to as the Mozilla Public License (MPL); which has become a popular open source license template. When Netscape decided to distribute its Web browser as open source software in 1998, Netscape had to overcome the licensing difficulties that accompanied the fact that its browser source code included code licensed under a number of distinct commercial licenses, some of which could impede licensing the code under the GNU GPL. Consequently, Netscape ultimately produced an additional license: the Netscape Public License.

The Netscape Public License covered code directed solely toward commercial or proprietary software, while the MPL covered open source licensing for commercial uses. Under the Netscape Public License, commercially licensing of derivative works is permitted. Changes to covered program source code must be made freely available to anyone. In addition, the Netscape Public License does not contain a copyleft provision. The license also uses the term *additions* rather than the less precise term *modifications* when referring to source code that forms a larger work and may be licensed differently and published or not even published at all.

Like the GPL, the MPL requires that licensees make additions freely and publicly available in source code form. Unlike the GPL, however, one interesting aspect of the MPL is that it allows developers to distribute their own files with *covered files* (any software or code that is covered by the MPL) under any licensing terms, provided that the developer has not modified the actual covered files; the licensee need not make the source code for a proprietary work available even if the work contains unmodified covered code.

If the licensee has modified the source code, then the work must be distributed under the MPL. Additions to the software include (1) changing anything within one of the files contained in the source code, (2) placing excerpts of source code from one of the files into a new file, or (3) renaming a file or combining two or more files contained in the source code. The MPL also contains several terms that protect the project and its developers against patent issues surrounding code that has been contributed to the project. For example, all contributors are required to release all "patent rights that may be exposed by the code." This clause may be intended to prevent or discourage open source participants from surreptitiously contributing patented code to a project wherein they subsequently attempt to extract patent fees for use and distribution of the software.

One interesting practice that Netscape adopted is the distribution of an electronic file that accompanies the software and the license, called *legal.txt*. This file is formatted as an ASCII text file that is readable by nearly any text editor. The file cannot be read until it is downloaded, and often it is viewed after the initial reading of the Web site license. Although the file need not contain text that materially alters the license or that adds additional terms, its presence or use could create unintended legal hurdles for both the licensee and the licensor. According to Netscape, the legal.txt file accompanies the MPL to help the licensee understand issues related to use of its code, and covers such matters as patent infringement liability, arbitration of claims, and notice of any included code under dispute.

The meaning of *derivative work* is clearly established as a term of art in the copyright statute, but use of the term *modification* is not. Hence, the use of the term *modification* as a licensing term introduces a peculiar risk that courts will view the term as having an unclear meaning. Of course, the MPL is not the only open source license that uses the term *modification* imprecisely. I think *modification* is used in a less precise manner than *derivative work* in some open source licenses, and, therefore, the use of the term is not helpful since a term or art exists to express a similar idea more precisely.

There is also the questionable premise that a software license may lawfully extinguish the floor and ceiling of derivative works; that is, under copyright law, some modifications need no permission from the copyright holder because they are fair uses, other modifications need no permission from the copyright holder because they are transformative, and somewhere between those two extremes are derivative works. To the extent that a software license uses the term *modification* in an attempt to capture what a court may determine to be a transformative work or to control what is or may be viewed as fair use, the license increases the likelihood that its terms may be deemed troublesome, to say the least. Open source licensors should substitute the term *modification* with *derivative work* when appropriate, and drafters should be very circumspect concerning the use of the term *modification*.

As noted previously, when considering commercial open source models, it is critically important to distinguish genuine open source methods from those that are masquerading as open source or are poor and unfaithful implementations of open source licensing. As a result of the recent media attention directed toward the open source community, there are increasing claims by commercial software developers to be open source adherents. Whether some of these developers intend to create confusion among end-users as to what actually constitutes open source or they actually intend to free-ride on the claims of open source as a marketing tactic, the fact remains that there are increasing attempts to capture open source licensing objectives.

Microsoft has initiated what it calls *Shared Source Licensing*. Under this framework, Microsoft is implementing what it considers a legitimate entry into the commercial open source licensing space. Its version of commercial open source licensing is apparently designed to enable increased customer and developer access to Microsoft source code, while preserving the intellectual property protections of its software business. This means Microsoft is willing to license its software and share its source code with those who desire source code access and promise not to share the code with anyone else. Not surprisingly, this model is known by many names, but it has never been considered an open source development model, which might be the point.

The Shared Source Licensing project was initiated in July 2001 and applied to some, but not all, of Microsoft's software projects, including one of the company's embedded operating systems—Windows CE (also known as Pocket PC 2000). Source code licensed under a so-called Shared Source License for Windows CE can be modified and redistributed in noncommercial usage scenarios and can be used for reference and debugging purposes in commercial scenarios. Microsoft has also extended the Shared Source Licensing initiative further with the Windows CE .NET product release, which includes nearly 1 million lines of code.

Interestingly, Microsoft and the United States entered into an agreement that resolved the outstanding issues in the antitrust litigation involving Microsoft's software licensing practices. The backdrop of the litigation centers upon Microsoft's practice of licensing its Windows software for multiyear periods on a per-processor basis. Presumably, to help prevent software piracy, Microsoft insisted that computer makers pay a royalty to Microsoft for each computer they shipped, whether or not Windows was installed as the operating system. In Microsoft's view, as a result of their physical presence, the computers were more reliably inventoried and accounted for than ephemeral and intangible copies of software.

The U.S. government, many states, and industry insiders, experts, and consumer groups were unconvinced. Instead, these groups rejected Microsoft's ostensible copyright-protection argument as a likely sham. These parties urged in court and the media that Microsoft exploited its dominant market position by insisting on unfair licensing arrangements.

As is conspicuous from the quoted provision below, paragraph III of the proposed agreement ostensibly regulates Microsoft's distribution of modifications to its operating system software. The company can no longer withhold those modifications in an attempt to punish a licensee or dislodge a potential competitor from the marketplace. Of course, if Microsoft distributed its operating system software under an authentic commercial open source license, there would be little need for this form of direct government intervention into the affairs of software development.

> III. Prohibited Conduct
>
> A. Microsoft shall not retaliate against an OEM by altering Microsoft's commercial relations with that OEM, or by withholding newly introduced forms of non-monetary Consideration (including but not limited to new versions of existing forms of non-monetary Consideration) from that OEM, because it is known to Microsoft that the OEM is or is contemplating:
>
> developing, distributing, promoting, using, selling, or licensing any software that competes with Microsoft Platform Software or any product or service that distributes or promotes any Non-Microsoft Middleware;
>
> shipping a Personal Computer that (a) includes both a Windows Operating System Product and a non-Microsoft Operating System, or (b) will boot with more than one Operating System; or
>
> exercising any of the options or alternatives provided for under this Final Judgment.
>
> Nothing in this provision shall prohibit Microsoft from enforcing any provision of any license with any OEM or any intellectual property right that is not inconsistent with this Final Judgment. Microsoft shall not terminate a Covered OEM's license for a Windows Operating System Product without having first given the Covered OEM written notice of the reasons for the proposed termination and not less than thirty days' opportunity to cure. Notwithstanding the foregoing, Microsoft shall have no obligation to provide such a termination notice and opportunity to cure to any Covered OEM that has received two or more such notices during the term of its Windows Operating System Product license.

Notably, the agreement relies upon a common understanding of the difficult-to-define term *software*. The term *software* includes application programs (like games and word processors) and system programs (like Windows and Java), which control the computer's hardware and provide a platform on which to run the applications. Apparently, Internet Explorer is software that is an intricate, elaborately embellished Web browser capable of standing alone.

In Microsoft's view, two products can be integrated even if they are not technically interdependent. The products need not function only in combination, nor be marketed only as a package. To be characterized as *integrated*, they just need to be combined in a manner that creates synergism—a whole that is better than the sum of its parts.

In our information-oriented technocentric capitalist economy, consumer preferences change quickly and technological obsolescence militates against noncompetitive or anticompetitive barriers, which should lead to courts enforcing most private contracts unless they are otherwise unlawful. It is at least arguable whether the government ought to be the arbiter of whether a computer buyer's self-interest is adversely affected by a software developer's contract terms. Undoubtedly, the role of arbiter forces the government to be inextricably shackled to the business of software development, which may ultimately prove to be a highly unsuitable form of governmental paternalism intermingled in open markets. Open source exposes, rather than undermines, the principle that genuinely free markets are sufficiently dynamic to ensure that market power cannot long subsist in the information technology industry [11].

Endnotes

[1] Controversial in the sense that the status of the license as a genuine open source license is highly disputed or in doubt.

[2] BSD is a short-hand reference to Berkeley Software Distribution, or BSD UNIX, an operating system software initially developed at the University of California at Berkeley.

[3] The BSD License grants licensees access to source code, rights of redistribution, and freedom to modify or develop derivative works, but there is no requirement that the redistribution occur under similar terms. There is no copyleft provision in the BSD License template.

[4] For example, it has been alleged that Internet protocol source code connected with the "FreeBSD" open source operating system is used in several places deeply hidden inside several versions of Microsoft's Windows software. In addition, and perhaps quite ironically for the leading proprietary software builder, some commentators have reported that Microsoft conceded that it currently uses or has used open source software to run the Web servers that manage its Hotmail e-mail service.

[5] Indeed, the free-riding that may be occurring could be enormous. The open source community itself may have little interest in policing the more deceptive forms of free-riding, since end-users and programmers would be discouraged even from participating in projects protected by the GNU GPL. It is noteworthy that even if open source adherents desired to strictly enforce the participation in open source projects or to adequately police the free-riding problem, that task has now been made far more difficult in the United States with the adoption of the Digital Millennium Copyright Act (DMCA), which authorizes technological measures to block access to source code without proof of copyright ownership as an element of a cause of action for circumvention of a technological control.

[6] As noted earlier, the restrictions of primary concern are those that accompany the copyleft provision, which essentially keeps the open source project an open source project.

[7] Depending upon the existence of contracts, the sources of funding, and the ownership of copyright, some academic research may not be considered immediately in the public domain.

[8] FreeBSD, NetBSD, and OpenBSD are different forks of the BSD codebase.

[9] The restriction goes to the use or, more precisely, the incorporation of the module in another work. In this context, *use* does not refer to dynamic linking or a use governed by section 117 of the Copyright Act; those distinct issues are addressed later in this chapter and in Chapter 2, which addresses the use of the GNU LGPL.

[10] The open source community is currently debating the need to develop an open technology standards policy or model as a preliminary bulwark to the overprotection of technology by patent law.

[11] An example of a proprietary software license:

Microsoft Windows 98

Systems Software: "You may install and use one copy of the SOFTWARE PRODUCT on a single computer, including a workstation, terminal or other digital electronic device ("COMPUTER"). If the SOFTWARE PRODUCT includes functionality that enables your single COMPUTER to act as a network server, any number of COMPUTERS may access or otherwise utilize the basic network services of that server. The basic network services, if available, are more fully described in the printed 8241*922 materials or electronic documentation accompanying the SOFTWARE PRODUCT."

Multiple Monitors: "If the SOFTWARE PRODUCT includes functionality that enables your COMPUTER to make use of additional displays such as additional monitors or a television: (i) any additional display must be physically and directly connected to your COMPUTER and (ii) your COMPUTER must be the only source of inputs utilized by the SOFTWARE PRODUCT."

4

About Protecting Property

Avoid the FUD Factor: fear from uncertainty has caused some to view the benefits of open source as doubtful, but in Cyberspace uncertainty is a reason for Being.

Copyright as Property

Fear, uncertainty, and doubt (FUD) spreads rapidly when those who oppose the adoption of a new or competing technology spend vast sums of money persuading others to enact restrictions on the use of the new technology or to outlaw use of the new technology entirely in order to thwart the undesired competition. The open source software development model is frequently attacked by those who habitually spread FUD.

The myths and perceived fears of what might become of open source software are legion, and many of the myths exist as a result of the intentional spread of FUD by software developers who fear the likely success of the open source community in dislodging their market dominance. In other instances, FUD has been used to reconstruct conceptions about normative behavior. For example, copyright holders might attempt to dissuade software users from exercising rights of fair use by urging that all uses of software come within the scope of the copyright holder's rights and, therefore, uses of a software program that have not been explicitly granted are deemed illicit uses regardless of traditional conceptions of fair use. In this regard, FUD has been used to transform both fair and illicit uses into claims of theft. In fact, many software users are likely to view some forms of copying of copyright-protected works without explicit permission as theft, rather than more precisely as copyright infringement or fair use.

How has the use of FUD by copyright holders been so successful? One factor has been the increasing appeal by many of viewing copyright as property right. In this chapter, a number of myths or uncertainties about the ownership model for software used by the open source community are addressed and exposed by demonstrating how the current default legal rules actually support and reinforce the fundamental appropriateness of using the open source business model in the context of software development.

Methods of Propertizing Copyright

In 1997, the U.S. Congress, for the first time, enacted a law, the No Electronic Theft (NET) Act, which punishes copyright infringement in the context of software or computers without the requirement that the government prove that the alleged infringer had intent to benefit or profit in a personal way from the alleged copyright infringement. Although the enactment of this law could reflect congressional wisdom concerning the need for tough criminal penalties to stem the increasing perception that content users had become disrespectful of copyright interests, the enactment of a new, tough criminal statute also coincides with the increasing acceptability of the view that copyright interests were coextensive with property rights. Since criminal laws already protect property owners from criminal conduct, few found it troubling when Congress expanded or strengthened the scope of federal criminal law to include conduct ostensibly involving copying.

Many would argue that copying can be sensibly punished through criminal penalties if illicit copying (or copyright infringement) is viewed as theft rather than simply the debatable exercise of a fair use right; in this light, few would urge that theft of property should not be discouraged through enforcement of criminal laws. To arrive at our current conception of the acceptable use of criminal sanctions for copying, however, popular conceptions of what *copy* means required progressively more serious conduct so that copyright would be viewed as property and copyright infringement as theft. Although this transformation of reality was accomplished before Congress adopted the NET Act, the NET Act seems to signal that it is no longer controversial to impose significant criminal penalties for copying.

Setting Criminal Penalties for Theft of Copyright

Even as a matter of legal theory, it had not been the case that copyright and property were viewed always as conjoined in the law of copyright. Indeed, the Copyright Act explicitly distinguishes between a copyright interest and the embodied personal property that constitutes or is used to display or distribute the copyright-protected material; namely, copyright law governs the exclusive interests of the copyright holder and is not, generally, thought to be a federal law of personal property.

Even so, Congress seems to have given considerable weight to the view of copyright as ostensibly a property right when it significantly strengthened the criminal copyright law. In doing so, the government no longer must prove that an alleged infringer acted for purposes of commercial advantage or private financial gain, or show that the alleged infringer actually realized a commercial advantage or financial gain. Internet users should be aware that Congress has removed what may have been, aside from the practical resources of law enforcement, the only critical bar to zealous prosecution of willful copyright infringement. To some extent, law enforcement already has shown a slight reluctance to bring Internet-based criminal copyright infringement cases against individual Internet users. Aside from that being a waste of limited resources, law enforcement may be reluctant to bring cases under the NET Act where the infringement arises from novel uses of technology.

When Congress decided to recalibrate the penalties for criminal copyright infringement, it placed nearly all copyright infringements within the reach of criminal punishment; those who willfully infringe copyright by reproduction or distribution of a single copy (or more) of a copyright-protected work by electronic means, during any 180-day period, with a total retail value exceeding $1,000 will be subject to criminal punishment.

What this really means is that today only the very casual copyright infringer, if casual is $1,000 or less, is not subject to criminal penalties for violating copyright in the United States.

The increasing acceptability of imposing criminal sanctions for unauthorized copying reflects more than an attempt to discourage infringement (to wit, that purpose is duly fulfilled by civil liability copyright); instead, it likely reflects, at least in part, a significant shift in our conceptual view of copyright in the direction toward propertizing copyright interests.

The Impact of the Internet on Conceptions of Copyright

The move toward propertizing copyright interests strongly reflects the debate over the impact of digital technologies and the Internet upon copyright. The debate concerns whether the ill-fated future—each side of this debate views the unfolding of the other side's prediction as ill-fated—of the scope of the relationship between copyright and digital works will include the elimination of copyright (or the need for copyright) or an evolution of ubiquitous, perpetual copyright. Without picking sides, it is clear that open source responds to the debate by recognizing that when Congress legislates an expansion of copyright (in terms of scope or subject matter), it ostensibly imposes a tax on content users; in doing so, consequently, Congress ought to exercise such power with considerable circumspection.

Returning to the FUD factor surrounding contemporary rhetoric about copyright and cyberspace, a survey of past practices discloses that fear is often spawned from those vested in current technologies when new technologies appear to threaten old ways of doing business or producing content. Indeed, although copyright holders are always quick to see doom in the context of new technologies applied to their business models, Congress should recognize when these doomsday predictions are the howls of crying wolves.

Not long ago, for example, during the congressional hearings held in response to the U.S. Supreme Court's fair use analysis in the *Universal City Studios, Inc. v. Sony* decision, the Motion Picture Association of America (MPAA) beseeched Congress to provide legal authorization for a licensing scheme to compensate Hollywood for the Court's decision. Among its many claims, the MPAA argued that fair use, in-home videotaping of audiovisual broadcast programs would inevitably result in a barren wasteland of uninspired and unentertaining television programs. It is not entirely clear whether Hollywood believed that its request for a royalty arrangement, which Congress granted, would alleviate what was inevitable or just make the painful reality less painful; quibble, if you must, but the MPAA may have been demonstrating proof of exceptional perspicacity rather than simply crying wolf in this instance! Even so, the tactic worked. And, perhaps, as a result, it has been a consistent response to the appearance of new technology that may deliver audio or audiovisual works in a new style, mode, or technological forum.

Software and the Public Domain

One irony of the increasing propertization of copyright is that the inclusion of software seems to undermine the very case for property. Of course, new technologies, like those produced to support the Internet's infrastructure, can produce gains for the public domain as well as provide greater opportunity for potential copyright holders to produce original works. In other words, without expanding copyright to the point at which it impersonates a property right, copyright holders and the public could obtain benefits from supporting the vitality of the public domain, to which the Internet itself stands out as an example.

For literary works that do not constitute software, once the work is published, the ideas contained in the work become apparent and conspicuous. In this manner, the work's basic ideas may provide a basis for further development of additional works by any member of the public with access to the work. This is a critical function of the public domain because it serves the primary purpose of copyright. Ironically, the relationship between the public domain and open access to ideas is quite like a recursive function in a computer program, wherein a call to a procedure may result in a procedure repeating itself indefinitely. The enrichment of the public domain of ideas increases the public's access to works, which further enriches the public domain. Software, however, presents a slight conundrum that disrupts the flow of the recursive enrichment of the public domain served by copyright law.

Software is executed or is run on general-purpose computers in a form that often hides the source code from all but the copyright holder. Although graphical user interfaces have enabled technologically savvy end-users greater access to the ideas embedded within software, many ideas that can be discerned from software are woven into the source code. The publication or distribution of software does not alter this result. Hence, the complete bargain ostensibly accomplished by the government's grant of copyright is unfulfilled by software developers who either block access or withhold access to the ideas embedded within source code from end-users through a number of methods, including technological barriers, hardware locks, licensing restrictions, and claims that reverse engineering software is unsupportable under copyright law.

When Source Code Lacks Originality

Is computer programming a highly creative and individualistic endeavor? Do programs consist predominantly of commonly known techniques and materials strung together with skill, but without significant originality? And what if a program has become so popular as to set a de facto standard, such that users come to expect its keystrokes and command structures to be present in any program designed to accomplish the same functionality? Leaving that precise debate for a different context, the common conception that an individual programmer develops software has yielded to our understanding that software development often requires development teams. Despite increasing clarity with regard to how software is actually produced, the debate concerning whether software development is much more science or engineering than art is ostensibly unresolved by law. You might say that the laws of patents and copyrights provide highly contrasting answers, which in some respect may disclose how far from resolved the matter is. Even within the circles of computer science, there is no clearer answer regarding how best to classify the craft of programming than to show that the recent adoption of eXtreme Programming as a genuine form of software development supports the view of programming as at least a social process that is a somewhat quasicreative endeavor.

What we know with a degree of greater certainty is that as cyberspace has become an increasingly useful environment for computer programming, programmers are developing a reluctance to systematically hide their source code from each other. Instead, programming is undergoing a paradigm shift away from viewing source code as the province of secrecy and toward sharing source code in the form of reusable modules or objects.

In an attempt to build more complex applications that can run on desktop computers and local networks, programmers routinely share and exchange code that can be used and reused for various projects. In this regard, source code modules are often distributed as works in the public domain. Therefore, it is doubtful that the arguments supporting reexamining the scope of copyright protection for software also undermine the arguments against the propertizing of copyright interests that apply to software.

Instead, the point is that public domain software exists, and highly efficient programmers often build existing public domain software into their works. In this respect, the conduct of programmers seems resistant to the public policy choices advocated by many within the information technology industry, who still insist that copyright be viewed appropriately as a propertylike right. Still, given the vast likelihood that any particular software program contains significant portions of unprotectable material—in some instances so significant that the program itself may be unprotectable—it strains logic and commonsense to believe that the case has been fully established that software programs ought to obtain the type of robust copyright protection commensurate with what we usually only accord for genuine property rights.

Open source represents a persuasive counterargument to the arguments favoring the propertizing of copyright interests; the very existence of the open source community bears the substantial support of a public domain–enhancing, nonpropertycentric conception of copyright. The use of open source software in specific contexts—such as those involving electronic government—could also directly enhance the public domain.

When governments as market participants—or in support of pertinent electronic government (e-government) initiatives—fulfill information technology needs by deploying open source software, the public domain of software is necessarily enhanced, which empowers innovation and further development software for new and different uses without high barriers to entry. As governments throughout the world make increasing improvements in the manner in which they implement and manage information infrastructures through electronic or digital means, they will slowly replace the outdated and inefficient methods of the analog world.

Computer databases are replacing file cabinets, e-mail (in certain contexts) already has replaced printed memoranda, digital signatures will soon replace hand-signed documents, and interactive on-line forms are slowly replacing paper forms that required a typewriter to complete. Very soon government will operate in a digital world. Consequently, the opportunities to fuel the public domain may be enormous. And while there is no reason for governments to abandon proprietary software development entirely, similarly, there is no reason to foreclose the opportunities offered through open source by failure to incorporate the open source model as well.

Not surprisingly, more than a couple of governments have discovered that open source software provides an opportunity to catch up with other countries who have advanced information infrastructures; those governments face unique challenges in using e-government to leverage their information infrastructure for multiple purposes. If students learn to program by using open source software because schools in a certain location cannot afford to buy costly site licenses or multiple upgrades of the same proprietary product each year, then it makes sense for those governments to leverage the knowledge of their students when they become workers by using the same open source software in the information infrastructure that schools use to teach programming, software engineering, and even open source productivity software for word processing and spreadsheets.

In the United States, there is significant opposition to using open source in government, regardless of the benefits gained. Despite the doomsday predictions of those who oppose open source software development, adopting open source development in government certainly need not result in exclusive use of open source software. Recognizing that governments exert deep and heavy purchasing power in the information technology marketplace, procurement officials should always use caution and exercise good judgment in determining the most beneficial mix of software development models.

By this point, it is apparent that a common theme here is that open source software development is not simply an alternative way to build a better program; instead, a bolder theme is being urged, namely, that open source software development fixes something that is broken—our scheme of intellectual property. Why in our scheme of intellectual property, you might ask, should we treat software differently than other works deserving protection under our copyright scheme? Good question. We ought to consider the bigger issue here, namely, whether (in view of the underlying principles of copyright law) one may sensibly or reasonably draw a pertinent distinction between the intellectual property developed by, for instance, a writer like the ancient Greek poet Homer (who authored the classic poems *The Iliad* and *The Odyssey*) or even a modern writer like Maya Angelou, and the intellectual property created by an individual software creator—be it a software production company like Microsoft or even Linus Torvald, the author of the original source code of the kernel for Linux.

In other words, should the rights granted to authors of code be different than those rights granted to authors of traditional creative works? When the law of copyright protects poets like Homer or writers like Maya Angelou, the protection exists along with the public's access to the information and ideas within the work. When you read a poem or a novel, you have access not only to the expression of the poem (which may include the words and sentences and idiomatic spelling of words, and, perhaps, how the words sound when audibly expressed), but the reader also has access to all of the ideas in the poem, including the meaning of what is expressed as well as the poem's grammar, style, syntax, and number of lines for each stanza; in other words, you have access to all of the methods and tools of the poem that could enable you as a reader to use tools to create your own poem (not a derivative, under current law, but nonetheless you may create an original work of your own). This distinction is critical because it draws out how the prevailing legal regime has balanced the competing interests of increasing the public domain of knowledge or information against the utility of providing a government grant of copyright exclusivity to copyright holders for traditional creative works and why that balance is undermined when the same legal regime is adopted for software. The goals of copyright are not advanced in the same manner when applied to software today. To some extent, the use of open source in some e-government projects might be viewed as a suitable way to redeploy software to serve its important function in enhancing the public domain and the public's access to a body of works that will necessarily lead to the creation of more works, not less.

On the other hand, proprietary software is often distributed as compiled code that is essentially unreadable, the code is closed off from the end-user, closed off from other software authors, closed off from government, and frequently closed off from licensees. Why is access to the source code denied in the context of copyright? Is not source code one of the clearly expressive aspects of software works? There is no inherent essence or principle of copyright that would require closed code. Indeed, the very question of whether aspects of a software program are or are not copyrightable often depends upon what is in the source code. The ideas underlying the software program are primarily in the source code.

By closing off the source code in the name of copyright, some software developers have been able to hide their ideas. Imagine a traditional creative author hiding the ideas, but disclosing the expression [1]. Is it even possible in a novel, a poem, a lyric? The frequent response to critics of the current copyright regime as it applies to software is that the present system is fine, since there have been billions of dollars and thousands of jobs created through software development over the last couple of decades, which has provided significant benefits to the public in return for letting software developers retain copyright control over their code. Moreover, even if one concedes that open source has its place, it does not follow that open source necessarily fits the needs of government.

There is no dispute that in the United States, in particular, the software industry has been able to produce great products and spur tremendous gains in the productivity of American business, and if that were all that mattered in copyright law, there is no debate. But copyright serves multiple purposes, including how best to achieve the public's access to ideas and thereby induce growth of the public domain. Similarly, open source values openness and access to information. In the context of software, information is in the source code and that is the target of the condition of free access—freely accessed information.

Endnote

[1] Closing source code to conceal a trade secret might serve to justify hiding ideas in the context of trade secrets, but that matter has no relevance to copyright.

5

Electronic Contract Formation

A longstanding principle of contract formation is that the offeror is the master of his offer. With few exceptions, the offeror may prescribe as many restrictions, limitations, cannots, shallnots, or conditions as he desires. This principle is considered one that well-serves the efficiencies of contract formation, since the offeree need not say "I accept."

Electronic Transactions

Are open source licenses enforceable as standard-form electronic contracts? The answer depends upon two general matters: the principles of contract formation and the applicability of commercial codes to software licensing. Implicitly, software licensing raises commercial contract law concepts such as contract formation, creation and disclaimer of warranties, measuring and limiting damages, basic contractual obligations, contractual background rules, the effect of contractual choice, and risk of loss.

Open source software licenses are often in electronic format as well as drafted as standard-form agreements. In other words, the terms of most open source licenses are rarely actually negotiated between licensor and licensee; instead, the licensor or open source copyright holder drafts the terms of the agreement, seeks approval, perhaps, of the license through the OSI, and posts the license on an applicable Web site. Electronic contract codes recognize the commercial necessity of enforcing standard-form agreements as mass-market transactions, despite the fact that the terms of such forms may not be available to the licensee prior to the commencement or completion of the transaction. To this extent, Web site software licenses are likely enforceable as form contracts under UCITA or any other commercial code. Even limitations in a standard form license, such as terms that prohibit the licensee from making multiple copies, that prohibit the licensee or others from using the information for commercial purposes, that limit the number of users authorized to access the information, or that prohibit the modification of software or informational content without the licensor's permission are typically enforceable.

Of course, it is highly unlikely that any software license could validly prohibit licensees from quoting limited material for purposes of education or criticism, preclude a nonprofit library licensee from making an archival (backup) copy, or prohibit persons from observing the visible characteristics of software and thereby reverse engineer the program's operations to develop noninfringing products. In other words, in the context of conducting transactions in information products or software, neither the use of software licenses nor the reliance upon contract law or commercial codes is likely to discourage courts from using federal copyright law as the backdrop, if not controlling, guidance on what types of limitations on the rights of owners of information ordinarily seem appropriate.

UCITA is a uniform contract law designed to address the perceived needs of the new information economy, and seems to accept that licensing has become a dominant way for distributing digital information in the marketplace. UCITA is intended to set the default rules for electronic contracts, which arguably will include open source software licensing transactions. In this regard, licensing is one way of acknowledging the usefulness of contracts in conducting transactions in computer information and information products and services. Contracts underlie both the creation and distribution of such information. However, legal rules that are not relevant to commercial practice or that are uncertain in application inhibit contracting or raise transaction costs. UCITA was drafted in response to this fundamental economic change and need for clarity in the law.

The crux of any mass-market public software license enforceability analysis lies in the application of the relevant contract doctrine to the license. The contract analysis proceeds with two prongs: first, scrutiny of the procedural aspects of the license, and second, review of the substantive terms. As noted below, the ProCD case and the proposed uniform law—UCITA—require that to be found procedurally acceptable, mass-market licensees must give the consumer three things: proper notice of the license before purchase, adequate time to review and decide whether to assent to the license's terms, and the opportunity to return the software for a full refund if the license is unacceptable.

Mass-market public software licenses typically, and those discussed here specifically, include provisions that fulfill each of these requirements. The GNU GPL requires that notice of the license be given to the licensee and it states that assent to the license results from the licensee performing specific acts. As such, the GPL seems procedurally enforceable.

In the context of computer programming and software development, there is no agreement on whether the distribution of software constitutes a sale of computer software, the execution of a contract, or only the grant of a copyright interest through software licensing. Regardless of whether the software development model is open source or proprietary, the default rules used to resolve disputes that arise from software distribution will come from one of the three areas mentioned. It is not entirely improbable that a court might determine that some open source licenses constitute electronic licensing transactions. As discussed more fully below, UCITA, the Uniform Electronic Transactions Act (UETA), and the Electronic Signatures in Global and National Commerce Act (E-Sign) [1] play a major role in electronic commerce. While many issues are still unclear, the three laws provide greater stability and uniformity with regard to electronic contracting. Many states have been reluctant to adopt UCITA and have gone so far as to enact anti-UCITA statutes. UETA has been more warmly received, as over 35 states and the District of Columbia have adopted UETA or a law based on UETA. E-Sign will likely continue to provide a framework for those states that have not yet adopted UETA or consistent legislation.

Some open source licenses have the hallmarks of electronic contracts because they provide terms that touch upon matters outside the scope of copyright law and take the form of Web site agreements that are not entered into through negotiations between the parties prior to the commencement and completion of the transaction. Whether an open source software transaction may appropriately be characterized as a purchase of software raises further difficulty in setting the default rules, but new electronic commerce laws might render the sale question impertinent. Consequently, it is critical for open source developers to raise their awareness of the default rules that matter in the world of electronic contract formation. A computer software acquisition typically provides the buyer with the tangible media that contains the object code of the computer software, but does not involve the transfer of title to the computer software.

Since it has long been established that an individual may be bound to the terms of a license perceived only aurally, it might not strike one as odd that an individual could become bound by the terms of a license existing in electronic form only. What may be perplexing, however, is how one should express consent to an electronic license. What happens, for instance, when the power source is turned off and the electronic license disappears? Who has witnessed your consent? In the real world, an individual may present an offer orally and thereby witness your consent personally. How might this be accomplished in a virtual world, where identification, authentication, and repudiation pose inherent difficulties for contract formation? More to the point, do the rules of electronic contract formation apply to open source software licenses in the first place? Some would say no.

As addressed in earlier chapters, some proponents of open source software development dispute whether software licenses are contractual rather than simply unilateral expressions of a copyright grant. For those who adopt the approach of denying the contractual nature of open source software licenses, the issues of electronic contract formation are not germane issues. But for those more cautious, or at least less certain, of how a court may construe an open source software license, there are benefits to ensuring that you draft your software license with the rules of electronic contract formation in mind.

In this chapter, we explore the legal questions that may arise from binding licensees to software licenses that are posted on Web sites or that are otherwise in electronic format. In this respect, it is assumed that open source electronic licenses are subject to commercial laws intended to regulate electronic transactions. Many open source participants eschew any possibility that their licenses might be governed by any of the electronic commerce legislation such as UCITA, UETA, or E-Sign. UCITA is a model uniform law for computer information transactions, such as software licenses. UETA, also a model uniform law, deals with electronic signatures, records, and agents. UCITA and UETA differ primarily in that UETA is basically a procedural law for electronic transactions, while UCITA is a substantive law governing contracts for the acquisition of computer information.

E-Sign is a procedural law for electronic transactions and is similar to UETA. E-Sign basically preempts state electronic transactions laws regarding commercial or business transactions in or affecting interstate commerce that are inconsistent with UETA. States may only supersede E-Sign by adopting UETA or adopting another statute or regulation that is consistent with Acts I and II of E-Sign, does not favor a specific technology (with some exceptions), and, if enacted after E-Sign, makes specific reference to E-Sign. While no one within the open source community would argue that compliance with electronic commerce legislation explicitly undermines compliance with open source goals, there is considerable concern over whether electronic contracting rules might undermine the objectives of open source licensing.

In *Specht v. Netscape Communications Corp. and America Online, Inc.*, 150 F.Supp.2d 585 (S.D.N.Y. 2001), the court refused to enforce an arbitration clause because Internet Web site visitors did not assent to the putative license agreement which contained the arbitration provision. The defendant offered its "SmartDownload" software free of charge to Internet users of its Web site. Users merely had to click their mouse in a box to receive the download. The only reference to the putative license agreement appeared in text that was visible only if the user scrolled down through the page to the next screen. Users were not required to affirmatively indicate their assent to the license agreement, or even view the agreement, before proceeding with the software download. In applying California law to the issue of whether a contract was formed, the court held that the license presented by the defendants in this case was more of a browser-wrap than a click-wrap or shrink-wrap license, the downloading of the software by the plaintiffs did not indicate assent, and no contract was formed. The court appears to be highlighting factors that distinguish browser-wrap from click-wrap, since other courts have generally viewed browser-wrap contracts as lacking strong indicia of mutual assent.

If the Web site appears to have merely set up a way to download software and provided potential downloaders with a notice (that is not hyperlinked) to read the license, then a court may view the notice as an invitation to read a license (browser-wrap) rather than conclude that there is sufficient indicia of mutual assent (e.g., click-wrap). To be clear, no court has required a set of dialog boxes or buttons with "I accept" or "I do not accept." Rather, the point is that the Web site that seeks potential user input (e.g., clicking buttons, pulling down menu items) may strengthen the licensor's claim that the contract or license is binding upon the parties because contract formation principles have been followed, and the court may infer that the license was read and that the licensee agreed to the terms.

A Massachusetts court explained that the forum selection clause in this case, which was contained in a click-wrap agreement, should not apply because the harm and damage to the plaintiffs' computers caused by software distributed by the defendants occurred before the parties entered into the contractual relationship. Although the court did acknowledge that a reasonable forum selection clause is generally enforceable, in this case plaintiffs did not have notice of the forum selection clause before the installation of the defendant's software that damaged plaintiffs' computers because the actual language of the "Terms of Service" agreement was not presented on the computer screen unless a customer specifically requested it twice by overriding the default "I Agree" choice.

Why is mutual assent important? The obvious reason is readily apparent—particularly if implementing a routine that establishes an opportunity to express mutual assent imposes a minimal burden upon the electronic license drafter. Licensing is a more efficient way to transact business if the parties can express agreement to the terms of the license. For the open source software developer who lacks deep pockets, there are other incentives to establish a mechanism for expressing mutual assent, namely, shifting risk of loss effectively.

Many, if not most, open source software licenses include terms that shift risk of loss onto the licensee. This risk shifting is troublesome enough when open source software is experimental (or beta), but a court may find such risk shifting impermissible (or otherwise unconscionable) under circumstances where the terms of the license (including the risk of loss provisions) are not disclosed prior to completion of the transaction (such as displaying the license for the first time in a conspicuous manner *after* the putative licensee downloads and installs the software). If a copy of the open source software license is not available in a manner permitting an opportunity for review by the licensee *before* the licensee becomes obligated or bound to the terms of the license and the licensee does not subsequently agree to the terms or explicitly fails to manifest assent to the license after having an opportunity to review the license, the licensor may be at the mercy of an angry licensee.

Under such circumstances, it is not entirely unlikely that the licensor could be forced to consider whether it is appropriate to reimburse any reasonable expenses incurred in complying with the licensor's instructions for returning or destroying the computer software or, in the absence of instructions, provide for reasonable expenses incurred for return of the software, and provide compensation for any reasonable and foreseeable costs of restoring the licensee's information processing system to reverse changes in the system caused by the installation, if the installation occurs because information must be installed to enable review of the license, and the installation alters the system or information in it but does not restore the system or information after removal of the installed information because the licensee rejected the license. Consequently, open source licensors have multiple reasons to give contract formation considerable attention, including matters of self-interest. If a licensee agrees to be bound by the terms of a software license, a court may, nonetheless, invalidate unconscionable terms or terms against fundamental public policy under rules that apply to all contracts as provided for under the doctrine of unconscionability, which invalidates terms that are bizarre or oppressive and hidden in boilerplate language.

During the 1990s, the existence of numerous legal questions concerning electronic contract formation led many to doubt the validity of electronic contracts. Although many of these problems have been nearly resolved by courts and the enactment of new state and federal laws, since electronic commerce seems to be an implacably growing commercial undertaking, important legal questions remain unresolved. Consequently, electronic contracting must be undertaken carefully to avoid costly errors in this new domain of contract formation. Of course, electronic commerce did not give birth to electronic contracting, but the breadth of electronic contracting has been enlarged significantly by the commercial activity on the Internet and by public policies adopted to promote electronic commerce.

To ensure that a license is binding, there are a few critical conventions or procedures to follow. The license terms should specify the property, interests, or rights granted by the license. The license should also identify the parties or legal entities involved, indicate the obligations of the parties, demonstrate that there is a mutuality of promises (a "meeting of the minds"), and evidence the acceptable form of consent that provides proof that the parties agreed to the terms of license. Given these conditions, most licensors should conclude that a written license is preferable to an oral agreement; indeed, in some instances, a written license is required. In addition to being in writing, some licenses may require other formalities as well, such as recordation or notarized signatures. For example, an exclusive license transferring copyright must be in writing and a copy of the license should be filed with the U.S. Copyright Office in Washington, D.C.

In July 1997, President Clinton directed the U.S. Department of Commerce to remedy the lack of a predictable legal environment governing e-commerce transactions by developing a uniform commercial legal framework that recognizes, facilitates, and enforces electronic transactions globally. Commerce Department officials and White House representatives lobbied the U.S. Congress, and their audacious task culminated in self-described success and the enactment of E-Sign. Most provisions of E-Sign took effect on October 1, 2000, which was well ahead of the full implementation of a comparable state law initiative that had begun before E-Sign was drafted. In the spirit of the moment, President Clinton signed the bill into law with a smart card (a credit-card-sized identification device).

E-Sign is federal legislation that is supposed to address concerns over the purported lack of uniformity of state laws governing electronic commerce. It is worth noting, however, that in the United States, state governments have traditionally established the legal rules governing contracts and commercial transactions. Through an organization of legal experts and state law officials called the National Conference of Commissioners on Uniform State Laws (NCCUSL), states traditionally deliberate on developing legal codes that should uniformly resolve disputes arising within states that involve commercial transactions. Consistent with this practice, in July 1999, NCCUSL approved a model uniform e-commerce law commonly identified as UETA. UETA has been sent to all state governments for adoption and has been enacted in nearly half the states.

Similarly, UCITA was promulgated in July 1999 as a uniform commercial code for the states. As noted later in this chapter, UCITA is in some respects revolutionary in tone and, hence, has met significant opposition in state legislatures, most of whom have failed to enact it. Even if UCITA is enacted by a majority of states, it is doubtful that this legislation could be appropriately referred to as uniform. As it stands now, most states are likely to alter or ignore some of UCITA's most controversial provisions.

UCITA, unlike UETA, is a substantive contract law statute that addresses computer information transactions. UETA is largely procedural in emphasis and does not alter traditional substantive contract law doctrine. This alphabet soup of federal and state legislation will inevitably influence the dominant methods of electronic contract formation. In the following sections, we will set out the prevailing view on how these new laws may affect software licensing.

E-Sign

On October 1, 2000, E-Sign became law in the United States. E-Sign is a federal law that provides that electronic signatures have the same legal validity as a signature on paper or in print. E-Sign is technology-neutral. It does not endorse one form of electronic signature as more authoritative or reliable than any other.

Technology-neutral laws are viewed favorably because they do not create artificial preferences for certain technologies or unduly interfere with market forces. Consequently, an electronic signature may include a range of electronic equivalents from an encryption-based digital signature to a literal digital or scanned copy of your signature on paper. While E-Sign relies upon a rather amorphous statutory description of electronic signature, the statute's lack of clarity should not prevent the significant benefits that the law is expected to bring to the growth of electronic commerce.

To some extent, E-Sign is modeled after UETA, which was finalized by the National Conference of Commissioners on Uniform State Laws in 1999 and has been adopted by at least 22 states since then. Generally, since E-Sign and UETA are technology-neutral, they leave virtually unchanged the substantive law governing commercial transactions.

Both E-Sign and UETA were enacted ostensibly to help stimulate e-commerce by removing real and perceived obstacles to the use of electronic contracts in commercial transactions; this is particularly true with regard to traditional contract and commercial law doctrine that favors or requires the use of a written contract to obtain legal enforcement of certain transactions, including federal consumer warranty disclaimer disclosure requirements or the applicable statute of frauds requirement (which requires a traditional written form for a valid contract for the sale of goods that cannot be performed within a statutory time period or that exceeds a statutory amount).

E-Sign stimulates e-commerce by declaring that no one may claim that an electronic contract or electronic signature renders a license unenforceable. E-Sign makes the use of electronic licenses a viable way to conduct commerce. Disputes arising under electronic contract formation need no longer include questions concerning the enforceability of an electronic license; removing that barrier, however, does not mean that a party cannot challenge the enforceability of an electronic software license on the basis of contract formation, negotiability, unconscionability, or any ground other than the fact that the license is in electronic form. If the purpose of E-Sign is beginning to sound confusing, you might want to recall that before E-Sign, some licenses and contracts were *required* to appear in printed form to be enforceable. This archaic constraint is no longer relevant under E-Sign.

In order to create an electronic equivalent of negotiability, both E-Sign and UETA recognize a new category of contract: a transferable record. A transferable record is the electronic analog to a negotiable instrument or document. An instrument embodies the right to payment of money; a document embodies rights to goods. Negotiability is created when an instrument or document is drawn up in accordance with the formal requirements of negotiability, the most important of which is that the contract rights be transferable to the order of a named party or to the bearer of the contract.

Few computer systems in use today provide the rigorous security procedures necessary to meet the requirements of the transferable record control provisions in E-Sign and UETA. These provisions are technology-neutral, however, and a competitive marketplace should quickly develop among providers of such services. A system capable of meeting the transferable-record control requirements is likely to rely on advanced computer security technologies such as cryptography, but it is also likely to rely on business policies and procedures administered by people.

The transferable-record control requirements of E-Sign and UETA may strike many lawyers and information technology professionals as odd, or even incomprehensible. Most modern business information systems do not try to re-create something approximating the possession of a piece of paper as a system for tracking ownership, relying instead on customer account systems or registry systems.

Software developers or computer programmers familiar with conventional computer security principles may believe that the use of sophisticated security technology such as digital signatures could be adequate to meet transferable-record control requirements, but this is incorrect. Digital signatures can guarantee the authenticity of signatures and the integrity of the contents of a transferable record, but unless combined with strong access controls, they are not sufficient to produce an authoritative copy of the transferable record.

By creating a new legal form, the transferable record, lawmakers have opened the door to the use of electronic negotiable instruments. This should provide an opportunity to industries, such as the real estate mortgage and equipment financing industries, which until now have resisted the use of electronic media to adopt new technologies. Quite directly, the business risk created by the uncertain legal status of electronic negotiable instruments has been eliminated for the real estate industry by E-Sign.

One last word on E-Sign: since E-Sign authorizes the formation of contracts by e-mail messages, electronic agents, or computer programs acting on your behalf, some commentators suggest including disclosures in the footer of a user's e-mail messages containing terms similar to the above as a notice of disclaimer unless, of course, the e-mail message is intended to authorize the formation of a contract or license. Whether such disclosure is essential to eliminate risk of unintentional contract formation or not is difficult to assess at this early stage in electronic contracting; even so, the suggestion that such a disclosure might be necessary reflects at least one real-world certainty: as the rules for contract formation are liberalized to encourage e-commerce, the potential that technology may push too far in the direction the law embraces should serve as an emphatic caution to be aware of the possible unintended consequences of our conduct in the increasingly complex world we simplistically call the information age.

UETA

Which law do you follow, UETA or E-Sign? Although UETA treats certain matters of electronic commerce entirely different from E-Sign, as stated earlier, UETA and E-Sign have similar objectives. E-Sign is a federal law. UETA is a uniform model law that states are enacting. To the extent that a state's enactment of UETA conflicts with the provisions of E-Sign, the federal law should prevail; otherwise, licensors should pay particular attention to the provisions of the relevant state law version of UETA.

Under UETA, an electronic record, signature, or contract may not be denied legal effect for the sole reason that the medium in which the record, signature, or contract was created is electronic. There may be other reasons to deny the enforceability of an electronic record or signature. UETA only provides that, to the extent that the law requires a writing or a signature, electronic records and signatures will satisfy that requirement. It is important to note that under UETA, an electronic record will be unenforceable against the recipient if it is not in a form capable of being printed or stored by the recipient. UETA is technology-neutral in that it does not require or disfavor any particular method or technology in recognizing electronic records, signatures or, contracts.

The drafters of UETA point out in UETA's prefatory note that "[UETA] is NOT a general contracting statute—the substantive rules of contracts remain unaffected by UETA." UETA applies only to electronic records and electronic signatures created, generated, sent, communicated, received, or stored after the effective date of UETA in the adopting jurisdiction. The electronic signatures and electronic records must relate to a transaction. A *transaction* is defined as "an action or set of actions occurring between two or more persons relating to the conduct of business, commercial, or governmental affairs." Consequently, transactions unrelated to business, commercial, or governmental transactions are not subject to UETA.

Aside from the notable conflict between the two laws, E-Sign appears to direct compliance with UETA under circumstances of doubt, despite the usual preference of Congress to make federal law paramount. To the extent that other states laws, such as the statute of frauds, conflict with E-Sign—and the state has not remedied the conflict in its enactment of UETA—the provisions of E-Sign would seem to apply, but in some instances the answer may be far from clear.

The principal way in which E-Sign and UETA differ is with regard to consumer protection. E-Sign regulates the manner in which consumer assent may be established electronically, while UETA simply directs parties to comply with state consumer protection laws. Since E-Sign sets forth specific provisions concerning the expression of agreement to the terms of a license directed toward consumers, it is worth noting the relevant requirements of E-Sign.

E-Sign specifies in section 101 that if a statute, law, or regulation requires that information be provided or made available in writing to a consumer, the use of electronic records is permitted upon compliance with detailed specifications and disclosures. E-Sign is not much of a consumer protection law. It does not require consumer consent before all electronic dealings; the consumer protection consent provision only applies where the law requires that information be provided or made available in writing to a consumer. The party required to furnish the information must:

- (a) Notify the consumer of any right or option to receive paper;
- (b) Notify the consumer of the right to withdraw consent to receive electronic notice and stating any consequences (including termination of the relationship) and fees upon termination;
- (c) Notify the consumer whether the consent is to the specific transaction or to notices during the course of the parties' relationship;
- (d) Inform the consumer how to obtain a paper copy of an electronic record and of any fee to be charged;
- (e) Furnish the consumer, prior to obtaining consent, with a statement of hardware and software needed for access to and retention of the records.

Under E-Sign, if a system change raises a material risk that the consumer will not be able to access or retain an electronic record, the consumer must be provided with another statement of hardware and software requirements and be given the right to withdraw the consent without imposition of any fees or other conditions. In addition, the consumer must once again either consent electronically or confirm the assent electronically. Quite remarkably, these detailed provisions do not seem to have the impact that one would expect. Under section 101(c)(3), E-Sign provides that the failure to obtain consent in compliance with its terms does not, of itself, affect the effectiveness, validity, or enforceability of any contract entered into with the consumer.

UETA's approach to consumer protection is like E-Sign, in that there is little substantive law protecting consumer transactions. Instead, UETA primarily focuses upon what the licensor must do to comply with state law requirements for notices, disclosures, and information. Generally, UETA provides that legal requirements to provide, send, or deliver information in writing may be satisfied with an electronic record capable of retention by the recipient at the time of receipt. A record is not capable of retention if the sender, or its information processing system, inhibits the ability of the addressee to print or store the record. UETA also preserves requirements concerning the manner of sending, posting, or displaying the record as provided for by the pertinent state law.

With regard to instances of how E-Sign and UETA might specifically differ, UETA, for example, specifies that persons may satisfy their record-keeping obligations through the use of third parties. E-Sign is silent on this point. In addition, UETA specifies that retained electronic records satisfy evidentiary, audit and similar requirements. While UETA explicitly limits its reach to electronic contracting or the use of electronic records and signatures, UETA is likely to affect a broad range of state laws if its enactment in any particular state closely follows the uniform model.

UCITA

UCITA is a proposed uniform body of law that applies broadly to computer information transactions, which includes licenses involving the distribution and use of software. UCITA was drafted to solve a problem, namely, the perceived need to standardize the commercial code that governed the licensing of all sorts of forms of digital information. To achieve optimal success by standardizing the law of information licensing, UCITA needs to be enacted uniformly in 50 states. Some of the proponents of UCITA might consider the enactment of the proposed law in significantly fewer than 50 states a success, nonetheless, since the debate over the drafting of UCITA already has standardized the conception of software transactions as licensing transactions rather than sales of goods or services.

What are UCITA's ground rules? To begin with, if a transaction includes computer information and goods, UCITA applies to the part of the transaction involving computer information, informational rights in it, and creation or modification of it. However, if a copy of a computer program is contained in and sold or leased as part of goods, UCITA applies to the copy and the computer program only if (1) the goods are a computer or computer peripheral, or (2) giving the buyer of the goods access to or use of the computer program is ordinarily a material purpose of the transaction. The motion picture industry, financial services, and the insurance industry are largely exempted from UCITA. In most other cases, UCITA applies to the entire transaction if the computer information and informational rights, or access to them, are the primary subject matter of the transaction.

Under UCITA, if a term of a contract violates a fundamental public policy, the court may refuse to enforce the contract, enforce the remainder of the contract without the impermissible term, or limit the application of the impermissible term so as to avoid a result contrary to public policy, in each case to the extent that the interest in enforcement is clearly outweighed by a public policy against enforcement of the term. In addition, generally, conflicts with a consumer protection statute and UCITA are resolved by UCITA yielding to the consumer protection statute. In this respect, UCITA will not provide a cover for a warranty disclaimer provision that runs afoul of a state's consumer protection rules.

Although the purpose of UCITA is to facilitate private transactions in information, licensing may be used to impose limitations on the information property rights of those that may exist in a copyright regime. Consequently, UCITA strikes its balance between fundamental interests in contract freedom and fundamental public policies such as those regarding innovation, competition, and free expression by establishing general principles that will enable the courts to react to changing practices and technology. General principles often do not serve the interests of those who would benefit from more specific prohibitions, but UCITA does provide for two important contingencies: (1) if federal law invalidates a state contract law rule or contract term, federal law controls, and (2) a contract term that varies the effect of a rule whose effect between the parties cannot be varied by agreement under the Copyright Act is unenforceable.

Except for rules that directly regulate specific contract terms, there is no general preemption of contracting by copyright or patent law under UCITA. Although a lack of a general preemption may be unacceptable for some states or a federal court may ostensibly invalidate the missing general preemption, UCITA's bias leans in the direction a contract statute might be expected to lean toward; namely, UCITA seems premised upon a general canon that courts should be reluctant to set aside terms of any agreement protecting the expectations of the parties, and this canon takes on special force where parties have actually negotiated the terms in their agreement. In other words, despite notions of fairness, consumer protection principles, or even intellectual property interests, the nearly unyielding rule is this: an agreement between parties should be presumed to be valid and a heavy burden of proof should be imposed on the party seeking to escape the terms of the agreement.

These generous contracting rules apply to open source license drafters as well as those who draft proprietary software licenses. Even so, opponents of UCITA frequently raise the point that UCITA is actually a rather bold attempt to reconstruct or distort the reality of what an information transaction is to one where such transactions are viewed in a light that greatly benefits digital information and content providers over the interests of consumers and nondigital business interests. Even so, UCITA is still likely to have a significant bearing on how courts may resolve disputes over information contracts in the near future;

its opponents might have blunted its impact, but UCITA's scope remains far-reaching.

Although UCITA does not apply to traditional goods such as cars, clothing, or consumer electronics in the form of television sets or video game machines, UCITA does apply to a significant range of commercial activity. UCITA broadly applies to electronic contracting: under UCITA, the parties may form an electronic contract in any manner sufficient to indicate assent. Generally, UCITA covers the creation and performance of computer information transactions, which is broadly defined to include almost anything from software transactions or Web sites to digital music files or electronic books. Not surprisingly, since the commercial activity that UCITA governs, information transactions, represents the fastest growing sector of the nation's economy, many view UCITA's potential importance as extraordinary.

Prior to UCITA's drafting, a sweeping legal reform in contract law was considered crucial to the continued growth and vitality of the new economy. Although traditional commercial contract law is governed by various state law enactments of the Uniform Commercial Code (UCC), lawmakers, courts, and lawyers disagreed over whether the UCC actually applied to many of the transactions characterized as *information* transactions. It was clear that the UCC governed commercial transactions such as the sale of books or cars, but software was assumed to be different.

Typically, software is both intangible and tangible. Software distribution involves the transfer or license of an intangible intellectual property right such as copyright or patent, as well as the acquisition of a tangible object that embodies the software, such as a computer diskette or CD-ROM. In some respects, software that is distributed on-line may not have any tangible representations, but downloadable software often involves a nonexclusive right to make backup copies of the software, which includes the option of storing the software in a tangible medium.

In this respect, sufficient numbers of lawyers and lawmakers were convinced of the dubious applicability of the UCC to software distribution, due to its intangible characteristics, that an exceptional drafting task for what ultimately became known as UCITA was initiated.

UCITA is the name selected for this legislation or model commercial code after the American Law Institute (ALI) rejected its predecessor formulation, which was known as the proposed Article 2B of the UCC. ALI, founded in 1923 to improve the administration of justice by setting forth a comprehensive Restatement of the Law in nine broad areas (Agency, Conflict of Laws, Contracts, Judgments, Property, Restitution, Security, Torts, and Trusts), cooperates with NCCUSL in developing and updating the UCC. Prior to UCITA, NCCUSL and ALI had agreed to consider amending the UCC directly by adding Article 2B. Since altering the default rules of contract law that are related to the increasing importance of information technology in the new economy necessarily involves complex issues and conflicting purposes, ALI and NCCUSL were unable to sufficiently agree on substantive provisions of Article 2B.

It is noteworthy that UCITA does not exclusively apply to software distribution, but also includes other intangible transactions under a more general category referred to as computer information. The pertinent terms of the UCC that *might* direct how courts may resolve disputes concerning the distribution of software are ostensibly Article 2. Article 2 of the UCC provides the substantive uniform code that governs transactions involving the sale of goods. When software makers began the ubiquitous distribution of software by use of licenses, considerable confusion arose as to whether the UCC should govern the legal significance of software transactions.

Initially, the confusion primarily concerned whether software distribution should be properly characterized as the sale of goods or as a type of transaction not governed under the UCC at all. In an attempt to arrest the confusion, NCCUSL and ALI commenced drafting sessions involving lawmakers, legal scholars, lawyers, and industry insiders to develop an amendment to the UCC labeled Article 2B.

Ultimately, this effort was unsuccessful. Proponents of Article 2B, however, did not abandon their efforts. Instead, they renamed Article 2B—since it could not become a provision of the UCC—UCITA. Undoubtedly, for opponents of Article 2B, the formation of UCITA represented a two-headed Hydra. Opposition to UCITA is significant, and opponents of UCITA have adopted a strategy to slay its enactment in each state in which it is under active consideration.

To be sure, some opponents of UCITA do not consider the proposed law entirely unsuitable for enactment. Many critics focus their opposition to UCITA on its anticonsumer protection provisions as well as those provisions that tend to undermine the fair use doctrine embodied in the Copyright Act. In this respect, and in the interests of full disclosure for the reader of this text, it should be noted that the author joined a group of legal scholars and consumer groups that lobbied the state legislature in Maryland to amend its adoption of UCITA to include certain consumer protections and fair use principles. A copy of a letter echoing this sentiment and sent to the Maryland General Assembly may be viewed at http://www.cdt.org.

Many of the reasons advanced for opposing the adoption of UCITA involve two general concerns, namely, the potential that UCITA may authorize licensors to draft nonnegotiated, mass-market licenses that undercut current consumer protection laws and that UCITA relies on private contract law to circumvent public policy determinations enacted by Congress in the Copyright Act.

With regard to the latter claim, the argument is that since the Copyright Act has never been determined to expressly forbid copyright holders from issuing licenses to offer end-users access to intellectual property, regardless of whether the license terms are contrary to privileges granted to users by the act, UCITA could be viewed as explicitly authorizing copyright holders to draft nonnegotiated, mass-market licenses that explicitly conflict with the Copyright Act. For instance, a licensor might draft a license for an electronic book or e-book that prohibited unauthorized copying by libraries and required libraries and their patrons to pay a fee for any and all uses of the e-book. Critics of UCITA argue that such a provision would be unlawful under the Copyright Act, but is authorized under UCITA.

The trend by copyright holders toward ubiquitous reliance on software licenses as the means of software or information distribution regardless of whether the licensee is a mass-market consumer or a well-positioned business end-user is the primary engine driving the inherent tension between a copyright holder's freedom to contract and an end-user's rights denominated by the public policy objectives of copyright law. Undoubtedly, UCITA adds signifi cantly to this tension. Even so, the merits and demerits of UCITA are only briefly reviewed in this book. Readers interested in further study of UCITA might begin their quest by consulting a number of sources found at http://www.2bguide.com and http://www.ucita.com.

Two states have enacted UCITA, Maryland [2] and Virginia, and in each state the battle over the proposed legislation has been fairly intense. New Jersey halted its consideration of UCITA to await the results of a state-empaneled commission to specifically consider whether UCITA may have an unsettling impact on public policy choices evidenced by legislative amendments to the Copyright Act. Iowa recently passed what commentators refer to as controversial "bomb shelter" legislation, which is a reference to a legislative attempt to protect a state's residents from being subject to UCITA licensing terms enacted by other states.

To be sure, some opponents of UCITA do not consider the proposed law entirely unsuitable for enactment. Many critics focus their opposition to UCITA on its anticonsumer protection provisions, as well as those provisions that tend to undermine the fair use doctrine embodied in the Copyright Act. Many are concerned whether open source licenses will be enforceable under UCITA, for example. As noted in Chapter 2, UCITA is a model contract law statute that may facilitate computer information transactions in cyberspace and clarify state laws governing computer information transactions, including commercial transactions involving agreements regarding the distribution of computer software, computer databases, and other Internet-based information. UCITA permits parties to opt in to its regime fully and thereby designate one set of rules to govern a software transaction.

Much of the debate on UCITA's substance at the state law level centers around its contract formation rules, especially in the context of mass-market transactions. Under UCITA's contract formation provisions, a party can manifest assent to a contract by acting or even by failing to act. The liberal construction of assent in UCITA undermines the fundamental contract law requirement of a "meeting of the minds" in contract formation. Although the efficiency from this mode of assent might make it desirable under traditional contract law, inferring assent from a failure to act allows for consumer abuse in the context of mass-market computer information licenses.

The potential danger to consumers becomes especially acute for shrinkwrap licenses, in which the terms are not presented until after payment. For these contracts, a mere failure to return the product can constitute assent and thus bind consumers to unintended terms. Consequently, open source licenses are enforceable at least to the same extent as the uniform law covers other mass-market information transactions. In some instances, a software developer may prefer to freely offer access to the source code for their software product, generally, but to withhold access from those who want to use the software in a commercial distribution or to exercise a degree of artistic control over modifications to the original software project.

Proponents of UCITA have widely dispersed substantive responses to each criticism leveled against the enactment of the proposed UCITA, and the reader will be directed to further reading on UCITA in the final chapter. The goal in this chapter is to highlight the general controversy surrounding UCITA's enactment within the broader context of software licensing issues. In this regard, essentially, UCITA, like other model commercial codes, provides a series of default rules that parties may rely upon in shaping their commercial transaction. In a contract or license, if the parties fail to specify certain terms, these terms will be determined, when necessary, as provided for by the default rules of UCITA. For example,

> The owners of a popular Web site might send a written request to a software vendor containing the message "Please send us by FTP a copy of the current upgrade of FreeSoft under the usual license terms ASAP." In response, the software vendor might send an electronic message indicating "Got the order, and will ship program under an upgrade license immediately." Assuming the parties have formed a binding contract, an issue might arise about whether a license term under the upgrade for FreeSoft is enforceable if it differs substantially from what the Web site owners intended by the use of the phrase "usual license terms." Under the circumstances, the default rules of the applicable commercial code would guide the court in filling in the terms of the agreement. It is noteworthy that issues concerning whether the upgrade license itself is enforceable is an entirely distinct matter that would also be subject to the default rules of UCITA.

In this manner, UCITA provides a legal regime to which parties may reliably resort when disputes over a given transaction or contract term arises. Parties may, for example, bind themselves to license terms on how a software product may be used, but not include in the license a provision on whether the license itself is transferable. If so, and a dispute arose concerning transferability, UCITA would provide the default rules on transferability.

Of course, the existence of the license usually supports the notion that the parties intended that many of the default rules were to be overridden by the license terms. Even so, it is noteworthy that some of UCITA's so-called default rules cannot be overridden by contract or consent. These rules include UCITA's provisions concerning the obligation to act in good faith when negotiating a computer information license and the provisions regarding consumer protection. For example, although software vendors routinely do not agree to pay consequential damages (lost profits), UCITA awards consequential damages as the default rule unless the parties contract otherwise.

Under certain circumstances, UCITA appears to make it more difficult for buyers to prevail in court on traditional breach of contract claims. For example, the software developer may distribute or tender the software program under a condition that is substantially less than perfect. In doing so, the buyer cannot prevail in a breach of contract claim unless the buyer shows that the claimed defect represents a material breach of the license agreement. In this manner, UCITA makes it more difficult than under the UCC to reject a software product before becoming obligated to pay for it.

In the area of e-commerce, Web sites that merely ask the customer to fill in basic payment information may leave critical issues undetermined, such as what warranties, if any, apply. What are the remedies for breach? There are significant areas of potential concern with UCITA. Software developers may be able to lawfully and technologically block users from certain uses of their software that are lawfully considered permissible as fair use under copyright law.

An additional area of concern regarding the enactment of UCITA may be the ironic support the uniform law would provide for the enforcement of sneak-wrap licenses on the Internet. Some legal scholars have warned that UCITA might authorize sneak-wrap licenses. UCITA does not prohibit software developers who are engaged in a continuing contractual relationship for an on-line service from imposing capricious and arbitrary changes in the terms of their software licenses without notice. This is particularly troublesome for on-line agreements where an altered click-wrap license for daily access to a database may become a sneak-wrap that an end-user almost unconsciously clicks each day until the new terms are noticed.

For example, UCITA seems to allow a software developer to prohibit end-users from doing reverse-engineering, prohibit the user from telling the public about bugs the user encounter, insist that the user can only sue them in the state of Virginia, change the license terms ex post facto, and enter the user computer to shut off the software if the use has violated the license. If this is a faithful reading of what UCITA permits, then software authors who rely upon proprietary software licensing may attempt to extract additional payments from users for private, noncommercial uses of software that have been traditionally viewed as fair uses under copyright law, but soon will be transformed into fair game for payment under UCITA.

A final hurdle to the enforceability of software licenses governed by the default rules of state law is potentially a high hurdle. Software licenses may be subject to preemption analysis under both section 301(a) of the Copyright Act and the Supremacy Clause. Although the court in ProCD held that mass-market licensing agreements held procedurally and substantively enforceable under the applicable contract law should not be preempted under section 301(a) of the Copyright Act, that court's holding is neither dispositive nor viewed, generally, as persuasive. More to the point, whether mass-market licenses should be subject to Supremacy Clause preemption independent of section 301(a) preemption, which was not decided by the Court, clearly shows that the matter remains unresolved.

Warranties and "As Is" Licensing

Consumer protection statutes have varied state by state. In this respect, open source software license drafters should be cognizant of, when applicable, consumer protection statutes or regulations that regulate warranty disclosures, limits on disclaimers of warranty, advertising, mandates, disclosure requirements of specified license terms regarding content, manner of disclosure, typeface or the like, and potential burdens imposed on licensors regarding recovery of treble damages for particular types of contract breaches—particularly if the licensee is a consumer.

The Magnuson-Moss Warranty Act is a federal law that governs the content and disclosure of written warranties. Generally, a warranty is a statement about a product that may assist the buyer or consumer in rendering a risk assessment of potential harm or damage that may flow from the product's use. As such, warranties aid consumers or buyers in determining whether they should buy a particular product.

The act is codified in the U.S. Code at 15 U.S.C. Section 2301 and following. There is some debate on whether software distribution meets the Magnuson-Moss Act's definition of consumer product or tangible personal property. Federal regulations promulgated under the authority of the act seem to resolve most doubts about the its applicability. See 16 CFR, Section 700.1(a) ("Where it is unclear whether a particular product is covered under the definition of consumer product, any ambiguity will be resolved in favor of coverage.")

The rule of Magnuson-Moss is unequivocally clear: If a written warranty is given for a consumer product, implied warranties cannot be eliminated. Since UCITA authorizes "as is" licenses, which ostensibly permit software to be distributed to the mass market as if it were sold at a fire sale or, as some might say, on a used car lot. An "as is" disclaimer usually means that the license contains no warranty that the software works for a particular purpose or any purpose at all or that the licensee may obtain a refund to vindicate a warranty interest. The question arises of whether other laws or legal presumptions conflict with UCITA.

Whether the focus is a software license or some other type of license intended for transactions involving consumers, the Magnuson-Moss Warranty Act governs the content of all license provisions in the United States that disclose written warranties for consumer products that are sold to consumers. Consequently, a warranty provision is a common term found in licenses. If a license contains a written warranty for a consumer product, then the license cannot contain provisions that attempt to eliminate implied warranties. Two points are worth stressing:

1. Although licenses containing written warranty provisions cannot entirely eliminate implied warranties, an implied warranty may be limited in duration by the terms of the license; indeed, it may prove reasonable to limit the duration of implied warranties to the duration of the written warranty.
2. A consumer is entitled to rely on the Magnuson-Moss Warranty Act's provisions as long as the transaction constitutes a sale of a consumer product. Consequently, the question arises, if a consumer acquires a software item that accompanies a license, does the consumer have legal support to insist that the written warranty, assuming there is one, not eliminate implied warranties? In practice, software developers often use licenses that warrant that the disk is free from defects, but provide that the software on the disk is offered "as is." In other words, the terms of the license warrant that the disk is fit for ordinary purposes, but provide no such warranty for the actual software. Of course, one might ask whether this is a fair business practice.

Under the Magnuson-Moss Warranty Act, sellers of products must make written warranties available, if such warranties are provided, where those products are sold so that consumers can read them before they purchase the product. As noted earlier in this chapter, there has been considerable debate over whether software distribution should be treated as a product that is sold or as a license for the use of software. Similarly, commentators raise the question of whether the warranty disclosures permitted by UCITA can withstand the Magnuson-Moss standard of conspicuous disclosures. UCITA permits software vendors to disclose warranty terms after payment has been made for software, so long as consumers have reason to know that the terms would follow after payment. In this manner, UCITA explicitly validates shrink-wrap and click-wrap licenses, while Magnuson-Moss would seem to render those licenses invalid.

Manufacturers and sellers of consumer products who provide consumers with detailed information about warranty coverage must also comply with three basic requirements:

1. Warrantors must designate whether the warranty is written as either full or limited warranty.
2. Warrantors must state certain specified information about the coverage of the warranty in a single, clear, and easy-to-read document.
3. Warrantors and sellers must ensure that warranties are available where the warranted consumer products are sold so that consumers can read them *before* buying.

The act applies to *all* written warranties on consumer products costing more than $10. The advantage to providing consumers with a way of learning what warranty coverage is offered on a product before they buy, the product is obvious; the consumer is provided with a way to know what to expect if something goes wrong, or if something occurs that may decrease customer satisfaction.

In addition, the act ensures that consumers can compare warranty coverage before buying. By comparing, consumers can choose a product with the best combination of price, features, and warranty coverage to meet their individual needs. The act also promotes competition on the basis of warranty coverage. By ensuring that consumers can get warranty information, the act encourages sales promotion on the basis of warranty coverage and competition among companies to meet consumer preferences through various levels of warranty coverage. Thus, the act makes it easier for consumers to pursue a remedy for breach of warranty in the courts, but it also creates a framework for companies to set up procedures for resolving disputes inexpensively and informally, without litigation.

To be precise, the Magnuson-Moss Act does not require the provision of a written warranty. Instead, the act does not apply to a given transaction *unless* a written warranty (oral warranties are not regulated by the act either) is provided. Once a written warranty is offered, however, the transaction must comply with the Magnuson-Moss Act.

Limitations on Liability

Most licenses include provisions that limit the liability of the software developer, copyright holder, distributor, or, perhaps, all parties. The limitation may include an exclusion from liability for all damages other than direct damages (i.e., only the costs incurred as a result of personal injuries or injury to property and costs to repair or replace damaged or defective goods delivered under the license).

A software developer may be liable for breaching both express and implied warranties. Although the extent to which implied warranties under the UCC or UCITA may apply to software distribution is unclear, since the law is rapidly changing in this area, if a state's commercial code does provide for liability of a licensor for breaching not only express warranties but also implied warranties of merchantability and fitness of the software for a particular purpose, the licensor may escape liability under these implied warranties by including a conspicuous disclaimer of all warranties as part of a software license agreement.

UCITA includes the implied warranties of merchantability and fitness for a particular purpose, along with newly created implied warranties of quiet enjoyment, system integration, and, in some situations, accuracy of data. The implied warranty of merchantability is a statement that the product is fit for the ordinary purposes for which goods of its type are purchased. In contrast, the implied warranty of fitness for a particular purpose is a statement that the product meets the needs the software developer knew about or should have known about. UCITA also provides for a statutory right to a cost-free refund if, under certain circumstances, the terms of a license are not acceptable to the buyer.

Since most states require limitations on liability to be conspicuous, the common practice is to ensure that the licensee is familiar with the limitations on liability imposed by the licensor by using capitalization for each letter of each word in this section of the license. Generally, the purpose of this provision is to limit the end-user's range of remedy for errors of any sort to those that the law forbids the licensor to disavow.

Under this provision, the drafter of the license should disclaim liability for damages. This provision may be used to preclude the government from obtaining unlimited rights in a developer's software. Most of its provisions are written in terms of the rights and obligations of the software vendor and end-user.

Choice of Law Provisions

This provision usually contains a governing law clause, a choice of law rule, a rule of adjudication, and/or a specified preference on personal jurisdiction. For example, the clause may indicate that disputes are to be resolved by arbitration or may specify that disputes must be resolved within a given state or that the dispute must be resolved by the laws of a given state—in this regard, for example, the provision may require that a dispute be resolved by a state's courts where UCITA has been enacted.

Although this provision allows the software author or licensor to attempt to control the law that will be applied to the license, if the license terms are disputed, other factors might limit the enforceability of the provision. Consequently, some provisions may be ignored if a court concludes that enforcement of the provision raises an impermissible burden on a licensee's constitutional due process by encumbering a party's ability to appear before a court of a certain jurisdiction.

To ensure that an electronic software license is binding, the license terms should specify the property, interests, or rights granted by the license. The license should also identify the parties or legal entities involved, indicate the obligations of the parties, demonstrate that there is a mutuality of promises (a "meeting of the minds"), and evidence the acceptable form of consent that provides proof that the parties agreed to the terms of the license.

Although it may be some time before we alter the current reality that for most licensors a written license will be preferable to an electronic agreement, the increasing use of electronic contracts and electronic commercial transactions has stimulated the development of commercial codes and contract law in a direction that seems to be both eroding the significance of some traditional legal formalities and encouraging the development of new formalities that may entirely replace the remaining traditional ones (for example, written signatures are being replaced by digital signatures and electronic signatures). Consequently, the use of electronic licensing to distribute software or other information products is likely to become a dominant business model.

Endnotes

[1] E-Sign, for example, contains various consumer protection provisions that apply where a law requires that certain information be made available to consumers in writing. In these types of situations, consumers must affirmatively consent before electronic media can be used to satisfy legal requirements with regard to the providing of information to the consumers in writing, and consumers must be provided with a clear and conspicuous notice of their ability to receive information in electronic and nonelectronic form if they so choose.

[2] Maryland enacted its own version of UCITA, which ultimately took into account some of the criticisms of the model code noted in the text of this chapter.

6

Commercial Models

With open standards, an open source participant can take a program; fix it, port it, or make it like new.

Open Commercial Uses

A point worthy of repetition: at the heart of the concept of open source software is the idea that the best software is developed under conditions where software is freely available in a manner that permits any licensee or end-user to modify it, use it, share it, or make it better. The emphasis for open source is on *information access*—open access to the ideas as well as the expression contained within software applications. The end result of open access to ideas is innovation and the development of more ideas. In this regard, open source software spawns open source software. In an environment so enriched with software, one might ponder how commercial developers may earn money selling it. The question, of course, assumes that a robust but competitive environment is inimical to commercial interests. Although copyright is a monopoly granted with a government imprimatur, artificial limitations on access to information is neither necessary nor desirable in a genuinely open and competitive marketplace. Consequently, commercial software vendors exist in increasingly significant numbers within the open source space.

Of course, the ease with which a proprietary commercial software project may be ported to an open source model should not be overstated. Several factors may impact the inevitable success or failure of porting a commercial proprietary software product to open source, including whether the complexity of the software project or the learning curve for understanding the existing codebase is too sharp for those unfamiliar with the project. If there is not much introductory documentation to familiarize new developers with or if the documentation is poor, it may be difficult to recruit new developers to the project. In addition, if the codebase is extremely unstable or developers are plagued with frequent changes from those managing the project, the project may not port successfully. Even the model of licensing may affect whether the project ports successfully; namely, open source developers may balk at a dual-licensing model that maintains an open source license that is highly restrictive.

An example of the commercial licensing of software under the GPL is Red Hat, which distributes Linux. As a commercial distributor of open source software, Red Hat obtains all of the freedoms granted by licensors of open source software, but, most importantly, an open source vendor, unlike a proprietary software vendor, is unable to restrict the distribution of the software. Red Hat cannot demand from PC makers control over the desktop in order for the computer manufacturer to obtain a Linux license. Indeed, Red Hat is just one of many distributors of Linux. The original equipment manufacturers (OEMs) could even continue to use the Red Hat Linux software they already obtained, since the GNU GPL gives them the right to redistribute anything it wants as long as it continues to make the source code available.

However, this forces these businesses into competition that springs from an equal footing, technologically, with other vendors and potential competitors. Thus, even if a market, such as that for operating systems, is a natural monopoly, any monopoly benefits would accrue to Linux generally, rather than to any specific Linux vendor [1].

In this chapter, we explore the types of licenses that can be drafted that secure the basic goals of open source licensing by offering alternative means of distribution for special purposes. In this context, we explore open source software under the distinct commercial models of dual licensing and application services [2]. For example, a dual license is primarily used to distribute software under an open source software license and a proprietary software license. Sun Microsystems uses a dual-licensing arrangement to distribute its Java interface and StarOffice office suite.

Dual licensing offers the developer the freedom to license a software program under an open source license, like the GNU GPL, as well as distribute the software as a proprietary commercial product. Dual licensing is primarily used by developers who desire to generate revenues from directly selling software rather than selling the services or hardware products that support or accompany the distribution of software. So the developer allows its source code to enter the open source codebase under the GPL while retaining the freedom to distribute the software under a less restrictive open source license or, perhaps, a closed source proprietary license. Notwithstanding the fact that dual licensing promotes code forking in a manner that is, generally, unappealing to many open source adherents, dual licensing also encourages the developers of highly desirable software to participate in open source projects, who otherwise would be disinclined to share any of the source code with other programmers.

Standard-form contracts constitute an efficient method of product distribution; mass-market and Web site licenses are examples of applying this efficiency to intangible goods and/or services. These licenses have been particularly important in the distribution of Internet technologies. Similarly, the ability to distribute software under a dual-licensing model seems central to bringing proprietary commercial software developers into open source development. The Artistic License has become a fairly common tool for releasing software under two licenses; in this respect, the licenses are aimed at slightly different objectives and are targeted toward different potential end-users.

One difficulty that may arise under dual licensing centers on whether a licensor may issue a dual license where the freedom to use source code that is distributed under a public license is restricted by a copyleft provision, while the second license issued applies to a proprietary version of the same software program. This is an especially thorny issue. If source code already has been released under the GNU GPL, for example, it is doubtful that a circumstance could arise that would permit a dual-licensing arrangement that included a license that did not contain a copyleft provision or that was incompatible with the GNU GPL in any pertinent manner.

Dual licensing requires careful consideration of whether the codebase of the open source software will suffer too significantly to prove worthy of programmers' efforts. There are a number of unsettled issues that may arise and a bevy of unanticipated consequences that may overtake the objective of dual licensing as that approach surmounts hurdles between the developer's desire to distribute a commercial version as well as fulfill the developer's obligations accompanying the release of the program as an open source product. There are limited examples that seem to be working: both Netscape and Sun Microsystems have used dual licensing to distribute open source works, and the AFPL has successfully governed the shared codebase for GhostScript [3]. Sun Microsystems designed its wireless protocol, the JavaPhone API, to simplify telephony development using an open platform, Java technology, as a core element of mobile telephony. The API is distributed under dual-licensing arrangements. The open source license provides for a "non-exclusive, royalty free, license to use, modify and redistribute this software in source and binary code form, provided that i) this copyright notice and license appear on all copies of the software; and ii) Licensee does not utilize the software in a manner which is disparaging to Sun Microsystems."

Interestingly enough, it appears that Sun has considered the impact of a court construing its license as a contractual legal instrument; hence, to address contract formation issues, Sun Microsystems signals potential licensees that consent to the terms of the license is being expressed when the end-user downloads the software: "By downloading this sample software, you accept and agree to the terms and conditions below. If you do not agree, do not download or use the sample software."

Apparently, the dual licensing also works by switching terms depending upon whether the software in which the Java source code is copied is open source software or a proprietary commercial product. Ostensibly, Sun may be the most visible enforcer of what is at least a nominal open source license. The developer appears to have successfully enforced the terms of one of its Java licenses against Microsoft, who paid substantial licensing fees to settle a legal dispute over the license with Sun. Sun's use of dual licensing seems to be closely patterned after others; namely, the developer simultaneously uses two licenses, open source and proprietary, to capture revenue from software sales as well as revenue leakage from free-riding.

Jabber is the name given to an open Extensive Markup Language (XML) protocol devised in 1998 for instant real-time asynchronous communication, or, in more crude terms, Jabber is a protocol for the development of open source instant message systems quite similar to AOL's, Yahoo's, or MSN's instant messenger (IM) programs. Its open source, open standard architecture readily allows end-users or developers to create their own IM services on servers they own. In Poland, it has been reported that nearly a million users send and receive IMs through a single server. As an open protocol, Jabber's project (http://www.jabber.org) is managed more loosely than an individual copyright holder might manage a software project. In this respect, developers of Jabber-compliant software are unrestricted in the software license used to distribute their software. Despite the lack of restrictions, many Jabber-compliant software applications are distributed under open source software licenses: the two favored open source licenses are the Jabber Open Source License and the GNU GPL.

The Jabber license, excerpted below, is a unique dual-licensing model in that it sets forth conditions for what might be best described as dual licensing by the licensee. The licensee is permitted in subsection 4(e) to establish an additional set of terms that would apply to those who agree to both licenses. Default contract formation rules might govern whether dual-licensing agreements are enforceable as the licensor intends. If the complete terms of a license cannot be clearly established in a single instrument, then additional documents should not substantially alter the terms of the original agreement to avoid contract formation issues from arising.

The Jabber Open Source License Template

This License provides that:

> 1. You may use, sell or give away the Licensed Product, alone or as a component of an aggregate software distribution containing programs from several different sources. No royalty or other fee is required.
>
> 2. Both Source Code and executable versions of the Licensed Product, including Modifications made by previous Contributors, are available for your use. (The terms "Licensed Product," "Modifications," "Contributors" and "Source Code" are defined in the License.)

The Jabber license raises an interesting question: What damages may a licensor seek for breach of a software license? The answer to this question turns, in part, on whether the open source software copyright holder elects to bring suit, after failed attempts at negotiation, on the basis of breach of the software license or copyright infringement. To be certain, the copyright holder may have no choice if the only interest redressed is copyright infringement; under that circumstance, the copyright holder must bring suit in federal court on the federal law claim of copyright infringement. If, however, the software license is drafted as a license with terms that may be enforced independently of copyright law (this is a matter that has not been generally resolved and may be highly dependent on the terms of the software license), the copyright holder may be able to proceed on multiple claims. For example, as noted previously, Sun Microsystems recently brought suit against Microsoft on the basis of copyright infringement and breach of contract.

It could do so because it contended that Microsoft had exceeded the scope of its license by creating an enhanced version of Java that was fully operable only on Microsoft's operating system and, further, by not adapting its implementation of Java to be compatible with Sun's addition to Java of a component known as the Java Native Interface (JNI). The court found significant support for Sun's argument that Microsoft's Java compiler violated the compatibility provisions of the software license agreement between the two parties. In particular, the court was persuaded that Microsoft's extended Java compiler did not execute properly with a JavaSoft Virtual Machine of the same version as required by the terms of the license. The court noted that even the preface of the license terms was instructive as to Sun's intent, which stated that the specification is meant to ensure that "the behavior of every language construct [be] specified, so that all implementations of Java will accept the same programs." At an early phase of the case, the district court held that Sun was likely to prevail on its claim that Microsoft had violated the license agreement. Why were Sun's software license terms so persuasive?

Interestingly enough, it turns out the license terms may not have been persuasive as a matter of contract law. The district court did not fully explain why, but apparently it had viewed Sun's claim primarily as a copyright infringement claim rather than a breach of contract dispute. In the short run, this required the district court to review its holding again, since Microsoft was able to convince an appellate court that the issue as to whether a software license created affirmative covenants, which are contractual in nature, or simply imposed restrictions on the copyright grant, which would require a statutory remedy, was insufficiently answered by the district court. The case settled before the district court further ruled on the matter, although it was quite likely clear to the parties that Sun would prevail on its claims ultimately.

Microsoft had argued that the broad grants provision in the software license allowed it an unrestricted right to modify Sun's source code and create derivative works, and that the company made a separate and distinct contractual promise to honor Sun's compatibility requirements. Sun had argued that the grants provision along with the compatibility terms of the software license must be read together to create a single license that granted Microsoft only the right to make compatible modifications and derivative works. Each side raised a plausible argument as to the parties' intent. To some extent, the enforcement of copyright licenses raises issues in a gray area between or beneath copyright and contract law.

Thus far, the rules that guide this area of the law include the clearly expressed terms of the contract and the copyright holder's clearly identified intent to enforce copyright interests. In other words, for the licensor to gain the benefits of copyright enforcement, the licensor should definitively establish that the rights it claims were violated are copyright, not contractual, rights (this is so regardless of whether a unified or dual-licensing system is established) [4]. In *S.O.S.*, the plaintiff, which held a copyright in a computer program, had granted the defendant a license to use the software and had explicitly reserved all other rights. The plaintiff claimed that by modifying the software, the defendant had exceeded the scope of the license and therefore infringed the copyright. The district court, using California contract law to construe the license, applied the rule that contracts should be construed against the drafter and held that the license therefore permitted any uses not explicitly forbidden. On appeal, the Ninth Circuit agreed, but cautioned that it would enforce a contract claim as long as doing so would not "interfere with federal copyright law or policy."

When Netscape decided to distribute its Web browser as open source software in 1998, it had to overcome the licensing difficulties that accompanied the fact that its browser source code included code licensed under a number of distinct commercial licenses, some of which could impede licensing the code under the GNU GPL. Consequently, Netscape ultimately produced two licenses: the Netscape Public License (NPL) and the MPL. The licenses were actually drafted in a manner similar to how open source software is created, namely through an open and public process in which the license drafters solicited comments on various provisions in the licenses and incorporated those suggestions in the released versions of the NPL and MPL [5].

The MPL satisfies the open source definition. It contains 13 sections with no preamble. Definitions are contained in section 1. Section 2 grants the licensee a worldwide, royalty-free, nonexclusive license to use, reproduce, modify, display, perform, sublicense, and distribute the licensed software. Section 3 grants the licensee the right to distribute modifications of the software only if the licensee (1) makes the source code of the modified version available under the same terms of the license, and (2) gives proper notice of the license. These requirements are similar to those in the GPL. Other provisions also containing clauses similar to the GPL include a disclaimer of warranties in section 7, a statement that termination results if the licensee fails to comply with the terms of the license in section 8, a limitation of liability in section 9, and a complete integration clause in section 11.

The NPL covered code directed solely toward commercial or proprietary software, while the MPL covered open source licensing for commercial uses. The NPL contains a clause that is absent in the MPL; namely, Netscape, in the NPL, reserves its unilateral right as copyright holder of the original work to use any modifications to the Web browser source code provided by licensees in both Netscape's open source product and any proprietary version of the program that is developed. Both the MPL and the NPL permit commercial licensing of derivative works and changes to covered program source code to be made freely available to anyone, and neither license contains a copyleft provision.

The NPL provides that Netscape grants licensees a "royalty-free, nonexclusive license…to use, reproduce, modify, display, perform, sublicense and distribute the Original Code (or portions thereof) with or without Modifications, or as part of a Larger Work." The NPL ensures that the licensor is able to protect existing proprietary product licenses that remain valid despite the licensor's subsequent adoption of open source distribution for the same or similar work. As noted, the NPL actually includes the terms of the MPL, but also retains for the licensor the right to use code licensed under the NPL in other products without having those products fall under the NPL, and the right to independently license the covered source code, including additions contributed by licensees, to third parties under proprietary product license terms.

As noted, one of Sun's dual licenses was introduced by Sun as the SCSL. According to Sun, this license template supports open development standards in two ways. First, the license template requires compatibility among released versions of the software. Secondly, the license promotes open licensing by allowing modifications and extensions to the software to be closed up through proprietary software licensing. Apparently, the SCSL is viewed by Sun as blending the best aspects of the proprietary and open source license models. Certainly, it is correct that by keeping control over the source code, the licensor ensures compatibility among the varying versions of software. Even so, the compatibility standard might blend the advantages of two worlds too intimately. One wonders whether the SCSL could confuse end-users as to what is and what is not open source software. Notably, compatibility issues arise under the SCSL because Sun uses a decentralized project management framework that is not required or frequently adopted by open source projects.

To implement the SCSL licensing method, the licensor uses two license templates: the Research Use license and the Commercial Use license. The Research Use license is a click-wrap license available on the Internet, while the Commercial Use license must actually be signed and executed by both the licensee and the licensor. The Research Use license primarily grants all licensees the royalty-free rights to use, reproduce, and modify the original and upgraded source code for research; to distribute copies to other licensees and students; and to test code. The Commercial Use license grants the licensee the right to reproduce and distribute compliant code that passes certain test suites and conforms to certain specifications. In addition, royalties may be required from the licensee.

The Open Software License

The content of the Web site managed by OSI is protected by the Open Software License (OSL). The OSL is intended to be a flexible license that is extendable for multiple open source purposes. It is an open source license that is listed on the approved list of licenses supported by OSI. As an approved open source license drafted under the close guidance of OSI itself, this license is intended to serve the same functions as the GPL has served for the FSF, with a few notable exceptions: (1) there is no contention by OSI that the OSL neatly avoids controversial issues regarding contract formation principles by being self-denominated as something other than a contractual obligation, (2) the license contains an apparent weak copyleft, and (3) the license contains a mutual termination clause as a defense to patent claims that undermine the goals of open source software. While these distinctions are intended to overcome weaknesses in the GNU GPL, the distinctions are also significant enough to render the OSL incompatible with the GNU GPL. Indeed, quite unlike the GPL, the OSL is intended to be used in compliance with contract formation principles as they apply to the Internet. In this regard, the OSL may provide a suitable alternative licensing template for those who are genuinely concerned about distributing their open source works under the GPL and other open source software licenses that do not clearly follow valid contract formation procedures.

The OSL "grants…world-wide, royalty-free, non-exclusive, perpetual, non-sublicenseable license to do the following: a) to reproduce the Original Work in copies; b) to prepare derivative works ("Derivative Works") based upon the Original Work; c) to distribute copies of the Original Work and Derivative Works to the public, with the proviso that copies of Original Work or

Derivative Works that You distribute shall be licensed under the Open Software License."

The OSL is a license used to distribute open source works that constitute any original work of authorship, including software, documentation, music, art, or any other copyrightable work; traditionally and by contrast, the GPL was drafted as a technology license and has primarily applied only to software. The first sentence of the license describes the mechanism for associating the OSL with an original work; there is no such provision in the GPL. Section 1 is an explicit copyright grant, including all rights listed in 17 U.S.C., section 106. The proviso in section 1(c) is a reciprocity provision equivalent to that in the GPL. This license is not sublicenseable, meaning that the licensor retains privity of contract (and hence the right to enforce this license) with all who obtain the original work.

Section 2 provides a "non-exclusive, perpetual, non-sublicenseable license, under patent claims owned," which is an explicit patent grant. The grant extends only to claims that are embodied in the original work. This license is also not sublicenseable, meaning that patent rights come directly from the licensor and the licensor can sue directly to enforce the patent license. Section 3 sets out the specific terms for how the licensee must ensure that the availability of source code must comply with the open source meaning of public access, that is, how to lawfully access and modify the work.

In the definition of source code, the OSL requires that the licensee make available all documentation to modify the original work—but not documentation to access it. To emphasize this point, the license contains a definition of *external deployment* that highlights that the documentation requirement is not meant to besiege potential licensees or otherwise inhibit the acceptance of open source software. What sets the OSL apart from other open source licenses are the terms that are regarded as a built-in defense against patent grants. There is an explicit patent grant from the licensor, but, more important, the license contains a mutual termination clause in the event there is an assertion of a patent interest that would disrupt the licensor's ability to maintain control over the project covered by the open source license. The OSL also contains an explicit copyleft provision, but the provision may be viewed as weak, since the license excludes the right to sublicense and does not contain an affirmative requirement regarding the licensee's obligation to cause any OSL software distributed to do so under the terms of the OSL.

> 4) Exclusions From License Grant. Nothing in this License shall be deemed to grant any rights to trademarks, copyrights, patents, trade secrets or any other intellectual property of Licensor except as expressly stated herein. No patent license is granted to make, use, sell or offer to sell embodiments of any patent claims other than the Licensed Claims defined in Section 2. No

> right is granted to the trademarks of Licensor even if such marks are included in the Original Work. Nothing in this License shall be interpreted to prohibit Licensor from licensing under different terms from this License any Original Work that Licensor otherwise would have a right to license.

The *external deployment* captures the essence of public distribution of software in its modern forms, including the use of software as an application service provider (ASP). Open source software licenses that do not use explicit definitions for public distribution may not be viewed as encompassing the use of open source software under the ASP business model, since it is unclear whether ASP services actually result in a distribution of software in the traditional denotation. The next version of the GPL is likely to address the ASP model by prohibiting licensees from modifying the free ASP software by deleting or disabling a command in the ASP program that allows end-users to download source code. Sections 4 and 5 also specifically exclude the licensor's trademark and other intellectual property from the license grant clause, and contains a warranty that the licensor owns (or has the right to license) the original work. The warranty provision differs from the GPL and most other open source licenses by explicitly providing an assurance to the licensee that the licensor has a lawful right to license the work at issue; subsequent users and licensees should find the warranty helpful as a potentially reliable safeguard if litigation arises over the authority to use a work or, subsequently, if a demand for royalties is sought by an unexpected rights-holder.

The license contains a mutual termination clause in the event there is an assertion of a patent interest that would disrupt the licensor's ability to maintain control over the project covered by the open source license. The OSL also makes it clear that this license is a contract. If the license is not accepted as a contract, however, no license is granted under copyright law. This is a departure from the GNU GPL and may be another instance of incompatibility with the GNU GPL under circumstances where the mutual termination clause imposes conditions upon the distribution of open source or free software licensed under the GPL, but distributed along with OSL software. The provision makes clear that the license is immediately terminated upon the licensee's failure to honor the reciprocity obligations of section 1(c).

Finally, the OSL contains a provision under section 9 that provides that any lawsuits relating to this license are to be maintained in the licensor's venue and under the laws of that jurisdiction. This venue clause is a provision that was believed necessary, but was missing from the earlier iterations of the GNU GPL.

Documentation Licenses

The open source model applies to more than software development, and even in the context of software, there has been increasing desire to apply open source standards to documentation of source code as well as to entire books written about software programs. The publisher of this book has been at the forefront of the trend to support open source documentation when doing so is suitable for the objectives of a given project.

According to the FSF, technical writers can earn money by writing free documentation for free software, and that opportunity is why the GNU Free Documentation License (GNU FDL or GFDL) was developed. The GNU FDL is a tool that encourages commercial publishers to fund development of free documentation without surrendering their vital interests.

The GNU FDL secures the interests of publishers through the license's *cover text* terms and through other aspects of the license that involve matters of concern related to documentation or book covers, title pages, history, and endorsements in a manner that makes the license appealing to commercial publishers for books whose authors are paid. To improve the appeal of the GNU FDL, the FSF consulted specifically with staffs of publishing companies, as well as lawyers, free documentation writers, and the community at large, in writing the GNU FDL; notably, this is an ambitious project because the open source and free software community must reshape a social system where publishers are loathe to pay writers for distribution of electronic versions of their written works (e.g., see, *Tasini v. New York Times*), much less pay authors to write commercial free manuals for free software.

The variety and number of commercial open source licenses attests, in part, to the vitality of the commercial space for open source software. SuSE Linux AG, a software company based in Germany, demonstrated its first enterprise corporate desktop designed for large information technology infrastructures: In 2003, the company debuted SuSE Linux Desktop, which retails for approximately $598 for a five-user license and a five-year maintenance contract. SuSE Linux Desktop is built to integrate easily with existing hardware and software, and expectations are that the operating system package will make substantial inroads into the desktop PC marketplace for operating systems as it begins replacing Microsoft's versions of the Windows operating system throughout Europe. StarOffice is another example of a successful commercial open source licensing arrangement.

With StarOffice continuing to increase its end-user base, it is quite possible that this productivity office suite will become one of the most successful open source commercial software ventures ever. StarOffice was acquired from StarDivision, a software development firm in Germany, by Sun Microsystems in 1999. Currently, StarOffice software is managed as an open source project using the OpenOffice.org source code, APIs, file formats, and reference implementation. The OpenOffice.org source code initially includes the technology that Sun Microsystems has been developing for the future versions of StarOffice software. The source is written in C++ and delivers language-neutral and scriptable functionality, including Java APIs. This source technology introduces the next-stage architecture, allowing use of the suite as separate applications or as embedded components in other applications. Numerous other features are also present, including XML-based file formats and other resources.

With regard to software licensing, OpenOffice.org uses a dual-licensing scheme for source code contributions: the GNU Lesser General Public License (LGPL) and Sun Industry Standards Source License (SISSL). The dual-licensing scheme is used to provide open and free access to the technology both for the GPL community and for other developers or companies that cannot use the GPL. In this regard, OpenOffice.org is using dual licensing in the same manner that projects like Perl and Mozilla (MPL) rely upon it. Through the combined use of GPL/LGPL and SISSL, developers will have a high degree of freedom, yet compatibility and interoperability will be preserved. Licensees may freely modify, extend, and improve the OpenOffice.org source code. Dual licensing provides a degree of autonomy in determining whether or not to provide source code and contribute modifications to the community.

Endnotes

[1] Red Hat (www.redhat.com), Debian (www.debian.org), and S.u.S.E. (www.suse.com) sell versions of the Linux operating system.

[2] Although some proprietary developers have begun to refer to open source software as *non-commercial* software, the characterization is obviously inappropriate. Many commercial software vendors distribute open source software.

[3] Sun previously had implemented a third licensing level, called the Internal Deployment license. *See* Jini Network Technology FAQs at http://www.sun.com/jini/faqs/index.html. Since the contractual requirements for the Internal Deployment license and the Commercial Use license were similar, Sun decided to collapse the two licensing levels into one. Those developers still operating under the Internal Use license must accept the terms of the new Sun Community Source License (SCSL) and sign and return a copy of the Commercial Use Supplement and its Technology Specific Attachment. *See* SCSL FAQs at http://www.sun.com/software/communitysource/faq.html. According to some, the Sun license essentially changes its color based on the kind of software in which the Java source code is included. If it is free open source software, it is like an open source license in that it allows source code redistribution without restrictions or fees. If it is commercial software, it is more like a commercial license, with fees payable to Sun.

[4] The Ninth Circuit seemed to have followed this rule in *S.O.S., Inc. v. Payday, Inc.*, 886 F.2d 1081 (9th Cir. 1989).

[5] Subsequently, the NPL and MPL were revised to address concerns about software patents and to ensure that licensees are freely able to distribute software licensed under the NPL or MPL through dual-licensing models.

7

Open Standards and Public Policy

Open Standards

Despite the incredible successes of the free software and open source software communities, open source cannot successfully overcome the threat posed by proprietary software to open access to information. Instead, an essential aspect of open access to information includes the development of new technologies and the spread of our knowledge base through open standards. Without vigilant support of open standards, open source software will inevitably succumb to the market power of closed software and the legal might of patent law.

Establishing open standards in protocols will help preserve the universality and interoperability of information products along with the healthy growth of the public domain. The Internet, thankfully, has not been patented yet, but a trend in that direction is advancing at an alarming pace. There seems to be the constant risk that a software patent might be claimed with respect to a software program—as is the case with regard to the SCO-UNIX dispute—and the devastating result could be that all open source software programs touching upon the patent's claims would be considered infringing; a court would likely block all use of the program and its derivative iterations until or unless a patent licensing fee is paid. One need only imagine the potential impact of a legitimate claim against Linux to understand the potential harmful and disruptive effect software patents could have upon open source developers and end-users. If the vitality of the software patent threat remains unabated, the adverse impact of unwarranted software patents will far exceed disrupting the progress of open source: What is at stake is the pace of technological innovation and its undisputable economic effects.

Universality and interoperability, together with the low cost of Internet access, which are essential for democracy and the economy, are sometimes threatened by software patents that affect Internet standards and the software essential for it to operate. No doubt one of the biggest threats for the open source community is the strategic reactions of Microsoft, which, as a highly successful proprietary software promoter, would like to erode the successes of open source. One area of attack upon the open source community has been Microsoft's urging of proprietary developers to use patent law and side agreements to break open standards.

Open standards have aided in the success of rapid adoption of open source protocols because developers need not waste valuable development time seeking bogus software patent claims. Microsoft may well attempt to utilize its market position to force software vendors to choose between Microsoft and Linux. The tactic of those who privilege proprietary technical standards and aim to subvert all paths toward open technical standards can be successful only if the tactic achieves sufficiently widespread implementation by ensuring that proprietary standards become the de facto standard. Efforts to constrain software developers through the use of Microsoft's market position could prove counterproductive. If proprietary third-party vendors are locked into closed-code de facto standards, the tactic might oddly backfire if the closed-code approach drives all vendor-backed innovation into the arms of the open source community as the only viable forum unfettered by closed-code lock-in tactics.

Open standards are being pursued by the open source community to thwart the effects of an increasing number of patent rights that are attaching to software and other important protocols. Although a complete exposition on standardization of technology is beyond the scope of this chapter, it is important to note that it has become more important today to consider what really constitutes an open standard. In that light, this final section describes what principles should govern an open standards process.

Open standards matter, particularly in the environment of the Internet and regarding software, because the promotion of successful technological innovation is best supported by open standards. Open standards often minimize technological uncertainty, reduce compatibility or interoperability obstacles, and diminish the likelihood that a market-based proprietary protocol will dominate and outlast its technical superiority or advantage.

Open source can support open standards through a licensing and trademark regime. More important than the method, however, is the goal of ensuring that not only is the standard derived as an open standard, but that the standardization process be representative and open as well. In other words, in the context of emerging technologies and innovation, the process or path toward standardization is as important as the standard itself. Therefore, the open standards approach can best retard the effects of software patents and proprietary lock-in on closed standards by deriving standards conducive to the successful development of objectively good standards where information is openly shared. The standardization process should include representation or input from affected interests, procedural safeguards ensuring an opportunity for potentially affected interests to be heard or meaningfully participate, and the organization's standardization process should not exhibit a bias against particular perspectives.

Recognizing that an open standard is more than just a specification, Bruce Perens, president of the OSI, proposed a set of guidelines that the open source community might use to convince Internet technology standards bodies to support open standards and join the community in halting the present push for software patents. The principles behind the open standards, and the practice of offering and operating the standard, are what make the standards open, according to Perens.

Public Policy

To date, no metaphor has proven more singularly pervasive as a concept of cyberspace than the term *cyberspace* [1]. Yet the metaphor (and some of its offspring) has come under increasing attack recently as a word that obscures more than it illuminates [2]. The perception of cyberspace, for example, as a place or space separate from real space, some say, tends to encourage belief in a myth that cyberspace is an actual jurisdiction separate from or independent of the political sphere of real space, which enhances implausible demands that cyberspace should be governed in its own way. Our common conception of computer-mediated technologies springs from our conception of cyberspace, but characterizing and defining cyberspace turns out to be a complex project. This has been particularly true for lawyers.

Cyberspace has made it possible for unprecedented forms of computer-mediated communications. Throughout the history of humankind, never before has there been a means of communication that has provided so many individuals with the ease and ability to engage in interactive communications of the scope and scale supported by cyberspace. The import of the technologies of cyberspace, however, lies not simply in the novel forms and scale of communication. Also of profound significance are the ways in which these communicative possibilities continue to lead to new types of expressions, identities, and social relationships, including relationships between man and machine—even disembodied robots (often called *bots*) interact with humans in cyberspace.

In other words, as it becomes more apparent that software persists in cyberspace in much the same way that it does in our physical world, ultimately, our legal conceptions of computers and software are likely to evolve as our understanding of computer-mediated communications technology continues to unfold in new directions. Our conceptions of software and hardware are beginning to be influenced by cyberspace, and this understanding is quite likely to challenge fundamental principles of law in the same manner that open source software development strains our endeavor to hold onto the unsound principles that infect our application of intellectual property law to software.

As we begin our journey of reflecting upon legal precedent and undergoing a careful thinking-through of computer-mediated communications technology, we will quickly come upon stumbling blocks if we are not disposed to discarding the tendency to impose common conceptions of familiar objects upon innovative computer-mediated technologies that call for equally innovative public policy choices.

A great deal of valuable intellectual effort already has been spent resisting new ideas about how the law ought to shape conduct in cyberspace. It is needless to say, perhaps, that regardless of what public policies are selected or adopted, what springs from our conceptions of cyberspace ought to be guided by continuing and unremitting reflections concerning what exactly is cyberspace and how this dynamic computer-mediated communications technology reinforces or dislodges assumptions about human relations.

For the public policy maker, for example, open source software development poses alternative explanations of human motivation for creative endeavors, which can be ignored or used to augment our public policy choices. These considerations can only spring from a deeper analytical thinking about software—including an appreciation of the significance of why using software is not the same as using it up or why developing a single software program is not the same as producing a single computer or machine.

In order to understand the uses or consequences of a software application in a specific setting, you have to first understand the unique manner that bits become software, which easily flows through a sociotechnical network; in this respect, whether the economics of software production ought to be safeguarded by the identical legal regime used for useful machines should be subject to extensive debate.

In addition, how hackers use software profoundly influences the kinds of uses made of the software's expression and its ideas. Hackers can demonstrate that a wide-ranging network of programmers can develop robust, popular, and reliable software, where source code is open and freely available to the public, not hidden by access controls or technological barriers of any sort. In this respect, certain distinct uses of software might be viewed as, itself, expressive.

More generally, open source programming eschews the use of software development to strategically control markets without regard to the production of superior software applications. In this regard, open source programming may weaken the anticompetitive effects of the proprietary software development practice, wherein developers invest creative efforts in proprietary software design methods in order to create or strengthen position in markets, lock in their technology, and limit consumer choice. Indeed, many commentators have noted that the current intellectual property regime, which has granted Microsoft property rights in its immensely successful operating system software, is a significant factor supporting Microsoft's continued market power in the operating system software market.

Even so, the one obvious, practical, and destabilizing effect of open source projects based on GPLs is the potential for the project to implacably acquire copyright interests in all the works for which the original source code is based. (Genuine public domain projects could not lead to the same result because copyright would not aggregate in one original author.) In this regard, open source could rather, ironically, distort the goals or aims of copyright—since the GPLs do not preclude the original copyright owner or the open source code project itself from altering the terms of the GPL at some later time by forking a successful project into a proprietary economic model—extracting royalty fees from all participants who fail to abandon their free efforts. Moreover, the meaning of *derivative* work may be strained under the terms of the GPL.

The answers to these public policy concerns about computer-mediated communications technologies may differ from social to legal context, perhaps, but assessing the answer is already taking us a step ahead of ourselves, since the question is quite complex and is far more difficult than one may presume.

Endnotes

[1] William Gibson, *Neuromancer* (1984). For Gibson, the term *cyberspace* meant the sense of place created by interaction and communication in a virtual computer environment.

[2] For those aware of the development of Internet and cyberspace law, arguing that cyberspace is a place will appear either ill-informed or quixotic. Even arguing that we think of cyberspace as a place goes against accepted views. For a brief moment, the legal conception of cyberspace as a place flared and then was gone. As a legal argument it peaked around 1996, was attacked soon thereafter, and, as one commentator has noted, by the year 2000 one was hard-pressed to find anyone foolish enough to subscribe to this theory.

8

Rolling Your Own Open Source License

> *Open source licensing illustrates the principle that the right to copy is* left *in place.*

Selecting a License Template

It appears inevitable that a uniform contract law for software and information licensing will be put in place to provide default rules for providers and users of information products. To be genuinely beneficial, however, a uniform law should: (1) affirm the basic principle of freedom of contract, (2) increase contract certainty, (3) be attuned to the unique practices of the affected industries and digital convergence, and (4) allow for innovative methods of distribution. A poorly drafted regulatory statute, a commercial code based on unfair consumer protection rules and proprietary distribution methods, and a patchwork of uncoordinated or intensely contradictory state laws are all probably best left unwritten. In this respect, a caveat to drafting an open source license is to remain aware of the unusually high level of uncertainty in the law today with respect to on-line licensing. Open source software licensors should try to ensure that the selected business model reflects options that can be fairly easily adapted if legislatures or courts impose dramatic or unwelcome changes in the law.

In the Appendix at the end of the book, there are excerpts of interesting open source licenses. These licenses have been included as templates, which may aid license drafters in rolling their open source license. Well-known or widely used licenses occasionally offer the advantages of stability and predictability within the open source community. These templates, therefore, may provide useful license provisions that you may use to express your own objectives, but in a language for which potential licensees are already familiar. Some open source license templates may be drafted less clearly than others or contain inaptly expressed legal terms. Consequently, license drafters should reflect upon comments in earlier chapters of the book before deciding on using common language found in any particular license template.

On December 16, 2002, in an attempt to increase the accessibility of free content as well as enhance the readability of open source licenses, Creative Commons (http://www.creativecommons.com) released its first version of an open source licensing project aimed at helping Internet users who create content and documentation and draft licenses that provide the same or similar freedoms provided by traditional open source licenses. Although some of the issues arising in the context of traditional literary works differ from the concerns of software makers, the Creative Commons project should provide useful assistance with common copyright law matters.

In addition, in selecting an appropriate license, a developer or licensor should carefully review licensing terms before deciding which license best meets the developer's needs; this is particularly relevant before contributing code to an open source project. At a minimum, a license might include a disclaimer of warranties to protect open source contributors from liability. The absence of such a disclaimer could potentially discourage open source developers from contributing code because of the risk of liability. A licensor must also ensure that the license is carefully drafted to grant the licensee the pertinent rights to copy, modify, and distribute the original code, rights normally protected by copyright law. Then the license should be reviewed to ensure that it reflects the developer's open source development model.

Notwithstanding the fact that an objective of this book is to illustrate the types of terms that must be included in a software license in order to prepare a legally enforceable open source license, one might ask whether there already are too many open source licenses. There is a genuine fear among some open source software licensees that license drafting is going to lead to chaos within the open source community. Drafting a software license requires planning and the instances in which programmers continuously envision uses of software that had not been considered by those who have drafted existing licenses leaves little doubt that open source licenses must proliferate.

Software licensing provides familiar benefits for all of those engaged in the transaction and provides an efficient form of computer information or technology transfer; for example, licensors ensure income streams for innovation, licensees use software without development, and the public benefits from the brisk diffusion of innovation enhanced by clear default contracting rules. To be sure that these benefits always inure to the parties enter a licensing agreement, the license should clearly set out the ground rules concerning the risks inherent in software transactions. Therefore, open source licenses should include a provision demonstrating a general consensus on the goals and limitations of the licensing agreement; the consensus could take the form of a preamble or be formulated as a policy statement concerning how the license should be interpreted regarding important specific terms or key provisions.

In addition, the open source licensor must determine what limitations there are, if any, on software enhancements or adaptations. Licensors refusing certain changes submitted by licensees should consider that their right to make or control the making of derivative works may be far from absolute; hence, rejections of code enhancements by licensees should be handled professionally by those participating in developing the community.

Regardless of its form, a license may be found to have implied as well as express terms; for instance, a term frequently implied in licenses that convey source code is a covenant of good faith and fair dealing. Even so, the licensor must take care to note that explicit terms of the license may nullify or alter the usual operation of an implied term; for example, an open source license that explicitly provides for competition in some respects between licensor and licensee, despite access to the licensor's source code, may alter the implied covenant of good faith and fair dealing. Hence, read, read, and reread your drafted licenses to ensure that the license does not contain terms that contradict sublicensing goals.

Generally, software licenses convey or transfer a portion of the copyright interests nonexclusively to another party while retaining the intellectual property rights. Drafting a software license reflecting what is being conveyed requires care, planning, and clear objectives regarding the nature of the interest transferred and the motivation behind the decision to license the software. Although open source software licenses are directed toward fairly specific goals, the same considerations that go into the decision to license any software ought to be evaluated by open source licensors as well.

Depending on the copyright owner's philosophy and objectives, a choice of using an established open source license template to begin the process of license drafting may be made fairly easily. But for those who are less certain about copyleft, the GNU GPL, or whether to support certain commercial uses, dual-licensing objectives, or simply have an unusual or novel business model, the specific decision to roll your own license should be assessed by an attorney familiar with open source software licensing.

The Appendix at the end of the book includes excerpts of relevant provisions of the DMCA and 22 open source license templates. Some of the license templates also contain annotations included after the complete text, which highlight unique provisions or explain a template's unique purpose. The Appendix is also available in hypertext, searchable format on the included CD-ROM. Among the many existing open source license templates included in the Appendix, the notable distinctions involve the following factors:

- *Compatibility.* Some open source license templates are drafted in a manner to ensure that the terms are compatible with the GNU GPL. The advantage of using a license that is compatible with the GNU GPL is that the codebase of the compatible software project may mix with a GNU GPL project without increasing the likelihood that the terms of the highly restrictive GPL will be violated.
- *Permissiveness.* A permissive license generally imposes no conditions on what a user can do with the software, including charging licensees a fee for distribution and imposing no obligation to include source code in the distribution.
- *Control.* Some open source license templates impose a greater degree of control over changes made to the original codebase, which may range from imposing restrictions on public distributions of private modifications to imposing a reporting requirement regarding sending copies of all publicly distributed modifications to the initial developer or copyright holder.
- *Commercial use restriction.* Some open source license templates impose some type of commercial restriction; most notably, some prohibit commercial use of the software by licensees. There is a strong likelihood that many commercial use restrictions are not compatible with the open source definition. Hence, these licenses may not be genuine open source licenses at all.
- *Copyleft.* Some open source license templates comply with the open source definition, but do not impose a copyleft condition upon licensees.

Software Licensing Law and Policy

The law of copyright provides software authors with an effective set of legal tools to protect the their interest in avoiding damage to their work from copyright infringement. In this light, the use of a software license in addition to the default rules provided by copyright law make it appear that software authors are piling on privileges that far exceed the interests of authors that lawmakers are assumed to have carefully considered in enacting copyright legislation.

Some commentators have argued that by issuing written software licenses that bind end-users who may never read the license or who may never understand the restrictions imposed on them by the license terms, software authors are undermining the public policy goals effectuated by copyright law; notably, however, most commentators do not direct their criticisms to all forms of software licensing. Copyright law itself specifically provides for the use of copyright licenses under particular circumstances. More to the point, an open source license should neither intend nor have the effect of imposing onerous restrictions on end-users.

As noted in Chapter 5, the propertycentric legal regime that reinforces the software developer's business practice of imposing restrictions on how a computer user may use a software program ostensibly arises from both copyright law and the law of contracts. The copyright holder offers terms regarding permissible copyright uses of the software, and the end-user accepts those terms by purchasing the software, using the software, and/or consenting to comply with the terms of the license by some explicit act manifesting consent. If the end-user breaches the license by failing to comply with a term or provision of the license, the copyright holder may file a lawsuit in federal court for copyright infringement unless the dispute is settled or the end-user otherwise satisfies the copyright holder that his copyright interests are protected.

In this respect, a software license is the legal instrument for a *copyright license.* Generally, it provides no contract remedy for the copyright holder. As long as the software license is limited to statutory copyright interests, then contract remedies are not applicable to the software author. There might be another legal basis that the licensor may be able to rely upon, such as a trademark interest, a patent interest, or some other relevant independent statutory remedy, but, traditionally, a breach of contract remedy will be unavailable. If, however, most proprietary software licenses never exceeded the scope of a copyright license, the open source movement may not have experienced the remarkable growth it has encountered.

The long-standing use of contracts to transact business or engage in commercial activity that involves the selling and buying of goods is supported by a well-established legal regime governing the law of contracts. Distributing software through the use of licenses, however, is not only a comparatively innovative way to provide customers with an essentially information product, but the practice actually has had meager legal support until very recently.

In fact, the legal question of whether the distribution of software actually constitutes a *sale*—in the sense of a car purchase—or a *license*—in the sense of a car lease—is not settled. Perhaps, quite unremarkably, courts have resolved conflicts over the validity of software licenses in inconsistent ways; and to some extent, whether you view the purchase of software as a sale of a product or a license to use information may depend on whether you are the buyer or the seller.

When it comes to drafting an open software license, one rule to use as a guide is: Do not reinvent the wheel. Reinventing the wheel is not an innovation, and that often suffices for why it should not be done, but in the case of legal instruments, the more important reason is that, generally, the law privileges what is dependable—the old and reliable. As noted, use clearly expressed, well-known, and well-understood terms in your license. Begin with an established license template listed in the Appendix at the end of this book or posted on the Web sites of the FSF (http://www.fsf.org) or the OSI (http://www.opensource.org). Even these three sources have only a microcosm of the open source or free software licenses used to distribute open source software.

Open Source Benefits Are Free

There are no guarantees in this business. Open source licensing brings to the end-user tremendous benefits that do not accompany proprietary software licensing. Still, open source licensors must be careful, thoughtful, and respectful intellectual property users because there remains a minefield of legal challenges that could arise due to the increasing popularity of open source licensing. Despite its rapid adoption by many programmers, open source is still a novel exercise, and there are numerous unsettled issues regarding the interplay between the Copyright Act and licensing.

Although what may constitute an authentic or approved open source/free software license is occasionally a matter of vigorous debate within the open source community, there are some licenses that represent generally accepted good examples of an open source license template; many of which, but not all, have been mentioned in this book. The Appendix contains the text excerpts of popular or accepted open source/free software license agreements, along with an Internet or Web site address where a current version of the license may be found. An elliptical mark (…) denotes license provisions that were not reprinted, either because they have been discussed extensively in other sections of this book or because the provisions were boilerplate passages that were not relevant to open source licensing.

You should visit each Web site that interests you to appreciate the many different uses and formalities that accompany open source licensing and to determine which styles may best suit your own future needs. The inclusion or exclusion of any license in this book or the Appendix does not constitute the author's approval or disapproval of the licensing objectives or open source business model of the software developers associated with any given license. Indeed, it is well worth repeating that as you draft your license, no license should be used by an adherent or member of the open source community without the advice and consult of a competent lawyer. Some well-regarded open source/free software licenses may contain deficiencies that could become paramount obstacles to legal enforcement, depending upon a given business model or software development method; many of these issues have been highlighted in this book, but the seeming inevitable promulgation of open source licenses has made this task a highly arbitrary endeavor. Caveat lector.

Appendix: Selected Open Source License Templates and Copyright Law Provisions

This CD-ROM contains the following rich text format RTF files, which correspond to a selected list of open source license templates or a copyright law provision pertinent to open source licensing. Each filename listed below is hyperlinked; please double-click on the filename to load the pertinent file.

Using the Hyperlinked RTF Documents

To use the RTF documents, double-click on a filename (in Windows Explorer, My Computer, File Manager, etc.) and your default word processor will then load the document. All modern word processors will automatically import RTFfiles. If you need support for this Appendix, please point your Web browser to the author's Web site at: www.cyberspaces.org/opensource/ or contact the publisher at www.artechhouse.com. Annotations follow most licenses unless the annotation would be largely repetitious or the terms are self-explanatory.

List of License Templates and File Names

License Section Number	Template Name	File Name	File Type
§1.01	GNU General Public License Template	LICENSE 1.01	RTF
§ 2.01	GNU Lesser General Public License Template	LICENSE 2.01	RTF
§ 3.01	Sun Community Source License (v.3) Template	LICENSE 3.01	RTF
§ 4.01	OpenIPCore Hardware General Public License Template	LICENSE 4.01	RTF
§ 5.01	IBM Public License (v.1) Template	LICENSE 5.01	RTF

License Section Number	Template Name	File Name	File Type
§ 6.01	Common Public License (v.0.5) Template	LICENSE 6.01	RTF
§ 7.01	The BSD License Template	LICENSE 7.01	RTF
§ 8.01	X Consortium License Template	LICENSE 8.01	RTF
§ 9.01	Artistic License Template	LICENSE 9.01	RTF
§ 10.01	Mozilla Public License Template	LICENSE 10.01	RTF
§ 11.01	The Q Public License Template	LICENSE 11.01	RTF
§ 12.01	The Apple Public Source License (v.1.2) Template	LICENSE 12.01	RTF
§ 13.01	The Python Copyright License Template	LICENSE 13.01	RTF
§ 14.01	The Jabber Open Source Licensee Template	LICENSE 14.01	RTF
§ 15.01	Amulet – CMU License (v.3.0) Template	LICENSE 15.01	RTF
§ 16.01	Red Hat eCos Public License (v.1.1) Template	LICENSE 16.01	RTF
§ 17.01	Eiffel Forum License (v.1.0) Template	LICENSE 17.01	RTF
§ 18.01	License Agreement for Mozart, an Implementation of Oz 3 - Template	LICENSE 18.01	RTF
§ 19.01	Lucent Technologies, Inc. Plan 9 Open Source License Agreement Template	LICENSE 19.01	RTF
§ 20.01	The LaTex Project Public License Template	LICENSE 20.01	RTF
§ 21.01	The Open Software License Template (Licensor’s Perspective)	LICENSE 21.01	RTF

License Section Number	Template Name	File Name	File Type
§ 22.01	The GNU Free Documentation License Template	LICENSE 22.01	RTF
	SELECTED PROVISIONS OF THE DIGITAL MILLENNIUM COPYRIGHT ACT (DMCA)		
§ 1201(a)	Anti-circumvention of copyright protection systems	Section 1201(a)	
§ 1201(b)	Anti-distribution of copyright circumvention systems	Section 1201(b)	
§ 1201(F)	Reverse Engineering	SECTION 1201(F)	
§ 1203	Civil Remedies	SECTION 1203	
§ 1204	Criminal offenses and penalties	SECTION 1204	

OPEN SOURCE SOFTWARE LAW

§ 1.01 GNU GENERAL PUBLIC LICENSE

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

Website: http://www.gnu.org/licenses/gpl.html

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.

§ 1.02 GNU GENERAL PUBLIC LICENSE – Application of new terms

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.

….

ANNOTATION:

The most salient aspect of the GNU General Public License when compared with other open source licenses is that this is a license written as a copyleft open source software license, which means that each iteration of a modified work derived under the permissions set out by this license renders the source code GPL-covered. The software program must be free software, even if it is put in a separate file. The code must be available for use in free software and not for use in proprietary software, in order to encourage other people who write software to make it free as well.

Section 2 of the GPL does not require the licensee to publicly release a modified version. Users are free to make modifications and use them privately, without ever releasing them. This applies to organizations (including companies), too; an organization can make a modified version and use it internally without ever releasing it outside the organization.

But *if* the licensee does publicly release the modified version, section 2 of the GPL requires the licensee to make the modified source code available to the program's users under the GPL.

Section 2 contains the well-known *copyleft* provision, which says that anyone who receives a copy of your version from the licensee has the right to redistribute copies (modified or not) of that version. It does not give the licensee permission to distribute the work on any more restrictive basis. The GPL allows everyone to do this. Section 2 says that modified versions you distribute must be licensed to all third parties under the GPL. All third parties means everyone--but this does not require the licensee to do anything other than extend a license under the GPL, for your version of the software.

Implicitly, the right to sell copies is part of the definition of free software. Except in one special situation, there is no limit on what price the licensee can charge. (The one exception is the required written offer to provide source code that must accompany binary-only release.)

The licensee is allowed to sell copies of the modified program commercially, but only under the terms of the GNU GPL. Thus, for instance, the licensee must make the source code available to the users of the program as described in the GPL, and they must be allowed to redistribute and modify it as described in the GPL. These requirements are the condition for including the GPL-covered code the licensee received in a program of your own.

If the program dynamically links plug-ins, and they make function calls to each other and share data structures, the program will form a single program according to the drafters of the GPL; in this respect, plug-ins must be treated as extensions to the original program, which essentially means the plug-ins must be released under the GPL or a GPL-compatible free software license, and that the terms of the GPL must be followed when those plug-ins are distributed.

If the program dynamically links plug-ins, but the communication between them is limited to invoking the 'main' function of the plug-in with some options and waiting for it to return, that is a borderline case.

Under section 2(c), mere aggregation of two programs means putting them side by side on the same CD-ROM or hard disk. In other words, where there are separate programs, not parts of a single program, if one of the programs is covered by the GPL, it has no effect on the other program.

Combining two modules means connecting the modules so that they form a single program. If either part is covered by the GPL, the whole combination must also be released under the GPL--if the licensee can't, or won't, do that, the licensee may not combine them.

What constitutes combining two parts into one program? This is a legal question, which ultimately judges will decide. But, the drafters of the GPL opine that if the modules are included in the same executable file, they are definitely combined in one program. If modules are designed to run linked together in a shared address space, it likely means combining them into one program. By contrast, pipes, sockets and command-line arguments are communication mechanisms normally used between two separate programs. So when they are used for communication, the modules normally are separate programs.

To be in the best position to enforce the GPL in court, the license drafters of the GNU GPL want to keep the copyright status of programs licensed under the GPL as simple as possible. For them, "simple" means asking each contributor to either assign the copyright on his contribution to the FSF, or disclaim copyright on it and thus put it in the public domain. A word of caution if your goal is to genuinely use a preexisting open source license as a template for your own uses: developers do not really "use" the GPL, but, rather, they comply with it. But, the licensee can use the GPL terms (possibly modified) in another license provided that the licensee call your license by another name and do not include the GPL preamble, and provided you modify the instructions-for-use at the end enough to make it clearly different in wording and not mention GNU (though the actual procedure you describe may be similar).

As sections 3 and 7 of the GPL provide, a system incorporating a GPL-covered program is an extended version of that program. The GPL says that any extended version of the program must be released under the GPL if it is released at all. This is for two reasons: to make sure that users who get the software get the freedom they should have, and to encourage people to give back improvements that they make.

Generally, the meaning of "incorporating" a GPL-covered software is not entirely clarified by the GPL, but standardized programming practices might provide the best guidance on what is meant. If the two programs are combined so that they become effectively two parts of one program, then the licensee cannot treat the resulting program as two separate programs.

If two "programs" remain technically separated, such as in the manner that a compiler is technically distinct from the kernel of an operating system, then the licensee should treat the two "programs" as separate programs.

§ 2.01 GNU Lesser General Public License

Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc. [This is the first released version of the Lesser GPL. It also counts as the successor of the GNU Library Public License, version 2, hence the version number 2.1.]

Website: http://www.gnu.org/licenses/lgpl.html

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, 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 and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

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

We call this license the “Lesser” General Public License because it does Less to protect the user’s freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users’ freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, 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 library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete 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 distribute a copy of this License along with the Library.

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 Library or any portion of it, thus forming a work based on the Library, 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) The modified work must itself be a software library.

b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, 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 Library, 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 Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you 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.

If distribution of 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 satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a “work that uses the Library”. Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a “work that uses the Library” with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a “work that uses the library”. The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a “work that uses the Library” uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a “work that uses the Library” with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer’s own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable “work that uses the Library”, as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be 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.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library 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.

9. 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 Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library 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 with this License.

11. 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 Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library 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 Library.

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.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library 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.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser 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 Library 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 Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, 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

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "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 LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. 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 LIBRARY 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 LIBRARY (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 LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

§ 2.02 GNU LESSER GENERAL PUBLIC LICENSE – Application of new terms

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. 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 library's name and an idea of what it does.

Copyright (C) year name of author

….

ANNOTATION:

The relevant respects that the GNU LGPL differs from the GNU GPL involve the limited purpose of the GNU LGPL, which is to permit linking to uniquely important libraries that are part of or constitute non-free programs; otherwise, licensees are cautioned to use the GNU GPL by the drafter of this license. The Lesser General Public License permits more lax criteria for linking other code with the library.

§ 3.01 Sun Community Source License, Version 3.0 (the SCSL)

Website: http://www.sun.com/jini/licensing/SCSL3_JiniTSA1.html

I. DEFINITIONS…

II. PURPOSES

Original Contributor is licensing the Reference Code and Technology Specifications and is permitting implementation of Technology under and subject to this Sun Community Source License (the "License") to promote research, education, innovation and product development using the Technology.

COMMERCIAL USE AND DISTRIBUTION OF TECHNOLOGY IS PERMITTED ONLY UNDER OPTIONAL SUPPLEMENTS TO THIS LICENSE.

III. RESEARCH USE RIGHTS

A. From Original Contributor. Subject to and conditioned upon Your full compliance with the terms and conditions of this License, including Sections IV (Restrictions and Community Responsibilities) and V.E.7 (International Use), Original Contributor:

1. Grants to You a non-exclusive, worldwide and royalty-free license to the extent of Original Contributor's copyrights and trade secret rights in and covering the Reference Code and Technology Specifications to do the following for Your Research Use only:
a) reproduce, prepare derivative works of, display and perform the Reference Code, in whole or in part, alone or as part of Covered Code;
b) reproduce, prepare derivative works of and display the Technology Specifications;
c) distribute source or object code copies of Reference Code, in whole or in part, alone or as part Covered Code, to other Community Members or to students; and
d) distribute object code copies of Reference Code, in whole or in part, alone or as part of object code copies of Covered Code, to third parties.

2. will not, during the term of Your License, bring against You any claim alleging that Your using, making, having made, importing or distributing Community Code for Your Research Use, insofar as permitted under Section III.A.1 of this License, necessarily infringes any patent now owned or hereafter acquired by Original Contributor whose claims cover subject matter contained in or embodied by the Reference Code or which would necessarily be infringed by the use or distribution of any and all implementations of the Technology Specifications.

3. grants to You a non-exclusive, worldwide and royalty-free license, to the extent of its intellectual property rights therein, to use (a) Original Contributor's class, interface and package names only insofar as necessary to accurately reference or invoke Your Modifications for Research Use, and (b) any associated software tools, documents and information provided by Original Contributor at the Technology Site for use in exercising the above license rights.

B. Contributed Code. Subject to and conditioned upon compliance with the terms and conditions of this License, including Sections IV (Restrictions and Community Responsibilities) and V.E.7 (International Use), each Community Member:

1. grants to each Community Member a non-exclusive, worldwide and royalty-free license to the extent of such Community Member's copyrights and trade secret rights in and covering its Contributed Code, to reproduce, modify, display and distribute Contributed Code, in whole or in part, in source code and object code form, to the same extent as permitted under such Community Member's License with Original Contributor (including all supplements thereto).

2. will not, during the term of the Community Member's License, bring against any Community Member any claim alleging that using, making, having made, importing or distributing Contributed Code as permitted under this License (including any supplements) infringes any patents or patent applications now owned or hereafter acquired by such Community Member which patents or patent applications are infringed by using, making, having made, selling, offering for sale, importing or otherwise transferring the Contributed Code ("Community Member Patents"). This covenant shall apply to the combination of the Contributed Code with other Covered Code if, at the time the Contributed Code is posted, such addition of the Contributed Code causes such combination to be covered by the Community Member Patents. The covenant shall not apply to any other combinations which include the Contributed Code or to the use or distribution of modified Contributed Code where the modifications made by the Community Member add to the functions performed by the Contributed Code in question and where, in the absence of such modifications, there would be no infringement of a Community Member Patent.

3. grants to Original Contributor, in addition to the rights set forth in Sections III.B.1 and III.B.2, the right to sublicense all such rights in Contributed Code, in whole or in part, as part of Reference Code or other technologies based in whole or in part on Reference Code or Technology and to copy, distribute, modify and prepare derivative works of Contributed Code Specifications, in whole or in part, in connection with the exercise of such rights.

C. Subcontracting. You may provide Covered Code to a contractor for the sole purpose of providing development services exclusively to You consistent with Your rights under this License. Such Contractor must be a Community Member or have executed an agreement with You that is consistent with Your rights and obligations under this License. Such subcontractor must assign exclusive rights in all work product to You. You agree that such work product is to be treated as Covered Code.

D. No Implied Licenses. Neither party is granted any right or license other than the licenses and covenants expressly set out herein. Other than the licenses and covenants expressly set out herein, Original Contributor retains all right, title and interest in Reference Code and Technology Specifications and You retain all right, title and interest in Your Modifications and associated specifications. Except as expressly permitted herein, You must not otherwise use any package, class or interface naming conventions that appear to originate from Original Contributor.

IV. RESTRICTIONS AND COMMUNITY RESPONSIBILITIES

As a condition to Your license and other rights and immunities, You must comply with the restrictions and responsibilities set forth below, as modified or supplemented, if at all, in Attachment B, Additional Research Use Terms and Conditions.

A. Source Code Availability. You must provide source code and any specifications for Your Error Corrections to Original Contributor as soon as practicable. You may provide other Contributed Code to Original Contributor at any time, in Your discretion. Original Contributor may, in its discretion, post Your Contributed Code and Contributed Code Specifications on the Technology Site. You may post Your Contributed Code and/or Contributed Code Specifications on another website of Your choice; provided, source code of Community Code and Technology Specifications must be provided to Community Members only and only following certification of Community Member status as required under Section IV.D.

B. Notices. You must reproduce without alteration copyright and other proprietary notices in any Covered Code that You distribute. The statement, "Use and Distribution is subject to the Sun Community Source License available at http://sun.com/software/communitysource" must appear prominently in Your Modifications and, in all cases, in the same file as all Your copyright and other proprietary notices.

C. Modifications. You must include a diff file with Your Contributed Code that identifies and details the changes or additions You made, the version of Reference Code or Contributed Code You used and the date of such changes or additions. In addition, You must provide any Contributed Code Specifications for Your Contributed Code. Your Modifications are Covered Code and You expressly agree that use and distribution, in whole or in part, of Your Modifications shall only be done in accordance with and subject to this License.

D. Distribution Requirements. You may distribute object code of Covered Code to third parties for Research Use only pursuant to a license of Your choice which is consistent with this License. You may distribute source code of Covered Code and the Technology Specifications for Research Use only to (i) Community Members from whom You have first obtained a certification of status in the form set forth in Attachment A-1, and (ii) students from whom You have first obtained an executed acknowledgment in the form set forth in Attachment A-2. You must keep a copy of each such certificate and acknowledgment You obtain and provide a copy to Original Contributor, if requested.

E. Extensions.

1. You may create and add Interfaces but, unless expressly permitted at the Technology Site, You must not incorporate any Reference Code in Your Interfaces. If You choose to disclose or permit disclosure of Your Interfaces to even a single third party for the purposes of enabling such third party to independently develop and distribute (directly or indirectly) technology which invokes such Interfaces, You must then make the Interfaces open by (a) promptly following completion thereof, publishing to the industry, on a non-confidential basis and free of all copyright restrictions, a reasonably detailed, current and accurate specification for the Interfaces, and (b) as soon as reasonably possible, but in no event more than thirty (30) days following publication of Your specification, making available on reasonable terms and without discrimination, a reasonably complete and practicable test suite and methodology adequate to create and test implementations of the Interfaces by a reasonably skilled technologist.

2. You shall not assert any intellectual property rights You may have covering Your Interfaces which would necessarily be infringed by the creation, use or distribution of all reasonable independent implementations of Your specification of such Interfaces by a Community Member or Original Contributor. Nothing herein is intended to prevent You from enforcing any of Your intellectual property rights covering Your specific implementation of Your Interfaces or functionality using such Interfaces other than as specifically set forth in this Section IV.E.2.

V. GOVERNANCE.

A. License Versions.

Only Original Contributor may promulgate new versions of this License. Once You have accepted Reference Code, Technology Specifications, Contributed Code and/or Contributed Code Specifications under a version of this License, You may continue to use such version of Reference Code, Technology Specifications, Contributed Code and/or Contributed Code Specifications under that version of the License. New code and specifications which You may subsequently choose to accept will be subject to any new License in effect at the time of Your acceptance of such code and specifications.

B. Disclaimer Of Warranties.

1. COVERED CODE, ALL TECHNOLOGY SPECIFICATIONS AND CONTRIBUTED CODE SPECIFICATIONS ARE PROVIDED “AS IS”,

WITHOUT WARRANTIES OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT ANY SUCH COVERED CODE, TECHNOLOGY SPECIFICATIONS AND CONTRIBUTED CODE SPECIFICATIONS ARE FREE OF DEFECTS, MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGING OF THIRD PARTY RIGHTS. YOU AGREE THAT YOU BEAR THE ENTIRE RISK IN CONNECTION WITH YOUR USE AND DISTRIBUTION OF ANY AND ALL COVERED CODE, TECHNOLOGY SPECIFICATIONS AND CONTRIBUTED CODE SPECIFICATIONS UNDER THIS LICENSE. NO USE OF ANY COVERED CODE, TECHNOLOGY SPECIFICATIONS OR CONTRIBUTED CODE SPECIFICATIONS IS AUTHORIZED EXCEPT SUBJECT TO AND IN CONSIDERATION FOR THIS DISCLAIMER.

2. You understand that, although each Community Member grants the licenses set forth in the License and any supplements hereto, no assurances are provided by any Community Member that Covered Code or any specifications do not infringe the intellectual property rights of any third party.

3. You acknowledge that Reference Code and Technology Specifications are neither designed nor intended for use in the design, construction, operation or maintenance of any nuclear facility.

C. Limitation Of Liability.

1. Infringement. Each Community Member disclaims any liability to all other Community Members for claims brought by any third party based on infringement of intellectual property rights. Original Contributor represents that, to its knowledge, it has sufficient copyrights to allow You to use and distribute the Reference Code as herein permitted (including as permitted in any Supplement hereto) and You represent that, to Your knowledge, You have sufficient copyrights to allow each Community Member and Original Contributor to use and distribute Your Shared Modifications and Error Corrections as herein permitted (including as permitted in any supplements to the License). You agree to notify Original Contributor should You become aware of any potential or actual infringement of the Technology or any of Original Contributor's intellectual property rights in the Technology, Reference Code or Technology Specifications.

2. Suspension. If any portion of, or functionality implemented by, the Reference Code, Technology or Technology Specifications becomes the subject of a claim or threatened claim of infringement ("Affected Materials"), Original Contributor

may, in its unrestricted discretion, suspend Your rights to use and distribute the Affected Materials under this License. Such suspension of rights will be effective immediately upon Original Contributor's posting of notice of suspension on the Technology Site. Original Contributor has no obligation to lift the suspension of rights relative to the Affected Materials until a final, non-appealable determination is made by a court or governmental agency of competent jurisdiction that Original Contributor is legally able, without the payment of a fee or royalty, to reinstate Your rights to the Affected Materials to the full extent contemplated hereunder. Upon such determination, Original Contributor will lift the suspension by posting a notice to such effect on the Technology Site. Nothing herein shall be construed to prevent You, at Your option and expense, and subject to applicable law and the restrictions and responsibilities set forth in this License and any Supplements, from replacing Reference Code in Affected Materials with non-infringing code or independently negotiating, without compromising or prejudicing Original Contributor's position, to obtain the rights necessary to use Affected Materials as herein permitted.

3. Disclaimer. ….

D. Termination.

You may terminate this License at any time by notifying Original Contributor in writing.

All Your rights will terminate under this License if You fail to comply with any of the material terms or conditions of this License and do not cure such failure in a reasonable period of time after becoming aware of such noncompliance.

If You institute patent litigation against a Community Member with respect to a patent applicable to Community Code, then any patent licenses or covenants granted by such Community Member to You under this License shall terminate as of the date such litigation is filed. In addition, if You institute patent litigation against any Community Member or Original Contributor alleging that Reference Code, Technology or Technology Specifications infringe Your patent(s), then the rights granted to You under Section III.A above will terminate.

Upon termination, You must discontinue all uses and distribution of Community Code, except that You may continue to use, reproduce, prepare derivative works of, display and perform Your Modifications, so long as the license grants and covenants of this license are not required to do so, for purposes other than to implement functionality designated in any portion of the Technology Specifications. Properly granted sublicenses to third parties will survive termination.

Provisions which, by their nature, should remain in effect following termination survive.

E. Miscellaneous.

1. Trademark. You agree to comply with Original Contributors Trademark & Logo Usage Requirements, as modified from time to time, available at the Technology Site. Except as expressly provided in this License, You are granted no rights in or to any Sun, Jini, Jiro or Java trademarks now or hereafter used or licensed by Original Contributor (the "Sun Trademarks"). You agree not to (a) challenge Original Contributor's ownership or use of Sun Trademarks; (b) attempt to register any Sun Trademarks, or any mark or logo substantially similar thereto; or (c) incorporate any Sun Trademarks into Your own trademarks, product names, service marks, company names or domain names.

2. Integration and Assignment. Original Contributor may assign this Research Use License to another by written notification to the other party. This License represents the complete agreement of the parties concerning the subject matter hereof.

3. Severability. If any provision of this License is held unenforceable, such provision shall be reformed to the extent necessary to make it enforceable unless to do so would defeat the intent of the parties, in which case, this License shall terminate.

4. Governing Law. This License is governed by the laws of the United States and the State of California, as applied to contracts entered into and performed in California between California residents. The United Nations Convention on Contracts for the International Sale of Goods shall not apply. Nor shall any law or regulation which provides that a contract be construed against the drafter.

5. Dispute Resolution….

ANNOTATIONS:

Sun has distinguished its license from other open source licenses by calling it a "community source" license. The community source development model differs from the open source development model by requiring compatibility among released versions of the software, and allowing proprietary modifications and extensions to the software.

Sun created community source development principles to blend the best aspects of the proprietary and open source license models. Its primary goal in implementing this model is to balance the need of an organization to innovate rapidly while maintaining proprietary advantages.

By keeping control over the source code, Sun ensures compatibility among the varying versions of software. The license requires commercial distributors to use relatively recent upgraded code and to pass the test suites.

The Sun Public License is a preferred open source template, not the SCSL. The Sun Public License, however, is essentially the same as the Mozilla Public License.

Sun Microsystems has released some of its core software technologies under a new license, the Sun Community Source License (SCSL). This is not an open source license; it lacks essential freedoms such as publication of modified versions of the original software. Even so, it has become a popular open source template, and its provisions are closely associated with the principles of open source. Consequently, if the license drafter pays special care to improve upon the problem areas of the SCSL, it may form the basis of a useful template. Some of the most widely apparent problems with viewing the SCSL as a genuine open source license, include:

Section IV's requirement that all code modifications is sent to Sun *before and if* they are to be distributed.

Section IV's requirement that Sun decides if which modifications are publicly distributed, rather than the appropriate licensee.

The general requirement that licensees comply with an extremely restrictive condition noted in sections III and IV; namely, that if a licensee agrees to the SCSL and downloads Sun code, the licensee seems to be prohibited from using the source code covered by the license with a similar technology or with a different open source license. If these restrictions are removed, the template could be improved.

§ 4.01 OpenIPCore Hardware General Public License "OHGPL"

Draft Version 0.20-15092000 September 2000 Copyright (C) 2000 OpenIPCore Organization.

Website: http://www.opencores.org/OIPC/OHGPL.shtml

1. This license applies to hardware designs, hardware design description, CAD or Fabrication files or any other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this License.

2. You may copy, distribute and/or implement this Hardware Design or any portion of it as is. Any time you copy or distribute this design you have to provide all of the source files and documentations that came with the original work or put them in a public place that anyone can reach without any kind of restrictions.

3. You can not sell the design description, design files or fabrication files but you may charge fee for the physical act of transferring a copy.

4. You can implement the design and charge fees for the physical hardware and you have to provide notice for the public about the source of the design description.

5. Any modification of this hardware design or any derivative work from it should be documented by providing list of changes, reasons behind the changes and the date of change.

6. You are allowed to use the design or design files on any work based on the hardware design.

7. You may not copy, modify, sublicense, or distribute the design/design description or files except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the design/design description or files 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.

8. Each time you redistribute the design description or files, 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.

9. 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 hardware design or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying, distributing or implementing the hardware design (or any work based on the hardware design), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the hardware design or works based on it.

10. NO WARRANTY of any kind is provided on the functionality, performance or risks cased by using this Hardware Design….

ANNOTATIONS:

The main goals of the OpenIPCore (or open source IP core technologies) is to introduce a license for open free hardware designs and cores... one of the basic objectives of the Free open designs is to let everyone contribute to the original design and return the changes back to the original design. This concept should be preserved by some kind of protection schemes. As a result the OpenIPCore group suggested several schemes of protection.

The license Objectives include:

Granting rights to allow anyone permission to manufacture the hardware design

Granting rights to allow anyone permission to customize the hardware design for his/her needs

Granting rights to allow anyone permission to transfer and modify the hardware design.

Granting rights to allow anyone permission to make all changes and modifications to the original work as well as make those modifications available for anyone.

The license restricts the commercial income of the hardware products

The license should be applied to new works and derivative works

§ 5.01 IBM Public License Version 1.0

Website: http://www-124.ibm.com/developerworks/oss/

Definitions….

GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

a) it complies with the terms and conditions of this Agreement; and

b) its license agreement:

i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;

ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;

iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and

iv) states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:

a) it must be made available under this Agreement; and

b) a copy of this Agreement must be included with each copy of the Program.

Each Contributor must include the following in a conspicuous location in the Program:

Copyright © {date here}, International Business Machines Corporation and others. All Rights Reserved.

In addition, each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows subsequent Recipients to identify the originator of the Contribution.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: a) promptly notify the Commercial Contributor in writing of such claim, and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this

Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against a Contributor with respect to a patent applicable to software (including a cross-claim or counterclaim in a lawsuit), then any patent licenses granted by that Contributor to such Recipient under this Agreement shall terminate as of the date such litigation is filed. In addition, If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

....

§ 6.01 Common Public License Version 0.5

Website: http://www-124.ibm.com/developerworks/oss/license-cpl.html

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

1. DEFINITIONS

"Contribution" means:

a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and

b) in the case of each subsequent Contributor:

i) changes to the Program, and

ii) additions to the Program;

where such changes and/or additions to the Program originate from and are distributed by that particular Contributor. A Contribution 'originates' from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include additions to the Program which: (i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Program.

"Contributor" means any person or entity that distributes the Program.

"Licensed Patents" mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement, including all Contributors.

2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

a) it complies with the terms and conditions of this Agreement; and

b) its license agreement:

i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;

ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;

iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and

iv) states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:

a) it must be made available under this Agreement; and

b) a copy of this Agreement must be included with each copy of the Program.

Contributors may not remove or alter any copyright notices contained within the Program.

Each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows subsequent Recipients to identify the originator of the Contribution.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: a) promptly notify the Commercial Contributor in writing of such claim, and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against a Contributor with respect to a patent applicable to software (including a cross-claim or counterclaim in a lawsuit), then any patent licenses granted by that Contributor to such Recipient under this Agreement shall terminate as of the date such litigation is filed. In addition, If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. IBM is the initial Agreement Steward. IBM may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.

....

§ 7.01 THE BSD License

Website: http://www.freebsd.org/copyright/freebsd-license.html

Copyright 1994-2002 FreeBSD, Inc. All rights reserved.

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

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE FREEBSD PROJECT "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE FREEBSD PROJECT OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR

BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

OR

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

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of the <ORGANIZATION> nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

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

ANNOTATIONS:

It's notable that the BSD license does not compel open source users to distribute the source code at all. Thus, there could be legitimate BSD-licensed code for which the source is not distributed and it would not be Open Source. In this respect, the OSD adds a veneer of additional restrictions on top of the licenses.

The BSD license grants unlimited rights to use and distribute modified or unmodified source and binary code, provided that the licensee agrees to keep intact the copyright notices, licensing terms, and disclaimer of warranties with each distribution of the source code as well as agrees to reproduce the copyright notices, licensing terms, and disclaimer of warranties in the documentation and other materials distributed with the code.

The BSD license restricts licensees from using the name of the university, or the names of its contributors, to endorse or promote derivative works without prior permission.

The BSD license does not require that any modifications or enhancements of the original code be contributed back to the open source community. The BSD license does not contain any provisions that keep licensees from making the code proprietary.

The BSD license may be an appropriate option for commercial entities because it avoids the "tainting effect" of the GPL and provides licensees the option of making their enhancements and modifications proprietary.

"Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met." The preceding is a provision of the BSD License template that constitutes the grant rights clause. This is the most important provision of any open source license, and it should reflect that importance by being clearly expressed and precisely written.

The BSD grant rights provision grants to the licensee the right to publicly distribute the copyright-protected work or any modification of it. It also grants the licensee the right to "use" the work.

The BSD license template also provides that "redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution." The two preceding provisions contain conditions that apply to source code and binary code distributions, respectively. The conditions are: (1) pass on the warranty disclaimer to subsequent licensees, (2) and ensure that the copyright notice is retained. Generally, these two provisions along with the absence of a copyleft provision anywhere in the license amount to what really makes the BSD License unique. In this regard, it is apparent that the BSD License is an open source license that is intended to fulfill two objectives: to offer licensees the opportunity to participate in typical open source development efforts and to offer software developers who might not otherwise participate in the open source community an opportunity use the work of open source as well as contribute to open source without being precluded from also using the same work in a closed or proprietary software development business venture.

§ 8.01 X Consortium License

Website: http://www.opensources.org

Copyright (c) <year> <copyright holders>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

ANNOTATIONS:

X.org, a non-profit, international consortium responsible for standards requirements in the X Window System. Source code made available under the X license, will provide open source software developers with an easy way to design applications which operate in virtually all national language environments and among the operating systems that utilize X-windowing technologies, such as Linux and Unix variants.

To date, many open source applications could not reach international markets because applications have needed additional customized code for each spoken language and locale. The Solaris Operating Environment X internationalization (X I18n) technology provides a clear and concise framework of rules that makes it easier for developers to write global software. Its support for 37 languages and 123 locales, including languages involving Complex Text Layout such as Arabic, Hebrew and Thai, has made the Solaris Operating Environment internationalization technology the leading environment for global application development. The decision to release the Solaris Operating Environment X internationalization enhancements under a commonly recognized open source license, emphasizes Sun's ongoing desire to improve the standards of global application development for open source platforms.

X open source users will be able to mix languages in the same document as well as use the open source license.

The X license is considered one of the most simple and open licenses in use today in the open source community. Under this license, anyone is free to use the code in any way they desire, without restriction. This is a commonly accepted license approved by the Open Standards Group, and has been used with the X Window System since its creation. The terms of the license can be viewed at: http://www.x.org/terms.htm.

The X Window System provides the only common windowing environment bridging the heterogeneous platforms in today's enterprise computing. The X Window System is one of the most successful open source, collaborative technologies developed to date and is the de facto standard graphical engine for the UNIX systems community.

The inherent independence of the X Window System from the operating system and hardware has made it widely available and deployed with over 20 million users worldwide. All major hardware vendors support the X Window System and many third parties provide technologies for integrating X Window System applications into the network computer or personal computer environments including DOS, Microsoft Windows, Windows 9x, Windows NT and Linux. Further, thousands of software developers provide X Window System applications, and with the emergence of Linux, the number of users is growing exponentially.

§ 9.01 Artistic License

Preamble

The intent of this document is to state the conditions under which a Package may be copied, such that the Copyright Holder maintains some semblance of artistic control over the development of the package, while giving the users of the package the right to use and distribute the Package in a more-or-less customary fashion, plus the right to make reasonable modifications.

Definitions:

"Package" refers to the collection of files distributed by the Copyright Holder, and derivatives of that collection of files created through textual modification.

"Standard Version" refers to such a Package if it has not been modified, or has been modified in accordance with the wishes of the Copyright Holder.

"Copyright Holder" is whoever is named in the copyright or copyrights for the package.

"You" is you, if you're thinking about copying or distributing this Package.

"Reasonable copying fee" is whatever you can justify on the basis of media cost, duplication charges, time of people involved, and so on. (You will not be required to justify it to the Copyright Holder, but only to the computing community at large as a market that must bear the fee.)

"Freely Available" means that no fee is charged for the item itself, though there may be fees involved in handling the item. It also means that recipients of the item may redistribute it under the same conditions they received it.

1. You may make and give away verbatim copies of the source form of the Standard Version of this Package without restriction, provided that you duplicate all of the original copyright notices and associated disclaimers.

2. You may apply bug fixes, portability fixes and other modifications derived from the Public Domain or from the Copyright Holder. A Package modified in such a way shall still be considered the Standard Version.

3. You may otherwise modify your copy of this Package in any way, provided that you insert a prominent notice in each changed file stating how and when you changed that file, and provided that you do at least ONE of the following:

a) place your modifications in the Public Domain or otherwise make them Freely Available, such as by posting said modifications to Usenet or an equivalent medium, or placing the modifications on a major archive site such as ftp.uu.net, or by allowing the Copyright Holder to include your modifications in the Standard Version of the Package.

b) use the modified Package only within your corporation or organization.

c) rename any non-standard executables so the names do not conflict with standard executables, which must also be provided, and provide a separate manual page for each non-standard executable that clearly documents how it differs from the Standard Version.

d) make other distribution arrangements with the Copyright Holder.

4. You may distribute the programs of this Package in object code or executable form, provided that you do at least ONE of the following:

a) distribute a Standard Version of the executables and library files, together with instructions (in the manual page or equivalent) on where to get the Standard Version.

b) accompany the distribution with the machine-readable source of the Package with your modifications.

c) accompany any non-standard executables with their corresponding Standard Version executables, giving the non-standard executables non-standard names, and clearly documenting the differences in manual pages (or equivalent), together with instructions on where to get the Standard Version.

d) make other distribution arrangements with the Copyright Holder.

5. You may charge a reasonable copying fee for any distribution of this Package. You may charge any fee you choose for support of this Package. You may not charge a fee for this Package itself. However, you may distribute this Package in aggregate with other (possibly commercial) programs as part of a larger (possibly commercial) software distribution provided that you do not advertise this Package as a product of your own.

6. The scripts and library files supplied as input to or produced as output from the programs of this Package do not automatically fall under the copyright of this Package, but belong to whomever generated them, and may be sold commercially, and may be aggregated with this Package.

7. C or perl subroutines supplied by you and linked into this Package shall not be considered part of this Package.

8. The name of the Copyright Holder may not be used to endorse or promote products derived from this software without specific prior written permission.

9. THIS PACKAGE IS PROVIDED "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTIBILITY AND FITNESS FOR A PARTICULAR PURPOSE.

§ 10.01 Mozilla Public License 1.0

Definitions….

The Initial Developer Grant.
The Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license, subject to third party intellectual property claims:

(a) to use, reproduce, modify, display, perform, sublicense and distribute the Original Code (or portions thereof) with or without Modifications, or as part of a Larger Work; and

(b) under patents now or hereafter owned or controlled by Initial Developer, to make, have made, use and sell ("Utilize") the Original Code (or portions thereof), but solely to the extent that any such patent is reasonably necessary to enable You to Utilize the Original Code (or portions thereof) and not to any greater extent that may be necessary to Utilize further Modifications or combinations.

2.2. Contributor Grant.
Each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license, subject to third party intellectual property claims:

(a) to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof) either on an unmodified basis, with other Modifications, as Covered Code or as part of a Larger Work; and

(b) under patents now or hereafter owned or controlled by Contributor, to Utilize the Contributor Version (or portions thereof), but solely to the extent that any such patent is reasonably necessary to enable You to Utilize the Contributor Version (or portions thereof), and not to any greater extent that may be necessary to Utilize further Modifications or combinations.

3. Distribution Obligations.

3.1. Application of License.
The Modifications which You create or to which You contribute are governed by the terms of this License, including without limitation Section 2.2. The Source Code version of Covered Code may be distributed only under the terms of this License or a future version of this License released under Section 6.1, and You must include a copy of this License with every copy of the Source Code You distribute. You may not offer or impose any terms on any Source Code version that alters or restricts the applicable version of this License or the recipients' rights hereunder. However, You may include an additional document offering the additional rights described in Section 3.5.

3.2. Availability of Source Code.
Any Modification which You create or to which You contribute must be made available in Source Code form under the terms of this License either on the same

media as an Executable version or via an accepted Electronic Distribution Mechanism to anyone to whom you made an Executable version available; and if made available via Electronic Distribution Mechanism, must remain available for at least twelve (12) months after the date it initially became available, or at least six (6) months after a subsequent version of that particular Modification has been made available to such recipients. You are responsible for ensuring that the Source Code version remains available even if the Electronic Distribution Mechanism is maintained by a third party.

3.3. Description of Modifications.
You must cause all Covered Code to which you contribute to contain a file documenting the changes You made to create that Covered Code and the date of any change. You must include a prominent statement that the Modification is derived, directly or indirectly, from Original Code provided by the Initial Developer and including the name of the Initial Developer in (a) the Source Code, and (b) in any notice in an Executable version or related documentation in which You describe the origin or ownership of the Covered Code.

3.4. Intellectual Property Matters

(a) Third Party Claims.
If You have knowledge that a party claims an intellectual property right in particular functionality or code (or its utilization under this License), you must include a text file with the source code distribution titled "LEGAL" which describes the claim and the party making the claim in sufficient detail that a recipient will know whom to contact. If you obtain such knowledge after You make Your Modification available as described in Section 3.2, You shall promptly modify the LEGAL file in all copies You make available thereafter and shall take other steps (such as notifying appropriate mailing lists or newsgroups) reasonably calculated to inform those who received the Covered Code that new knowledge has been obtained.

(b) Contributor APIs.
If Your Modification is an application programming interface and You own or control patents which are reasonably necessary to implement that API, you must also include this information in the LEGAL file.

3.5. Required Notices.
You must duplicate the notice in Exhibit A in each file of the Source Code, and this License in any documentation for the Source Code, where You describe recipients' rights relating to Covered Code. If You created one or more Modification(s), You may add your name as a Contributor to the notice described in

Exhibit A. If it is not possible to put such notice in a particular Source Code file due to its structure, then you must include such notice in a location (such as a relevant directory file) where a user would be likely to look for such a notice. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Code. However, You may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear than any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.6. Distribution of Executable Versions.
You may distribute Covered Code in Executable form only if the requirements of Section 3.1-3.5 have been met for that Covered Code, and if You include a notice stating that the Source Code version of the Covered Code is available under the terms of this License, including a description of how and where You have fulfilled the obligations of Section 3.2. The notice must be conspicuously included in any notice in an Executable version, related documentation or collateral in which You describe recipients' rights relating to the Covered Code. You may distribute the Executable version of Covered Code under a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable version does not attempt to limit or alter the recipient's rights in the Source Code version from the rights set forth in this License. If You distribute the Executable version under a different license You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or any Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

3.7. Larger Works.
You may create a Larger Work by combining Covered Code with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Code.

4. Inability to Comply Due to Statute or Regulation.

If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Code due to statute or regulation then You must: (a) comply with the terms of this License to the maximum extent possible;

and (b) describe the limitations and the code they affect. Such description must be included in the LEGAL file described in Section 3.4 and must be included with all distributions of the Source Code. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

5. Application of this License.

This License applies to code to which the Initial Developer has attached the notice in Exhibit A, and to related Covered Code.

6. Versions of the License.

6.1. New Versions.
Netscape Communications Corporation ("Netscape") may publish revised and/or new versions of the License from time to time. Each version will be given a distinguishing version number.

6.2. Effect of New Versions.
Once Covered Code has been published under a particular version of the License, You may always continue to use it under the terms of that version. You may also choose to use such Covered Code under the terms of any subsequent version of the License published by Netscape. No one other than Netscape has the right to modify the terms applicable to Covered Code created under this License.

6.3. Derivative Works.
If you create or use a modified version of this License (which you may only do in order to apply it to code which is not already Covered Code governed by this License), you must (a) rename Your license so that the phrases "Mozilla", "MOZILLAPL", "MOZPL", "Netscape", "NPL" or any confusingly similar phrase do not appear anywhere in your license and (b) otherwise make it clear that your version of the license contains terms which differ from the Mozilla Public License and Netscape Public License. (Filling in the name of the Initial Developer, Original Code or Contributor in the notice described in Exhibit A shall not of themselves be deemed to be modifications of this License.)

7. DISCLAIMER OF WARRANTY.

COVERED CODE IS PROVIDED UNDER THIS LICENSE ON AN "AS IS" BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION,

WARRANTIES THAT THE COVERED CODE IS FREE OF DEFECTS, MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGING. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED CODE IS WITH YOU. SHOULD ANY COVERED CODE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED CODE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

8. TERMINATION.

This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. All sublicenses to the Covered Code which are properly granted shall survive any termination of this License. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

9. LIMITATION OF LIABILITY.

….

ANNOTATION:

The MPL and the NPL strikes a balance between the BSD license and the GPL, and ensures that developers return their modifications to the open source community, and they assist Netscape in retaining specific rights that allow it to continue proprietary development of other packages and maintain existing contracts with third parties.

These licenses serve as a good model for any commercial company looking to develop an open source product, to convert a proprietary product to open source, or to make open source extensions or additions to proprietary products.

The MPL requires that licensees make modified or new files freely and publicly available in source code form, and allows developers to distribute their own files with "covered files" (any software or code that is covered by the MPL) under any licensing terms, provided that the developer has not modified the actual covered files.

The MPL satisfies the Open Source definition.

It contains thirteen sections with no preamble. Definitions are contained in section 1.

Section 2 grants the licensee a worldwide, royalty-free, non-exclusive license to use, reproduce, modify, display, perform, sublicense and distribute the licensed software.

Section 3 grants the licensee the right to distribute modifications of the software only if she (1) makes the source code of the modified version available under the same terms of the license and (2) gives proper notice of the license. These requirements are similar to those in the GPL.

The remaining provisions are substantially similar to the GPL, and include: a disclaimer of warranties in section 7; a statement that termination results if the licensee fails to comply with the terms of the license in section 8; a limitation of liability in section 9; and a complete integration clause in section 11.

§ 11.01 The Q Public License Version 1.0

Website: http://www.opensource.org

The intent of this license is to establish freedom to share and change the software regulated by this license under the open source model.

This license applies to any software containing a notice placed by the copyright holder saying that it may be distributed under the terms of the Q Public License version 1.0. Such software is herein referred to as the Software. This license covers modification and distribution of the Software, use of third-party application programs based on the Software, and development of free software which uses the Software.

Granted Rights

1. You are granted the non-exclusive rights set forth in this license provided you agree to and comply with any and all conditions in this license. Whole or partial distribution of the Software, or software items that link with the Software, in any form signifies acceptance of this license.

2. You may copy and distribute the Software in unmodified form provided that the entire package, including - but not restricted to - copyright, trademark notices and disclaimers, as released by the initial developer of the Software, is distributed.

3. You may make modifications to the Software and distribute your modifications, in a form that is separate from the Software, such as patches. The following restrictions apply to modifications:

a. Modifications must not alter or remove any copyright notices in the Software.

b. When modifications to the Software are released under this license, a non-exclusive royalty-free right is granted to the initial developer of the Software to distribute your modification in future versions of the Software provided such versions remain available under these terms in addition to any other license(s) of the initial developer.

4. You may distribute machine-executable forms of the Software or machine-executable forms of modified versions of the Software, provided that you meet these restrictions:

a. You must include this license document in the distribution.

b. You must ensure that all recipients of the machine-executable forms are also able to receive the complete machine-readable source code to the distributed Software, including all modifications, without any charge beyond the costs of data transfer, and place prominent notices in the distribution explaining this.

c. You must ensure that all modifications included in the machine-executable forms are available under the terms of this license.

5. You may use the original or modified versions of the Software to compile, link and run application programs legally developed by you or by others.

6. You may develop application programs, reusable components and other software items that link with the original or modified versions of the Software. These items, when distributed, are subject to the following requirements:

a. You must ensure that all recipients of machine-executable forms of these items are also able to receive and use the complete machine-readable source code to the items without any charge beyond the costs of data transfer.

b. You must explicitly license all recipients of your items to use and re-distribute original and modified versions of the items in both machine-executable and source code forms. The recipients must be able to do so without any charges whatsoever, and they must be able to re-distribute to anyone they choose.

c. If the items are not available to the general public, and the initial developer of the Software requests a copy of the items, then you must supply one.

….

ANNOTATIONS:

If a developer plans to distribute or publish any work that contains or is derived from code licensed under the Q license template, then

(1) the work must contain prominent notices stating it is a modified version and the dates of the changes made;

(2) the work must be licensed as a whole to all third parties under the terms of the license;

(3) copyright notices, instruction on how to view the license, and disclaimers of warranties must be displayed each time the work commences operation, if the work reads commands interactively when run;

(4) the work must be accompanied by the complete corresponding source code;

(5) any distributed written or printed material must include a written copy of the license or a notice that the work is covered by the license and instructions on how to view the license; and

(6) no further restrictions other than those enumerated in the license may be imposed on the recipient of the code.

§ 12.01 The Apple Public Source License Version 1.2 (the APSL)

Website: http://www.publicsource.apple.com/apsl/

Permitted Uses; Conditions & Restrictions. Subject to the terms and conditions of this License, Apple hereby grants You, effective on the date You accept this License and download the Original Code, a world-wide, royalty-free, non-exclusive license, to the extent of Apple's Applicable Patent Rights and copyrights covering the Original Code, to do the following:

2.1 You may use, reproduce, display, perform, modify and distribute Original Code, with or without Modifications, solely for Your internal research and development and/or Personal Use, provided that in each instance:

(a) You must retain and reproduce in all copies of Original Code the copyright and other proprietary notices and disclaimers of Apple as they appear in the Original Code, and keep intact all notices in the Original Code that refer to this License; and

(b) You must include a copy of this License with every copy of Source Code of Covered Code and documentation You distribute, and You may not offer or impose any terms on such Source Code that alter or restrict this License or the recipients' rights hereunder, except as permitted under Section 6.

2.2 You may use, reproduce, display, perform, modify and Deploy Covered Code, provided that in each instance:

(a) You must satisfy all the conditions of Section 2.1 with respect to the Source Code of the Covered Code;

(b) You must duplicate, to the extent it does not already exist, the notice in Exhibit A in each file of the Source Code of all Your Modifications, and cause the modified files to carry prominent notices stating that You changed the files and the date of any change;

(c) You must make Source Code of all Your Deployed Modifications publicly available under the terms of this License, including the license grants set forth in Section 3 below, for as long as you Deploy the Covered Code or twelve (12) months from the date of initial Deployment, whichever is longer. You should preferably distribute the Source Code of Your Deployed Modifications electronically (e.g. download from a web site); and

(d) if You Deploy Covered Code in object code, executable form only, You must include a prominent notice, in the code itself as well as in related documentation, stating that Source Code of the Covered Code is available under the terms of this License with information on how and where to obtain such Source Code.

2.3 You expressly acknowledge and agree that although Apple and each Contributor grants the licenses to their respective portions of the Covered Code set forth herein, no assurances are provided by Apple or any Contributor that the Covered Code does not infringe the patent or other intellectual property rights of any other entity. Apple and each Contributor disclaim any liability to You for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, You hereby assume sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow You to distribute the Covered Code, it is Your responsibility to acquire that license before distributing the Covered Code.

3. Your Grants. In consideration of, and as a condition to, the licenses granted to You under this License:

(a) You hereby grant to Apple and all third parties a non-exclusive, royalty-free license, under Your Applicable Patent Rights and other intellectual property rights (other than patent) owned or controlled by You, to use, reproduce, display, perform, modify, distribute and Deploy Your Modifications of the same scope and extent as Apple's licenses under Sections 2.1 and 2.2; and

(b) You hereby grant to Apple and its subsidiaries a non-exclusive, worldwide, royalty-free, perpetual and irrevocable license, under Your Applicable Patent Rights and other intellectual property rights (other than patent) owned or controlled by You, to use, reproduce, display, perform, modify or have modified (for Apple and/or its subsidiaries), sublicense and distribute Your Modifications, in any form, through multiple tiers of distribution.

4. Larger Works. You may create a Larger Work by combining Covered Code with other code not governed by the terms of this License and distribute the Larger Work as a single product. In each such instance, You must make sure the requirements of this License are fulfilled for the Covered Code or any portion thereof.

….

7. Versions of the License. Apple may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Once Original Code has been published under a particular version of this License, You may continue to use it under the terms of that version. You may also choose to use such Original Code under the terms of any subsequent version of this License published by Apple. No one other than Apple has the right to modify the terms applicable to Covered Code created under this License.

§ 13.01 The Python Copyright license

Website: http://www.python.org/doc/Copyright.html

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the names of Stichting Mathematisch Centrum or CWI or Corporation for National Research Initiatives or CNRI not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.

While CWI is the initial source for this software, a modified version is made available by the Corporation for National Research Initiatives (CNRI) at the Internet address ftp://ftp.python.org.

STICHTING MATHEMATISCH CENTRUM AND CNRI DISCLAIM ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL STICHTING MATHEMATISCH CENTRUM OR CNRI BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

§ 14.01 The Jabber Open Source License Version 1.0

Website: http://www.osdn.com/oslicenses/01/01/15/1914248.shtml

This Jabber Open Source License (the “License”) applies to Jabber Server and related software products as well as any updates or maintenance releases of that software(“Jabber Products”) that are distributed by Jabber.Com, Inc. (“Licensor”). Any Jabber Product licensed pursuant to this License is a “Licensed Product.” Licensed Product, in its entirety, is protected by U.S. copyright law. This License identifies the terms under which you may use, copy, distribute or modify Licensed Product.

Preamble

This Preamble is intended to describe, in plain English, the nature and scope of this License. However, this Preamble is not a part of this license. The legal effect of this License is dependent only upon the terms of the License and not this Preamble.

This License complies with the Open Source Definition and has been approved by Open Source Initiative. Software distributed under this License may be marked as “OSI Certified Open Source Software.”

This License provides that:

1. You may use, sell or give away the Licensed Product, alone or as a component of an aggregate software distribution containing programs from several different sources. No royalty or other fee is required.

2. Both Source Code and executable versions of the Licensed Product, including Modifications made by previous Contributors, are available for your use. (The terms “Licensed Product,” “Modifications,” “Contributors” and “Source Code” are defined in the License.)

3. You are allowed to make Modifications to the Licensed Product, and you can create Derivative Works from it. (The term “Derivative Works” is defined in the License.)

4. By accepting the Licensed Product under the provisions of this License, you agree that any Modifications you make to the Licensed Product and then distribute are governed by the provisions of this License. In particular, you must make the Source Code of your Modifications available to others.

5. You may use the Licensed Product for any purpose, but the Licensor is not providing you any warranty whatsoever, nor is the Licensor accepting any

liability in the event that the Licensed Product doesn't work properly or causes you any injury or damages.

6. If you sublicense the Licensed Product or Derivative Works, you may charge fees for warranty or support, or for accepting indemnity or liability obligations to your customers. You cannot charge for the Source Code.

7. If you assert any patent claims against the Licensor relating to the Licensed Product, or if you breach any terms of the License, your rights to the Licensed Product under this License automatically terminate.

You may use this License to distribute your own Derivative Works, in which case the provisions of this License will apply to your Derivative Works just as they do to the original Licensed Product.

Alternatively, you may distribute your Derivative Works under any other OSI-approved Open Source license, or under a proprietary license of your choice. If you use any license other than this License, however, you must continue to fulfill the requirements of this License (including the provisions relating to publishing the Source Code) for those portions of your Derivative Works that consist of the Licensed Product, including the files containing Modifications.

New versions of this License may be published from time to time. You may choose to continue to use the license terms in this version of the License or those from the new version. However, only the Licensor has the right to change the License terms as they apply to the Licensed Product.

This License relies on precise definitions for certain terms. Those terms are defined when they are first used, and the definitions are repeated for your convenience in a Glossary at the end of the License.

License Terms

1. Grant of License From Licensor. Licensor hereby grants you a world-wide, royalty-free, non-exclusive license, subject to third party intellectual property claims, to do the following:

a. Use, reproduce, modify, display, perform, sublicense and distribute Licensed Product or portions thereof (including Modifications as hereinafter defined), in both Source Code or as an executable program. "Source Code" means the preferred form for making modifications to the Licensed Product, including all modules contained therein, plus any associated interface definition files, scripts used to control compilation and installation of an executable program, or a list of differential comparisons against the Source Code of the Licensed Product.

b. Create Derivative Works (as that term is defined under U.S. copyright law) of Licensed Product by adding to or deleting from the substance or structure of said Licensed Product.

c. Under claims of patents now or hereafter owned or controlled by Licensor, to make, use, sell, offer for sale, have made, and/or otherwise dispose of Licensed Product or portions thereof, but solely to the extent that any such claim is necessary to enable you to make, use, sell, offer for sale, have made, and/or otherwise dispose of Licensed Product or portions thereof or Derivative Works thereof.

2. Grant of License to Modifications From Contributor. "Modifications" means any additions to or deletions from the substance or structure of (i) a file containing Licensed Product, or (ii) any new file that contains any part of Licensed Product. Hereinafter in this License, the term "Licensed Product" shall include all previous Modifications that you receive from any Contributor. By application of the provisions in Section 4(a) below, each person or entity who created or contributed to the creation of, and distributed, a Modification (a "Contributor") hereby grants you a world-wide, royalty-free, non-exclusive license, subject to third party intellectual property claims, to do the following:

a. Use, reproduce, modify, display, perform, sublicense and distribute any Modifications created by such Contributor or portions thereof, in both Source Code or as an executable program, either on an unmodified basis or as part of Derivative Works.

b. Under claims of patents now or hereafter owned or controlled by Contributor, to make, use, sell, offer for sale, have made, and/or otherwise dispose of Modifications or portions thereof, but solely to the extent that any such claim is necessary to enable you to make, use, sell, offer for sale, have made, and/or otherwise dispose of Modifications or portions thereof or Derivative Works thereof.

3. Exclusions From License Grant. Nothing in this License shall be deemed to grant any rights to trademarks, copyrights, patents, trade secrets or any other intellectualproperty of Licensor or any Contributor except as expressly stated herein. No patent license is granted separate from the Licensed Product, for code that you delete from the Licensed Product, or for combinations of the Licensed Product with other software or hardware. No right is granted to the trademarks of Licensor or any Contributor even if such marks are included in the Licensed Product. Nothing in this License shall be interpreted to prohibit

Licensor from licensing under different terms from this License any code that Licensor otherwise would have a right to license.

4. Your Obligations Regarding Distribution.

a. Application of This License to Your Modifications. As an express condition for your use of the Licensed Product, you hereby agree that any Modifications that you create or to which you contribute, and which you distribute, are governed by the terms of this License including, without limitation, Section 2. Any Modifications that you create or to which you contribute may be distributed only under the terms of this License or a future version of this License released under Section 7. You must include a copy of this License with every copy of the Modifications you distribute. You agree not to offer or impose any terms on any Source Code or executable version of the Licensed Product or Modifications that alter or restrict the applicable version of this License or the recipients' rights hereunder. However, you may include an additional document offering the additional rights described in Section 4(e).

b. Availability of Source Code. You must make available, under the terms of this License, the Source Code of the Licensed Product and any Modifications that you distribute, either on the same media as you distribute any executable or other form of the Licensed Product, or via a mechanism generally accepted in the software development community for the electronic transfer of data (an "Electronic Distribution Mechanism"). The Source Code for any version of Licensed Product or Modifications that you distribute must remain available for at least twelve (12) months after the date it initially became available, or at least six (6) months after a subsequent version of said Licensed Product or Modifications has been made available. You are responsible for ensuring that the Source Code version remains available even if the Electronic Distribution Mechanism is maintained by a third party.

c. Description of Modifications. You must cause any Modifications that you create or to which you contribute, and which you distribute, to contain a file documenting the additions, changes or deletions you made to create or contribute to those Modifications, and the dates of any such additions, changes or deletions. You must include a prominent statement that the Modifications are derived, directly or indirectly, from the Licensed Product and include the names of the Licensor and any Contributor to the Licensed Product in (i) the Source Code and (ii) in any notice displayed by a version of the Licensed Product you distribute or in related documentation in which you describe the origin or ownership of the Licensed Product. You may not modify or delete any preexisting copyright notices in the Licensed Product.

d. Intellectual Property Matters.

i. Third Party Claims. If you have knowledge that a license to a third party's intellectual property right is required to exercise the rights granted by this License, you must include a text file with the Source Code distribution titled "LEGAL" that describes the claim and the party making the claim in sufficient detail that a recipient will know whom to contact. If you obtain such knowledge after you make any Modifications available as described in Section 4(b), you shall promptly modify the LEGAL file in all copies you make available thereafter and shall take other steps (such as notifying appropriate mailing lists or newsgroups) reasonably calculated to inform those who received the Licensed Product from you that new knowledge has been obtained.

ii. Contributor APIs. If your Modifications include an application programming interface ("API") and you have knowledge of patent licenses that are reasonably necessary to implement that API, you must also include this information in the LEGAL file.

iii. Representations. You represent that, except as disclosed pursuant to 4(d)(i) above, you believe that any Modifications you distribute are your original creations and that you have sufficient rights to grant the rights conveyed by this License.

e. Required Notices. You must duplicate this License in any documentation you provide along with the Source Code of any Modifications you create or to which you contribute, and which you distribute, wherever you describe recipients' rights relating to Licensed Product. You must duplicate the notice contained in Exhibit A (the "Notice") in each file of the Source Code of any copy you distribute of the Licensed Product. If you created a Modification, you may add your name as a Contributor to the Notice. If it is not possible to put the Notice in a particular Source Code file due to its structure, then you must include such Notice in a location (such as a relevant directory file) where a user would be likely to look for such a notice. You may choose to offer, and charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Licensed Product. However, you may do so only on your own behalf, and not on behalf of the Licensor or any Contributor. You must make it clear that any such warranty, support, indemnity or liability obligation is offered by you alone, and you hereby agree to indemnify the Licensor and every Contributor for any liability incurred by the Licensor or such Contributor as a result of warranty, support, indemnity or liability terms you offer.

f. Distribution of Executable Versions. You may distribute Licensed Product as an executable program under a license of your choice that may contain terms different from this License provided (i) you have satisfied the requirements of Sections 4(a) through 4(e) for that distribution, (ii) you include a conspicuous notice in the executable version, related documentation and collateral materials stating that the Source Code version of the Licensed Product is available under the terms of this License, including a description of how and where you have fulfilled the obligations of Section 4(b), (iii) you retain all existing copyright notices in the Licensed Product, and (iv) you make it clear that any terms that differ from this License are offered by you alone, not by Licensor or any Contributor. You hereby agree to indemnify the Licensor and every Contributor for any liability incurred by Licensor or such Contributor as a result of any terms you offer.

g. Distribution of Derivative Works. You may create Derivative Works (e.g., combinations of some or all of the Licensed Product with other code) and distribute the Derivative Works as products under any other license you select, with the proviso that the requirements of this License are fulfilled for those portions of the Derivative Works that consist of the Licensed Product or any Modifications thereto.

....

7. Versions of This License.

a. New Versions. Licensor may publish from time to time revised and/or new versions of the License.

b. Effect of New Versions. Once Licensed Product has been published under a particular version of the License, you may always continue to use it under the terms of that version. You may also choose to use such Licensed Product under the terms of any subsequent version of the License published by Licensor. No one other than Licensor has the right to modify the terms applicable to Licensed Product created under this License.

c. Derivative Works of this License. If you create or use a modified version of this License, which you may do only in order to apply it to software that is not already a Licensed Product under this License, you must rename your license so that it is not confusingly similar to this License, and must make it clear that your license contains terms that differ from this License. In so naming your license, you may not use any trademark of Licensor or any Contributor.

....

License:

The contents of this file are subject to the Jabber Open Source License Version 1.0 (the "License"). You may not copy or use this file, in either source code or executable form, except in compliance with the License. You may obtain a copy of the License at http://www.jabber.com/license/ or at http://www.opensource.org/.

Software distributed under the License is distributed on an "AS IS" basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License for the specific language governing rights and limitations under the License.

….

§ 15.01 Amulet v3.0 - CMU

Website: http://www-2.cs.cmu.edu/afs/cs/project/amulet/amulet3/manual/overview.html

The Amulet research project in the School of Computer Science at Carnegie Mellon University is creating a comprehensive set of tools which make it significantly easier to create graphical, highly-interactive user interfaces. The lower levels of Amulet are called the 'Amulet Toolkit,' and these provide mechanisms that allow programmers to code user interfaces much more easily. Amulet stands for Automatic Manufacture of Usable and Learnable Editors and Toolkits. Although the website for Amulet indicates that the program has been "placed in the public domain," the license below is also provided.

TERMS

….

1.11 Changes since Version 2

There have been many changes in Amulet since version 2. In particular, all code that worked with V2 must be edited to compile with version 3. A complete and up-to-date list of changes and known problems is provided in the V3 release notes on the Amulet Web site. For complete information about Amulet's new features, you should look in the specific sections of this manual.

The previous version of the manual described all the changes from Version 1.2 to Version 2, but they have been eliminated to save space. If you are still using V1.2, you should probably check in the old manual first.

1.12 Formal, Legal Language

1. This License Agreement, effective as of April 1, 1996, is between: Carnegie Mellon University having a principal place of business at 5000 Forbes Avenue, Pittsburgh, PA 15213-3890 ("CMU"); and a company ("COMPANY").

2. CMU owns intellectual property rights to the computer software, electronic information and data, in all forms and versions, identified as Amulet, described in CMU Docket 96-050 ("Software"), and associated documentation ("Document"), collectively ("Program").

3. CMU grants to COMPANY, upon the terms and conditions set out below, a fully-paid, nonexclusive, world-wide, royalty-free, non-revocable, commercial license to use the Program, or any portion thereof, for any purpose, including, but not limited to, the right to grant sublicenses under CMU's patent and trade secret rights, and copyrights, including any renewals and extensions, the right to use, copy adapt, prepare derivative works of, distribute, sell, lease, or otherwise dispose of, reverse engineer, or disassemble the Software, or any portion thereof (including all subsequent editions, revisions, supplements, and versions thereof), and CMU acknowledges that COMPANY hereby grants no reciprocal rights.

4. COMPANY acknowledges that the Program is a research tool still in the development stage, that it is being supplied "as is," without any accompanying services or improvements from CMU.

5. CMU MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE, OR MERCHANTABILITY, EXCLUSIVITY OR RESULTS OBTAINED FROM SPONSOR'S USE OF ANY INTELLECTUAL PROPERTY DEVELOPED UNDER THIS AGREEMENT, NOR SHALL EITHER PARTY HERETO BE LIABLE TO THE OTHER FOR INDIRECT, SPECIAL, OR CONSEQUENTIAL DAMAGES SUCH AS LOSS OF PROFITS OR INABILITY TO USE SAID INTELLECTUAL PROPERTY OR ANY APPLICATIONS AND DERIVATION THEREOF. CMU DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT, OR THEFT OF TRADE SECRETS AND DOES NOT ASSUME ANY LIABILITY

HEREUNDER FOR ANY INFRINGEMENT OF ANY PATENT, TRADEMARK, OR COPYRIGHT ARISING FROM THE USE OF THE PROGRAM, INFORMATION, INTELLECTUAL PROPERTY, OR OTHER PROPERTY OR RIGHTS GRANTED OR PROVIDED TO IT HEREUNDER. THE USER AGREES THAT IT WILL NOT MAKE ANY WARRANTY ON BEHALF OF CMU, EXPRESSED OR IMPLIED, TO ANY PERSON CONCERNING THE APPLICATION OF OR THE RESULTS TO BE OBTAINED WITH THE PROGRAM UNDER THIS AGREEMENT.

6. COMPANY hereby agrees to defend, indemnify and hold harmless CMU, its trustees, officers, employees, attorneys and agents from all claims or demands made against them (and any related losses, expenses or costs) arising out of or relating to COMPANY's and/or its sublicensees' use of, disposition of, or conduct regarding the Licensed Technology and/or Licensed Product including but not limited to, any claims of product liability, personal injury (including, but not limited to, death) damage to property or violation of any laws or regulations including, but not limited to, claims of active or passive negligence.

7. COMPANY agrees that it will not make any warranty on behalf of CMU, express or implied, to any person concerning the application of or the results to be obtained with the Program.

8. Title to copyright to the Program and to Document shall at all times remain with CMU, and COMPANY agrees to preserve same. COMPANY agrees not to make any copies of the Document except for COMPANY's internal use, without prior written consent of CMU. COMPANY agrees to place the appropriate copyright notice on any such copies. Nothing herein shall be deemed to grant any license or rights in any other technology owned by CMU related to the Program.

9. COMPANY owns the rights to derivative works made by or on behalf of COMPANY. Nothing herein shall be deemed to grant to CMU, or any other party, any license or any rights in any technology owned by COMPANY whether or not related to the Program, or any license or any rights to COMPANY's products whether or not incorporating any portion of the Software, or any portion of any derivative works thereof, and CMU acknowledges that CMU has no rights to same.

10. This Agreement shall be construed, interpreted and applied in accordance with the laws of the Commonwealth of Pennsylvania.

11. Nothing in this Agreement shall be construed as conferring rights to use in advertising, publicity or otherwise any trademark or the name of "CMU".

….

§ 16.01 Red Hat eCos Public License v1.1

Website: http://sources.redhat.com/ecos/license.html

Definitions

….

SOURCE CODE LICENSE

2.1. The Initial Developer Grant.

The Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license, subject to third party intellectual property claims:

(a) to use, reproduce, modify, display, perform, sublicense and distribute the Original Code (or portions thereof) with or without Modifications, or as part of a Larger Work; and

(b) under patents now or hereafter owned or controlled by Initial Developer, to make, have made, use and sell ("Utilize") the Original Code (or portions thereof), but solely to the extent that any such patent is reasonably necessary to enable You to Utilize the Original Code (or portions thereof) and not to any greater extent that may be necessary to Utilize further Modifications or combinations.

2.2. Contributor Grant.

Each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license, subject to third party intellectual property claims:

(a) to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof) either on an unmodified basis, with other Modifications, as Covered Code or as part of a Larger Work; and

(b) under patents now or hereafter owned or controlled by Contributor, to Utilize the Contributor Version (or portions thereof), but solely to the extent that any such patent is reasonably necessary to enable You to Utilize the Contributor Version (or portions thereof), and not to any greater extent that may be necessary to Utilize further Modifications or combinations.

3. DISTRIBUTION OBLIGATIONS

3.1. Application of License.

The Modifications which You create or to which You contribute are governed by the terms of this License, including without limitation Section 2.2. The Source Code version of Covered Code may be distributed only under the terms of this License or a future version of this License released under Section 6.1, and You must include a copy of this License with every copy of the Source Code You distribute. You may not offer or impose any terms on any Source Code version that alters or restricts the applicable version of this License or the recipients' rights hereunder. However, You may include an additional document offering the additional rights described in Section 3.5.

3.2. Availability of Source Code.

Any Modification which You create or to which You contribute must be made available in Source Code form under the terms of this License via an accepted Electronic Distribution Mechanism to anyone to whom you made an Executable version available and to the Initial Developer; and if made available via Electronic Distribution Mechanism, must remain available for at least twelve (12) months after the date it initially became available, or at least six (6) months after a subsequent version of that particular Modification has been made available to such recipients. You are responsible for ensuring that the Source Code version remains available even if the Electronic Distribution Mechanism is maintained by a third party. You are responsible for notifying the Initial Developer of the Modification and the location of the Source if a contact means is provided. Red Hat will be acting as maintainer of the Source and may provide an Electronic Distribution mechanism for the Modification to be made available. You can contact Red Hat to make the Modification available and to notify the Initial Developer. (http://sourceware.cygnus.com/ecos/)

3.3. Description of Modifications.

You must cause all Covered Code to which you contribute to contain a file documenting the changes You made to create that Covered Code and the date of any change. You must include a prominent statement that the Modification is derived, directly or indirectly, from Original Code provided by the Initial Developer and including the name of the Initial Developer in (a) the Source Code, and (b) in any notice in an Executable version or related documentation in which You describe the origin or ownership of the Covered Code.

3.4. Intellectual Property Matters

(a) Third Party Claims.
If You have knowledge that a party claims an intellectual property right in particular functionality or code (or its utilization under this License), you must include a text file with the source code distribution titled “LEGAL” which describes the claim and the party making the claim in sufficient detail that a recipient will know whom to contact. If you obtain such knowledge after You make Your Modification available as described in Section 3.2, You shall promptly modify the LEGAL file in all copies You make available thereafter and shall take other steps (such as notifying appropriate mailing lists or newsgroups) reasonably calculated to inform those who received the Covered Code that new knowledge has been obtained.

(b) Contributor APIs.
If Your Modification is an application programming interface and You own or control patents which are reasonably necessary to implement that API, you must also include this information in the LEGAL file.

3.5. Required Notices.

You must duplicate the notice in Exhibit A in each file of the Source Code, and this License in any documentation for the Source Code, where You describe recipients’ rights relating to Covered Code. If You created one or more Modification(s), You may add your name as a Contributor to the Source Code. If it is not possible to put such notice in a particular Source Code file due to its structure, then you must include such notice in a location (such as a relevant directory file) where a user would be likely to look for such a notice. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Code.

However, You may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

….

6. VERSIONS OF THE LICENSE

6.1. New Versions.

Red Hat may publish revised and/or new versions of the License from time to time. Each version will be given a distinguishing version number.

6.2. Effect of New Versions.

Once Covered Code has been published under a particular version of the License, You may always continue to use it under the terms of that version. You may also choose to use such Covered Code under the terms of any subsequent version of the License published by Red Hat. No one other than Red Hat has the right to modify the terms applicable to Covered Code beyond what is granted under this and subsequent Licenses.

6.3. Derivative Works.

If you create or use a modified version of this License (which you may only do in order to apply it to code which is not already Covered Code governed by this License), you must (a) rename Your license so that the phrases “ECOS”, “eCos”, “Red Hat”, “RHEPL” or any confusingly similar phrase do not appear anywhere in your license and (b) otherwise make it clear that your version of the license contains terms which differ from the Red Hat eCos Public License. (Filling in the name of the Initial Developer, Original Code or Contributor in the notice described in Exhibit A shall not of themselves be deemed to be modifications of this License.)

….

§ 17.01 Eiffel Forum License, version 1

Website: http://www.eiffel-forum.org/license/#efl

Permission is hereby granted to use, copy, modify and/or distribute this package, provided that:

copyright notices are retained unchanged

any distribution of this package, whether modified or not, includes this file

Permission is hereby also granted to distribute binary programs which depend on this package, provided that:

if the binary program depends on a modified version of this package, you must publicly release the modified version of this package

THIS PACKAGE IS PROVIDED "AS IS" AND WITHOUT WARRANTY. ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHORS BE LIABLE TO ANY PARTY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES ARISING IN ANY WAY OUT OF THE USE OF THIS PACKAGE.

§ 18.01 License Agreement for Mozart, an implementation of Oz 3

Website: http://www.mozart-oz.org/LICENSE.html

This software and its documentation are copyrighted by the German Research Center for Artificial Intelligence (DFKI), the Swedish Institute of Computer Science (SICS), and other parties. The following terms apply to all files associated with the software unless explicitly disclaimed in individual files (see the file LICENSE-others for a list of these exceptions).

The authors hereby grant permission to use, copy, modify, distribute, and license this software and its documentation for any purpose, provided that existing copyright notices are retained in all copies and that this notice is included verbatim in any distributions. No written agreement, license, or royalty fee is required for any of the authorized uses. Modifications to this software may be copyrighted by their authors and need not follow the licensing terms described here, provided that the new terms are clearly indicated on the first page of each file where they apply.

IN NO EVENT SHALL THE AUTHORS OR DISTRIBUTORS BE LIABLE TO ANY PARTY FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OF THIS SOFTWARE, ITS DOCUMENTATION, OR ANY DERIVATIVES THEREOF, EVEN IF THE AUTHORS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

THE AUTHORS AND DISTRIBUTORS SPECIFICALLY DISCLAIM ANY WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT. THIS SOFTWARE AND ITS DOCUMENTATION ARE PROVIDED ON AN “AS IS” BASIS, AND THE AUTHORS AND DISTRIBUTORS HAVE NO OBLIGATION TO PROVIDE MAINTENANCE, SUPPORT, UPDATES, ENHANCEMENTS, OR MODIFICATIONS.

§ 19.01 LUCENT TECHNOLOGIES INC. PLAN 9 OPEN SOURCE LICENSE AGREEMENT

Website: http://plan9.bell-labs.com/plan9dist/license.html

IF YOU DO NOT AGREE TO ALL OF THE TERMS OF THIS AGREEMENT, CLICK ON THE “DO NOT ACCEPT” BUTTON AND THE INSTALLATION/DOWNLOAD PROCESS WILL NOT CONTINUE.

1. DEFINITIONS

“Agreement” means this Lucent Technologies Inc. Plan 9 Open Source License Agreement (including Exhibits).

“Contributor(s)” means any individual or legal entity that creates or contributes to a Modification of the Original Software.

“Licensee” means an individual or a legal entity entering into and exercising rights under this Agreement. For the purposes hereunder, Licensee includes any entity that controls, is controlled by, or is under common control with Licensee. For purposes of this definition, “control” means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise; or (ii) ownership of fifty percent (50%) or more of the controlling shares or beneficial ownership of such entity. Licensee is also referred to herein as "You" with "Your" as the possessive.

"Licensed Software" means the Original Software, Modifications, or any combination of the Original Software and Modifications.

"Lucent" means Lucent Technologies Inc., a Delaware corporation having an office at 600 Mountain Ave., Murray Hill, NJ 07974, its related companies and/or affiliates.

"Modification(s)" means any addition, deletion, change, or improvement to the Original Software or prior Modifications thereto. Modifications do not include additions to the Original Software or prior Modifications which (i) are separate modules of software which may be distributed in conjunction with Licensed Software; or (ii) are not derivative works of the Licensed Software itself.

"Object Code" means machine executable software code.

"Original Contributor" means Lucent and its Licensors, collectively.

"Original Software" means the Plan 9 Software, in both Source Code form and Object Code form, and any associated documentation, as furnished under this Agreement.

"Plan 9 Software" means a network operating system designed for research into distributed services, applications and software development.

"Plan 9 Trademark" means the trademark PLAN 9 (for which Lucent has acquired common law rights and for which Lucent owns U.S. Trademark Registration Number 2,065,577).

"Recipient" means any individual or legal entity receiving the Licensed Software under this Agreement, including all Contributors, or receiving the Licensed Software under another license agreement as authorized herein.

"Source Code" means human readable software code.

2.0 GRANT OF RIGHTS

2.1 Subject to the terms of this Agreement and to third party intellectual property claims, Lucent grants to Licensee, a royalty-free, nonexclusive, nontransferable, worldwide license to use, reproduce, modify, execute, display, perform, distribute and sublicense, the Original Software (with or without Modifications) in Source Code form and/or Object Code form for commercial and/or non-commercial purposes. This grant includes a nonexclusive and nontransferable license under any patents which Lucent has a right to license and which, but for this license, are unavoidably and necessarily infringed by the execution of the inherent functionality of the Original Software in the form furnished under this Agreement. Nothing in this Agreement shall be construed as conferring in any way (by implication, estoppel or otherwise) any license or right under any existing or future patent claim which is directed to a combination of the functionality of the Original Software with the functionality of any other software programs, or a combination of hardware systems other than the combination of the Original Software and the hardware or firmware into which the Original Software is loaded. Distribution of Licensed Software to third parties pursuant to this grant shall be subject to the same terms and conditions as set forth in this Agreement, and may, at Your option, include a reasonable charge for the cost of any media. You may also, at Your option, charge for any other software, product or service that includes or incorporates the Original Software as a part thereof.

2.2 No right is granted to Licensee to create derivative works of or to redistribute (other than with the Original Software or a derivative thereof) the screen imprinter fonts identified in subdirectory /lib/font/bit/lucida and printer fonts (Lucida Sans Unicode, Lucida Sans Italic, Lucida Sans Demibold, Lucida Typewriter, Lucida Sans Typewriter83), identified in subdirectory /sys/lib/postscript/font.

2.3 Exhibit A contains additional terms and conditions relating to the printer fonts identified in subdirectory /sys/lib/ghostscript/font. In the case of any conflict between the provisions of the body of this Agreement and Exhibit A regarding such printer fonts, the provisions of Exhibit A shall control.

2.4 The Original Software licensed herein contains material copyrights by the Original Contributor, including but not limited to Lucent, B&H Inc., and Y&Y Inc. No rights are granted with respect to Original Software except as expressly provided herein.

2.5 Lucent grants to Licensee a nonexclusive, royalty free, worldwide license to use the Plan 9 Trademark solely in connection with the Plan 9 operating system source code (or object code) and documentation. Such use by Licensee of the Plan 9 Trademark shall be in accordance with the following quality standards and controls:

Any use of the Plan 9 Trademark must be made under the terms of this Agreement;

The Plan 9 Trademark may not be combined with any other mark or logo to form a composite mark or logo or suggest that the Parties are part of one company.

Upon Lucent's written request and at Licensee's expense, Licensee will provide Lucent with a representative sample of Licensee's promotional materials bearing the Plan 9 Trademark. If, for any reason, Lucent determines that the quality standards or controls applied by Licensee to the Plan 9 system source code and documentation fall below those that are consistent with Lucent's standards, upon written notice of the deficiency to Licensee, Lucent may, at its sole option and discretion, terminate Licensee's right to use the Plan 9 Trademark upon written notice to Licensee.

Licensee acknowledges that Lucent is the owner of the Plan 9 Trademark and all goodwill attached thereto. This Agreement does not give Licensee any interest in the Plan 9 Trademark except the right to use the mark in accordance with the provisions of this Agreement. Licensee agrees not to attempt to register the Plan 9 Trademark nor to adopt, attempt to register or register anywhere in the world a mark the same as or confusingly similar to the Plan 9 Trademark.

3.0 DISTRIBUTION OBLIGATIONS

3.1 Modifications which You create or to which You contribute are governed by the terms of this Agreement and must be made available under the terms this Agreement in at least the same form as the Source Code version of Original Software furnished hereunder. Any distribution by You of the Source Code version of Licensed Software must be made under the terms of this Agreement or any future version of this Agreement under Section 11.0, and You must include a copy of this Agreement with each and every copy of such Source Code version of Licensed Software which You distribute. You may not offer or impose any terms on any such Source Code version of Licensed Software that alters or restricts the terms of the applicable version of this Agreement or the Recipients' rights and obligations hereunder.

3.2 You must cause all Licensed Software to which You contribute, i.e. Your Modifications, to contain a clear identification, e.g., a separate file, documenting the changes made by You and identifying You as the Contributor that reasonably allows subsequent Recipients to identify the originator of the Modification. To the extent You create at least one Modification, You may add Your name as a Contributor to the requisite notice described in Section 3.3.

3.3 With respect to Your distribution of Licensed Software (or any portion thereof), You must include the following information in a conspicuous location governing such distribution (e.g., a separate file) and on all copies of any Source Code version of Licensed Software You distribute:

"The contents herein includes software initially developed by Lucent Technologies Inc. and others, and is subject to the terms of the Lucent Technologies Inc. Plan 9 Open Source License Agreement. A copy of the Plan 9 open Source License Agreement is available at: http://plan9.bell-labs.com/plan9dist/download.html or by contacting Lucent Technologies at http: //www.lucent.com http://www.lucent.com/research/

All software distributed under such Agreement is distributed on an "AS IS" basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the Lucent Technologies Inc. Plan 9 Open Source License Agreement for the specific language governing all rights, obligations and limitations under such Agreement.

Portions of the software developed by Lucent Technologies Inc. and others are Copyright © 2000. All rights reserved.

Contributor(s):______________________________"

3.4 You may distribute Licensed Software in Object Code form using this Agreement, or under a license of Your choice provided that You are in compliance with this Agreement and Your license: (a) complies with the terms and conditions of this Agreement; (b) does not limit or alter the Recipient's rights and obligations in the Source Code version of the Licensed Software set forth in this Agreement; (c) states that the Source Code version of the Licensed Software is available from You, and describes how to it may be obtained by Recipient; (d) effectively disclaims on behalf of Original Contributor and all Contributors all warranties and conditions, express or implied, including warranties or conditions of title or non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose; (e) effectively excludes on behalf of Original Contributor and all Contributors all liability for damages, including direct, indirect, special, incidental, and consequential damages; and (f) clearly states that any terms which differ from this Agreement are offered by You alone, not by Original Contributor or any other Contributor. You hereby agree to indemnify Original Contributor or any other Contributor for any liability incurred by Original Contributor or any other Contributor as result of any such differing terms You offer in Your license.

3.5 You may not use the names "Lucent Technologies", "Bell Labs" or any other name associated with Lucent or any Lucent trademark for any purposes other than as specifically provided in this Agreement.

3.6 You must include all of the original copyright, labels or other notices on the Licensed Software on any copies of the Licensed Software which You make; and include with the distribution of any Modifications You create a copy (or an offer to provide such a copy at no charge) of the Licensed Software, on the same terms as set forth in this Agreement.

3.7 While this Agreement contemplates the commercial use and distribution of Licensed Software, commercial distributors of software may, for a variety of reasons, accept certain responsibilities with respect to customers, licensees, business partners and the like. As such, if You or any Contributor include Licensed Software in a commercial offering ("Commercial Contributor"), such Commercial Contributor agrees to defend and indemnify Original Contributor and all other Contributors (collectively "Indemnified Contributors") against any liability, losses, damages and costs arising from claims, lawsuits and other legal actions brought by any third party against the Indemnified Contributors to the extent caused by the acts or omissions of such Commercial Contributor in connection with its use or distribution of Licensed Software in a commercial offering of any kind.

4.0 MODIFICATIONS

You agree to provide the Original Contributor, at its request, with a copy of the complete Source Code version, Object Code version and related documentation for Modifications created or contributed to by You if distributed in any form, e.g., binary or source. Original Contributor and/or other Contributors shall have unrestricted, nonexclusive, worldwide, perpetual, royalty-free rights, to use, reproduce, modify, display, perform, sublicense and distribute such Modifications, and to grant third parties the right to do so, including without limitation as a part of or with the Licensed Software; and Original Contributor and/or other Contributors shall have the right to license or to otherwise transfer to third parties such Modifications without notice, obligation or recourse to You. You grant to Original Contributor, Contributors and their respective licensees all rights and licenses (including patents) as are necessary to incorporate the Modifications created or contributed and so distributed by You into the Licensed Software and to use, distribute or otherwise exploit such Licensed Software without payment or accounting to You.

TITLE

Title, ownership rights, and intellectual property rights in the Original Software and the Plan 9 Trademark shall remain in the Original Contributor. Original Contributor and/or the other Contributors reserve all rights not expressly granted to You, and no other licenses are granted or implied. The Licensed Software is protected by copyright laws and treaties.

6.0 TERMINATION

The licenses and rights granted under this Agreement shall terminate automatically if (i) You fail to comply with all of the terms and conditions herein; or (ii) You initiate or participate in any intellectual property action against Original Contributor.

The rights and obligations of the parties hereto which by their nature would continue beyond termination of this Agreement shall survive and continue after any such termination of this Agreement.

Upon termination for any reason, You must destroy all copies of the Licensed Software in Your possession. All sublicenses of Licensed Software which were validly granted by You to third parties under this Agreement shall survive such termination.

....

12.0 MISCELLANEOUS

This Agreement sets forth the entire agreement and understanding between the parties as to the subject matter hereof and merges all prior discussions between them. This Agreement shall be governed by the laws of the State of New York, USA, excluding its conflict of law provisions. The application of the United Nations Convention of Contracts for the International Sale of Goods is expressly excluded. YOUR DOWNLOAD, INSTALLATION AND USE, MODIFICATION OR DISTRIBUTION OF THE LICENSED SOFTWARE IS EXPRESSLY MADE CONDITIONAL ON YOUR ASSENT TO THE TERMS SET FORTH HEREIN. You further agree and acknowledge that by clicking on the "ACCEPT" button below, You shall have manifested acceptance to enter into this Agreement and shall be deemed to have manually signed and executed this Agreement making this an enforceable Agreement between the parties. If any provision of this Agreement is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable.

ANNOTATIONS:

The Plan 9 license is intended to cover a complex set of programs and tools constituting a Unix-like operating system (although the system is a windowing system). Since the license was approved by the Open Source Initiative this summer after revisions to the license were implemented, potential users of this template should also consult the Plan 9 website to review the revisions at http://plan9.bell-labs.com/plan9dist/download.html.

The license is particularly careful at pointing out the origins of the original source code, and that emphasis is likely to be appealing to developers who would like to draft a license that covers an open source project that began in-house as a project for a different purpose. Section 3 of the license, for example, notes that: "the contents herein includes software initially developed by Lucent Technologies Inc…"

The Plan 9 OS began in the late 1980's as an attempt to cheaply build a UNIX version operating system that was centrally administered in other words to build or patch a UNIX out of a number of other systems since the prevailing perception by some was that the problems with the standard UNIX were too deep to fix. The Plan 9 OS has its own compilers, languages, libraries, and a windowing user interface system.

It is important to note that the Plan 9 license is intended to be used with an initial interface on a website where the prospective licensee expresses agreement to be bound by the license by completing an online form requiring the name and address of the licensee as well as mouse-click answers to a few check-off questions.

If courts find Plan 9's use of the website agreement, an effective contract formation tool for the actual Plan 9 license, then the click-wrap aspect of this license is likely to be an essential part of the license that should also be read by those considering on using the license as a template.

§ 20.01 The LaTeX Project Public License (the LPPL)

Website: http://www.latex-project.org/lppl.html

Copyright 1999 LaTeX3 Project

PREAMBLE

The LaTeX Project Public License (LPPL) is the license under which the base LaTeX distribution is distributed.

You may use this license for any program that you have written and wish to distribute. This license may be particularly suitable if your program is TeX-related (such as a LaTeX package), but you may use it even if your program is unrelated to TeX. The section 'WHETHER AND HOW TO DISTRIBUTE PROGRAMS UNDER THIS LICENSE', below, gives instructions, examples, and recommendations for authors who are considering distributing their programs under this license.

In this license document, 'The Program' refers to any program distributed under this license.

This license gives conditions under which The Program may be distributed and conditions under which modified versions of The Program may be distributed. Individual files of The Program may bear supplementary and/or superseding conditions on modification of themselves and on the distribution of modified versions of themselves, but *no* file of The Program may bear supplementary or superseding conditions on the distribution of an unmodified copy of the file. A distributor wishing to distribute a complete, unmodified copy of The Program therefore needs to check the conditions only in this license and nowhere else.

Activities other than distribution and/or modification of The Program are not covered by this license; they are outside its scope. In particular, the act of running The Program is not restricted.

We, the LaTeX3 Project, believe that the conditions below give you the freedom to make and distribute modified versions of The Program that conform with whatever technical specifications you wish while maintaining the availability, integrity, and reliability of The Program. If you do not see how to achieve your goal while meeting these conditions, then read the document 'cfgguide.tex' in the base LaTeX distribution for suggestions.

CONDITIONS ON DISTRIBUTION AND MODIFICATION

You may distribute a complete, unmodified copy of The Program. Distribution of only part of The Program is not allowed.

You may not modify in any way a file of The Program that bears a legal notice forbidding modification of that file.

You may distribute a modified file of The Program if, and only if, the following eight conditions are met:

1. You must meet any additional conditions borne by the file on the distribution of a modified version of the file as described below in the subsection 'Additional Conditions on Individual Files of The Program'.

2. If the file is a LaTeX software file, then you must meet any applicable additional conditions on the distribution of a modified version of the file that are described below in the subsection 'Additional Conditions on LaTeX Software Files'.

3. You must not distribute the modified file with the filename of the original file.

4. In the modified file, you must acknowledge the authorship and name of the original file, and the name (if any) of the program which contains it.

5. You must change any identification string in the file to indicate clearly that the modified file is not part of The Program.

6. You must change any addresses in the modified file for the reporting of errors in the file or in The Program generally to ensure that reports for files no longer

maintained by the original maintainers will be directed to the maintainers of the modified files.

7. You must distribute the modified file under a license that forbids distribution both of the modified file and of any files derived from the modified file with the filename of the original file.

8. You must do either (A) or (B):

(A)

distribute a copy of The Program (that is, a complete, unmodified copy of The Program) together with the modified file; if your distribution of the modified file is made by offering access to copy the modified file from a designated place, then offering equivalent access to copy The Program from the same place meets this condition, even though third parties are not compelled to copy The Program along with the modified file;

(B)

provide to those who receive the modified file information that is sufficient for them to obtain a copy of The Program; for example, you may provide a Uniform Resource Locator (URL) for a site that you expect will provide them with a copy of The Program free of charge (either the version from which your modification is derived, or perhaps a later version).

Note that in the above, 'distribution' of a file means making the file available to others by any means. This includes, for instance, installing the file on any machine in such a way that the file is accessible by users other than yourself. 'Modification' of a file means any procedure that produces a derivative file under any applicable law - that is, a file containing the original file or a significant portion of it, either verbatim or with modifications and/or translated into another language.

Changing the name of a file (other than as necessitated by the file conventions of the target file systems) is considered to be a modification of the file.

The distribution conditions in this license do not have to be applied to files that have been modified in accordance with the above conditions. Note, however, that Condition 7. does apply to any such modified file.

The conditions above are not intended to prohibit, and hence do not apply to, the updating, by any method, of a file so that it becomes identical to the latest version of that file of The Program.

A Recommendation on Modification Without Distribution

It is wise never to modify a file of The Program, even for your own personal use, without also meeting the above eight conditions for distributing the modified file. While you might intend that such modified files will never be distributed, often this will happen by accident - you may forget that you have modified the file; or it may not occur to you when allowing others to access the modified file that you are thus distributing it and violating the conditions of this license. It is usually in your best interest to keep your copy of The Program identical with the public one. Many programs provide ways to control the behavior of that program without altering its licensed files.

Additional Conditions on Individual Files of The Program

An individual file of The Program may bear additional conditions that supplement and/or supersede the conditions in this license if, and only if, such additional conditions exclusively concern modification of the file or distribution of a modified version of the file. The conditions on individual files of The Program therefore may differ only with respect to the kind and extent of modification of those files that is allowed, and with respect to the distribution of modified versions of those files.

Additional Conditions on LaTeX Software Files

If a file of The Program is intended to be used with LaTeX (that is, if it is a LaTeX software file), then the following additional conditions, which supplement and/or supersede the conditions above, apply to the file according to its filename extension:

You may not modify any file with filename extension '.ins' since these are installation files containing the legal notices that are placed in the files they generate.

You may distribute modified versions of files with filename extension '.fd' (LaTeX font definition files) under the standard conditions of the LPPL as described above. You may also distribute such modified LaTeX font definition files with their original names provided that:

(1)

the only changes to the original files either enable use of available fonts or prevent attempts to access unavailable fonts;

(2)

you also distribute the original, unmodified files (TeX input paths can be used to control which set of LaTeX font definition files is actually used by TeX).

You may distribute modified versions of files with filename extension '.cfg' (configuration files) with their original names. The Program may (and usually will) specify the range of commands that are allowed in a particular configuration file.

Because of portability and exchangeability issues in LaTeX software, The LaTeX3 Project deprecates the distribution of modified versions of components of LaTeX or of generally available contributed code for them, but such distribution can meet the conditions of this license.

NO WARRANTY

There is no warranty for The Program. Except when otherwise stated in writing, The Copyright Holder provides 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.

In no event unless agreed to in writing will The Copyright Holder, or any author named in the files of The Program, or any other party who may distribute and/or modify The Program as permitted above, be liable to you for damages, including any general, special, incidental or consequential damages arising out of any use of The Program or out of inability to use The Program (including, but not limited to, loss of data, data being rendered inaccurate, or losses sustained by anyone as a result of any failure of The Program to operate with any other programs), even if The Copyright Holder or said author or said other party has been advised of the possibility of such damages.

§ 20.02 The LaTeX Project Public License – Applications

WHETHER AND HOW TO DISTRIBUTE PROGRAMS UNDER THIS LICENSE

This section contains important instructions, examples, and recommendations for authors who are considering distributing their programs under this license. These authors are addressed as 'you' in this section.

Choosing This License or Another License

If for any part of your program you want or need to use *distribution* conditions that differ from those in this license, then do not refer to this license anywhere in your program but instead distribute your program under a different license. You may use the text of this license as a model for your own license, but your license should not refer to the LPPL or otherwise give the impression that your program is distributed under the LPPL.

The document 'modguide.tex' in the base LaTeX distribution explains the motivation behind the conditions of this license. It explains, for example, why distributing LaTeX under the GNU General Public License (GPL) was considered inappropriate. Even if your program is unrelated to LaTeX, the discussion in 'modguide.tex' may still be relevant, and authors intending to distribute their programs under any license are encouraged to read it.

How to Use This License

To use this license, place in each of the files of your program both an explicit copyright notice including your name and the year and also a statement that the distribution and/or modification of the file is constrained by the conditions in this license.

Here is an example of such a notice and statement:

> This program may be distributed and/or modified under the
>
> conditions of the LaTeX Project Public License, either version 1.2
>
> of this license or (at your option) any later version.

The latest version of this license is in

http://www.latex-project.org/lppl.txt

and version 1.2 or later is part of all distributions of LaTeX

version 1999/12/01 or later.

This program consists of the files pig.dtx and pig.ins

Given such a notice and statement in a file, the conditions given in this license document would apply, with 'The Program' referring to the two files 'pig.dtx' and 'pig.ins', and 'The Copyright Holder' referring to the person 'M. Y. Name'.

Important Recommendations

Defining What Constitutes The Program

The LPPL requires that distributions of The Program contain all the files of The Program. It is therefore important that you provide a way for the licensee to determine which files constitute The Program. This could, for example, be achieved by explicitly listing all the files of The Program near the copyright notice of each file or by using a line like

This program consists of all files listed in manifest.txt.

in that place. In the absence of an unequivocal list it might be impossible for the licensee to determine what is considered by you to comprise The Program.

Noting Exceptional Files

If The Program contains any files bearing additional conditions on modification, or on distribution of modified versions, of those files (other than those listed in 'Additional Conditions on LaTeX Software Files'), then it is recommended that The Program contain a prominent file that defines the exceptional conditions, and either lists the exceptional files or defines one or more categories of exceptional files.

Files containing the text of a license (such as this file) are often examples of files bearing more restrictive conditions on modification. LaTeX configuration files (with filename extension '.cfg') are examples of files bearing less restrictive conditions on the distribution of a modified version of the file. The additional conditions on LaTeX software given above are examples of declaring a category of files bearing exceptional additional conditions.

§ 21.01 The Open Software License (the OPL)

Website: http://www.opensource.org/licenses/osl.php

Copyright 2002 Lawrence E. Rosen

v. 1.1

This Open Software License (the "License") applies to any original work of authorship (the "Original Work") whose owner (the "Licensor") has placed the following notice immediately following the copyright notice for the Original Work:

Licensed under the Open Software License version 1.1

1) Grant of Copyright License. Licensor hereby grants You a world-wide, royalty-free, non-exclusive, perpetual, non-sublicenseable license to do the following:

a) to reproduce the Original Work in copies;

b) to prepare derivative works ("Derivative Works") based upon the Original Work;

c) to distribute copies of the Original Work and Derivative Works to the public, with the proviso that copies of Original Work or Derivative Works that You distribute shall be licensed under the Open Software License;

d) to perform the Original Work publicly; and

e) to display the Original Work publicly.

2) Grant of Patent License. Licensor hereby grants You a world-wide, royalty-free, non-exclusive, perpetual, non-sublicenseable license, under patent claims owned or controlled by the Licensor that are embodied in the Original Work as furnished by the Licensor ("Licensed Claims") to make, use, sell and offer for sale the Original Work. Licensor hereby grants You a world-wide, royalty-free, non-exclusive, perpetual, non-sublicenseable license under the Licensed Claims to make, use, sell and offer for sale Derivative Works.

3) Grant of Source Code License. The term "Source Code" means the preferred form of the Original Work for making modifications to it and all available documentation describing how to modify the Original Work. Licensor hereby agrees to provide a machine-readable copy of the Source Code of the Original Work along with each copy of the Original Work that Licensor distributes. Licensor reserves the right to satisfy this obligation by placing a machine-readable copy of the Source Code in an information repository reasonably calculated to permit inexpensive and convenient access by You for as long as Licensor continues to distribute the Original Work, and by publishing the address of that information repository in a notice immediately following the copyright notice that applies to the Original Work.

4) Exclusions From License Grant. Nothing in this License shall be deemed to grant any rights to trademarks, copyrights, patents, trade secrets or any other intellectual property of Licensor except as expressly stated herein. No patent license is granted to make, use, sell or offer to sell embodiments of any patent claims other than the Licensed Claims defined in Section 2. No right is granted to the trademarks of Licensor even if such marks are included in the Original Work. Nothing in this License shall be interpreted to prohibit Licensor from licensing under different terms from this License any Original Work that Licensor otherwise would have a right to license.

5) External Deployment. The term "External Deployment" means the use or distribution of the Original Work or Derivative Works in any way such that the Original Work or Derivative Works may be used by anyone other than You, whether the Original Work or Derivative Works are distributed to those persons or made available as an application intended for use over a computer network. As an express condition for the grants of license hereunder, You agree that any External Deployment by You of a Derivative Work shall be deemed a distribution and shall be licensed to all under the terms of this License, as prescribed in section 1(c) herein.

6) Attribution Rights. You must retain, in the Source Code of any Derivative Works that You create, all copyright, patent or trademark notices from the

Source Code of the Original Work, as well as any notices of licensing and any descriptive text identified therein as an "Attribution Notice." You must cause the Source Code for any Derivative Works that You create to carry a prominent Attribution Notice reasonably calculated to inform recipients that You have modified the Original Work.

7) Warranty and Disclaimer of Warranty. Licensor warrants that the copyright in and to the Original Work is owned by the Licensor or that the Original Work is distributed by Licensor under a valid current license from the copyright owner. Except as expressly stated in the immediately proceeding sentence, the Original Work is provided under this License on an "AS IS" BASIS and WITHOUT WARRANTY, either express or implied, including, without limitation, the warranties of NON-INFRINGEMENT, MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY OF THE ORIGINAL WORK IS WITH YOU. This DISCLAIMER OF WARRANTY constitutes an essential part of this License. No license to Original Work is granted hereunder except under this disclaimer.

8) Limitation of Liability. Under no circumstances and under no legal theory, whether in tort (including negligence), contract, or otherwise, shall the Licensor be liable to any person for any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or the use of the Original Work including, without limitation, damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses. This limitation of liability shall not apply to liability for death or personal injury resulting from Licensor's negligence to the extent applicable law prohibits such limitation. Some jurisdictions do not allow the exclusion or limitation of incidental or consequential damages, so this exclusion and limitation may not apply to You.

9) Acceptance and Termination. If You distribute copies of the Original Work or a Derivative Work, You must make a reasonable effort under the circumstances to obtain the express and volitional assent of recipients to the terms of this License. Nothing else but this License (or another written agreement between Licensor and You) grants You permission to create Derivative Works based upon the Original Work or to exercise any of the rights granted in Sections 1 herein, and any attempt to do so except under the terms of this License (or another written agreement between Licensor and You) is expressly prohibited by U.S. copyright law, the equivalent laws of other countries, and by international treaty. Therefore, by exercising any of the rights granted to You in Sections 1 herein, You indicate Your acceptance of this License and all of its terms and conditions. This License shall terminate immediately and you may no

longer exercise any of the rights granted to You by this License upon Your failure to honor the proviso in Section 1(c) herein.

10) Mutual Termination for Patent Action. This License shall terminate automatically and You may no longer exercise any of the rights granted to You by this License if You file a lawsuit in any court alleging that any OSI Certified open source software that is licensed under any license containing this "Mutual Termination for Patent Action" clause infringes any patent claims that are essential to use that software.

11) Jurisdiction, Venue and Governing Law. Any action or suit relating to this License may be brought only in the courts of a jurisdiction wherein the Licensor resides or in which Licensor conducts its primary business, and under the laws of that jurisdiction excluding its conflict-of-law provisions. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any use of the Original Work outside the scope of this License or after its termination shall be subject to the requirements and penalties of the U.S. Copyright Act, 17 U.S.C. § 101 et seq., the equivalent laws of other countries, and international treaty. This section shall survive the termination of this License.

12) Attorneys Fees. In any action to enforce the terms of this License or seeking damages relating thereto, the prevailing party shall be entitled to recover its costs and expenses, including, without limitation, reasonable attorneys' fees and costs incurred in connection with such action, including any appeal of such action. This section shall survive the termination of this License.

13) Miscellaneous. This License represents the complete agreement concerning the subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable.

14) Definition of "You" in This License. "You" throughout this License, whether in upper or lower case, means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, "You" includes any entity that controls, is controlled by, or is under common control with you. For purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

15) Right to Use. You may use the Original Work in all ways not otherwise restricted or conditioned by this License or by law, and Licensor promises not to interfere with or be responsible for such uses by You.

ANNOTATIONS:

As an approved open source license drafted under the close guidance of OSI, itself, this license is intended to serve the same functions as the GPL has served for the Free Software Foundation, with a few notable exceptions:

(1) there is no contention by OSI that the OSL neatly avoids controversial issues regarding contract formation principles by being self-denominated as something other than a contractual obligation;

(2) the license contains an apparent weak copyleft, and

(3) the license contains a mutual termination clause as a defense to patent claims that undermine the goals of open source software.

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

The OSL is a license used to distribute open source works that constitute any original work of authorship, including software, documentation, music, art, or any other copyrightable work; traditionally and by contrast, the GPL was drafted as a technology license and has primarily applied only to software.

The first sentence of the license describes the mechanism for associating the OSL with an Original Work; there is no association provision in the GPL.

Section 1 is an explicit copyright grant, including all rights listed in 17 U.S.C. §106. The proviso in section 1(c) is a reciprocity provision equivalent to that in the GPL. This license is not sublicenseable, meaning that the licensor retains privity of contract (and hence the right to enforce this license) with all who obtain the Original Work.

Section 2 provides a "non-exclusive, perpetual, non-sublicenseable license, under patent claims owned," which is an explicit patent grant. The grant extends only to claims that are embodied in the Original Work. This license is also not sublicenseable, meaning that patent rights come directly from the licensor and the licensor can sue directly to enforce the patent license. Section 3 sets out the specific terms for how the licensee must ensure that the availability of source code must comply with the *open source* meaning of public access; in other words, how to lawfully access and modify the work.

In the definition of Source Code, the OSL requires that the licensee make available all documentation to modify the Original Work -- but not documentation to "access" it -- is what must be provided. To emphasize this point, the license contains a definition of "External Deployment" that highlights that the documentation requirement is not meant to besiege potential licensees or otherwise inhibit the acceptance of open source software.

What sets the OSL apart from other open source licenses is the terms that are regarded as a built-in defense against patent grants. There is an explicit patent grant from the licensor, but, more important, the license contains a mutual termination clause in the event there is an assertion of a patent interest that would disrupt the licensor's ability to maintain control over the project covered by the open source license.

The OSL also contains an explicit copyleft provision, but the provision may be viewed as weak since the license excludes the right to sublicense and does not contain an affirmative requirement regarding the licensee's obligation to cause any OSL software distributed to do so under the terms of the OSL.

Sections 4 and 5 also specifically exclude the licensor's trademark and other intellectual property from the license grant clause as well as contains a warranty that the licensor owns (or has the right to license) the Original Work.

The warranty provision differs from the GPL and most other open source licenses by explicitly providing an assurance to the licensee that the licensor has a lawful right to license the work at issue; subsequent users and licensees should find the warranty helpful as a potentially reliable safeguard, if litigation arises over the authority to use a work or, subsequently, if a demand for royalties is sought by an unexpected rights-holder.

The license contains a mutual termination clause in the event there is an assertion of a patent interest that would disrupt the licensor's ability to maintain control over the project covered by the open source license.

The OSL also makes it clear that this license is a contract. If the license is not accepted as a contract, however, no license is granted under copyright law. This is a departure from the GNU GPL, and may be incompatible with the GNU GPL under circumstances where the mutual termination clause imposes conditions upon the distribution of open source or free software licensed under the GPL, but distributed along with OSL software. The provision makes clear that the license is immediately terminated upon the licensee's failure to honor the reciprocity obligations of section 1(c).

Finally, the OSL contains a provision under section 9 that provides that any lawsuits relating to this license are to be maintained in the licensor's venue and under the laws of that jurisdiction. This venue clause is a provision that was believed necessary, but missing from the earlier iterations of the GNU GPL.

§ 22.01 The GNU Free Documentation License (the GFDL)

Website: http://www.fsf.org/licenses/fdlhtml

Copyright 2000 Free Software Foundation, Inc.

v. 1.1

Everyone is permitted to copy and distribute verbatim copies

of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other written document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you".

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (For example, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, whose contents can be viewed and edited directly and straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup has been designed to thwart or discourage subsequent modification by readers is not Transparent. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML designed for human modification. Opaque formats include PostScript, PDF, proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies of the Document numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a publicly-accessible computer-network location containing a complete Transparent copy of the Document, free of added material, which the general network-using public has access to download anonymously at no charge using public-standard network protocols. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has less than five).

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section entitled "History", and its title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. In any section entitled "Acknowledgements" or "Dedications", preserve the section's title, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section as "Endorsements" or to conflict in title with any Invariant Section.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections entitled “History” in the various original documents, forming one section entitled “History”; likewise combine any sections entitled “Acknowledgements”, and any sections entitled “Dedications”. You must delete all sections entitled “Endorsements.”

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, does not as a whole count as a Modified Version of the Document, provided no compilation copyright is claimed for the compilation. Such a compilation is called an “aggregate”, and this License does not apply to the other self-contained works thus compiled with the Document, on account of their being thus compiled, if they are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one quarter of the entire aggregate, the Document’s Cover Texts may be placed on covers that surround only the Document within the aggregate. Otherwise they must appear on covers around the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License provided that you also include the original English version of this License. In case of a disagreement between the translation and the original English version of this License, the original English version will prevail.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document 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.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation 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. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

> Copyright (c) YEAR YOUR NAME.
>
> Permission is granted to copy, distribute and/or modify this document
>
> under the terms of the GNU Free Documentation License, Version 1.1
>
> or any later version published by the Free Software Foundation;
>
> with the Invariant Sections being LIST THEIR TITLES, with the
>
> Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.
>
> A copy of the license is included in the section entitled "GNU
>
> Free Documentation License".

If you have no Invariant Sections, write "with no Invariant Sections" instead of saying which ones are invariant. If you have no Front-Cover Texts, write "no Front-Cover Texts" instead of "Front-Cover Texts being LIST"; likewise for Back-Cover Texts.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

"§ 1201. Circumvention of copyright protection systems

"(a) VIOLATIONS REGARDING CIRCUMVENTION OF TECHNOLOGICAL

MEASURES.—(1)(A) No person shall circumvent a technological measure that effectively controls access to a work protected under this title. The prohibition contained in the preceding sentence shall take effect at the end of the 2-year period beginning on

the date of the enactment of this chapter.

"(B) The prohibition contained in subparagraph (A) shall not apply to persons who are users of a copyrighted work which is in a particular class of works, if such persons are, or are likely to be in the succeeding 3-year period, adversely affected by virtue of such prohibition in their ability to make noninfringing uses of that particular class of works under this title, as determined under subparagraph (C).

"(C) During the 2-year period described in subparagraph (A), and during each succeeding 3-year period, the Librarian of Congress, upon the recommendation of the Register of Copyrights, who shall consult with the Assistant Secretary for Communications and Information of the Department of Commerce and report and

comment on his or her views in making such recommendation, shall make the determination in a rulemaking proceeding on the record for purposes of subparagraph (B) of whether persons who are users of a copyrighted work are, or are likely to be in the

succeeding 3-year period, adversely affected by the prohibition under subparagraph (A) in their ability to make noninfringing uses under this title of a particular class of copyrighted works.

In conducting such rulemaking, the Librarian shall examine—

"(i) the availability for use of copyrighted works;

"(ii) the availability for use of works for nonprofit archival, preservation, and educational purposes;

"(iii) the impact that the prohibition on the circumvention of technological measures applied to copyrighted works has on criticism, comment, news reporting, teaching, scholarship, or research;

"(iv) the effect of circumvention of technological measures on the market for or value of copyrighted works; and

"(v) such other factors as the Librarian considers appropriate.

"(D) The Librarian shall publish any class of copyrighted works for which the Librarian has determined, pursuant to the rulemaking conducted under subparagraph (C), that noninfringing uses by persons who are users of a copyrighted work are, or are likely to

be, adversely affected, and the prohibition contained in subparagraph (A) shall not apply to such users with respect to such class of works for the ensuing 3-year period.

"(E) Neither the exception under subparagraph (B) from the applicability of the prohibition contained in subparagraph (A), nor any determination made in a rulemaking conducted under subparagraph (C), may be used as a defense in any action to enforce

any provision of this title other than this paragraph.

"(2) No person shall manufacture, import, offer to the public, provide, or otherwise traffic in any technology, product, service, device, component, or part thereof, that—

"(A) is primarily designed or produced for the purpose of circumventing a technological measure that effectively controls access to a work protected under this title;

"(B) has only limited commercially significant purpose or use other than to circumvent a technological measure that effectively controls access to a work protected under this title;

"(C) is marketed by that person or another acting in concert with that person with that person's knowledge for use in circumventing a technological measure that effectively controls access to a work protected under this title.

"(3) As used in this subsection—

"(A) to 'circumvent a technological measure' means to descramble a scrambled work, to decrypt an encrypted work, or otherwise to avoid, bypass, remove, deactivate, or impair

a technological measure, without the authority of the copyright owner; and

"(B) a technological measure 'effectively controls access to a work' if the measure, in the ordinary course of its operation, requires the application of information, or a process or a treatment, with the authority of the copyright owner, to gain access to the work.

"§ 1201. Circumvention of copyright protection systems

"(b) ADDITIONAL VIOLATIONS.—(1) No person shall manufacture, import, offer to the public, provide, or otherwise traffic in any technology, product, service, device, component, or part thereof, that—

"(A) is primarily designed or produced for the purpose of circumventing protection afforded by a technological measure that effectively protects a right of a copyright owner under this title in a work or a portion thereof;

"(B) has only limited commercially significant purpose or use other than to circumvent protection afforded by a technological measure that effectively protects a right of a copyright owner under this title in a work or a portion thereof; or

"(C) is marketed by that person or another acting in concert with that person with that person's knowledge for use in circumventing protection afforded by a technological measure that effectively protects a right of a copyright owner under this title in a work or a portion thereof.

"(2) As used in this subsection—

"(A) to 'circumvent protection afforded by a technological

measure' means avoiding, bypassing, removing, deactivating, or otherwise impairing a technological measure; and

"(B) a technological measure 'effectively protects a right of a copyright owner under this title' if the measure, in the ordinary course of its operation, prevents, restricts, or otherwise limits the exercise of a right of a copyright owner under this title.

"(f) REVERSE ENGINEERING.—(1) Notwithstanding the provisions of subsection (a)(1)(A), a person who has lawfully obtained the right to use a copy of a computer program may circumvent a technological measure that effectively controls access to a particular portion of that program for the sole purpose of identifying and analyzing those elements of the program that are necessary to achieve interoperability of an independently created computer program with other programs, and that have not previously been readily available to the person engaging in the circumvention, to

the extent any such acts of identification and analysis do not constitute infringement under this title.

"(2) Notwithstanding the provisions of subsections (a)(2) and (b), a person may develop and employ technological means to circumvent a technological measure, or to circumvent protection afforded by a technological measure, in order to enable the identification

and analysis under paragraph (1), or for the purpose of enabling interoperability of an independently created computer program with other programs, if such means are necessary to achieve such interoperability, to the extent that doing so does not constitute

infringement under this title.

"(3) The information acquired through the acts permitted under paragraph (1), and the means permitted under paragraph (2), may be made available to others if the person referred to in paragraph (1) or (2), as the case may be, provides such information or means solely for the purpose of enabling interoperability of an independently

created computer program with other programs, and to the extent that doing so does not constitute infringement under this title or violate applicable law other than this section.

"(4) For purposes of this subsection, the term 'interoperability' means the ability of computer programs to exchange information, and of such programs mutually to use the information which has been exchanged.

....

"(4) USE OF TECHNOLOGICAL MEANS FOR RESEARCH ACTIVITIES.

—Notwithstanding the provisions of subsection (a)(2), it is not a violation of that subsection for a person to—

"(A) develop and employ technological means to circumvent a technological measure for the sole purpose of that person performing the acts of good faith encryption research described in paragraph (2); and

"(B) provide the technological means to another person with whom he or she is working collaboratively for the purpose of conducting the acts of good faith encryption

research described in paragraph (2) or for the purpose of having that other person verify his or her acts of good faith encryption research described in paragraph (2).

....

"§ 1203. Civil remedies

"(a) CIVIL ACTIONS.—Any person injured by a violation of section 1201 or 1202 may bring a civil action in an appropriate United States district court for such violation.

"(b) POWERS OF THE COURT.—In an action brought under subsection (a), the court—

"(1) may grant temporary and permanent injunctions on such terms as it deems reasonable to prevent or restrain a violation, but in no event shall impose a prior restraint on free speech or the press protected under the 1st amendment to the Constitution;

"(2) at any time while an action is pending, may order the impounding, on such terms as it deems reasonable, of any device or product that is in the custody or control of

the alleged violator and that the court has reasonable cause to believe was involved in a violation;

"(3) may award damages under subsection (c);

"(4) in its discretion may allow the recovery of costs by or against any party other than the United States or an officer thereof;

"(5) in its discretion may award reasonable attorney's fees to the prevailing party; and

"(6) may, as part of a final judgment or decree finding a violation, order the remedial modification or the destruction of any device or product involved in the violation that is in the custody or control of the violator or has been impounded under paragraph (2).

"(c) AWARD OF DAMAGES.—

"(1) IN GENERAL.—Except as otherwise provided in this title, a person committing a violation of section 1201 or 1202 is liable for either—

"(A) the actual damages and any additional profits of the violator, as provided in paragraph (2), or

"(B) statutory damages, as provided in paragraph (3).

"(2) ACTUAL DAMAGES.—The court shall award to the complaining party the actual damages suffered by the party as a result of the violation, and any profits of the violator

that are attributable to the violation and are not taken into account in computing the actual damages, if the complaining party elects such damages at any time before final judgment is entered.

"(3) STATUTORY DAMAGES.—(A) At any time before final judgment is entered, a complaining party may elect to recover an award of statutory damages for each violation of section 1201 in the sum of not less than $200 or more than $2,500 per act of circumvention, device, product, component, offer, or performance of service, as the court considers just.

"(B) At any time before final judgment is entered, a complaining party may elect to recover an award of statutory damages for each violation of section 1202 in the sum of not less than $2,500 or more than $25,000.

"(4) REPEATED VIOLATIONS.—In any case in which the injured party sustains the burden of proving, and the court finds, that a person has violated section 1201 or 1202 within 3 years after a final judgment was entered against the person for another such violation, the court may increase the award of damages up to triple the amount that would otherwise be awarded, as the court considers just.

“(5) INNOCENT VIOLATIONS.—

“(A) IN GENERAL.—The court in its discretion may reduce or remit the total award of damages in any case in which the violator sustains the burden of proving, and the court finds, that the violator was not aware and had no reason to believe that its acts constituted a violation.

“(B) NONPROFIT LIBRARY, ARCHIVES, OR EDUCATIONAL INSTITUTIONS.—In the case of a nonprofit library, archives, or educational institution, the court shall remit damages in any case in which the library, archives, or educational institution sustains the burden of proving, and the court finds, that the library, archives, or educational institution

was not aware and had no reason to believe that its acts constituted a violation.

“§ 1204. Criminal offenses and penalties

“(a) IN GENERAL.—Any person who violates section 1201 or 1202 willfully and for purposes of commercial advantage or private financial gain—

“(1) shall be fined not more than $500,000 or imprisoned for not more than 5 years, or both, for the first offense; and

“(2) shall be fined not more than $1,000,000 or imprisoned for not more than 10 years, or both, for any subsequent offense.

Circumventing the DMCA with open source

American copyright law aims to strike a balance between the public’s right to create works - - by building upon the work of others or works in the public domain - - and the exclusive right of authors of original works to control certain uses of their works. This balance was somewhat disrupted in 1998 when the United States Congress enacted the Digital Millennium Copyright Act (DMCA). Among other things, the DMCA prohibits the circumvention of any technological measure used to protect access to a copyright-protected work.

Under the authority of the DMCA, the doctrine of reverse engineering is being redefined in a manner that may retard innovation rather than promote it. Reverse engineering, under the DMCA may only protect a limited practice of decompilation of object code into source code authorized solely as an apparent economic efficiency primarily benefiting pertinent intellectual property owners. Apparently, the supporters of the DMCA successfully convinced congress to divorce the reverse engineering doctrine from its historical context in fair use and free expression. In doing so, proponents of the DMCA successfully removed the public's primary gateway of access to the ideas and functional elements embodied in proprietary software.

The first court to apply the DMCA to computer source code, *Universal Studios, Inc., v. Reimerdes*, rejected the argument that under the circumstances of the case the practice of reverse engineering copyright-protected works like a software program was supported by doctrinal as well as statutory privilege, which allows defendants to avoid the liability that would otherwise apply under the DMCA. According to both the district court and the appellate court upholding the district court's findings, if a software user successfully obtained access to source code by circumventing a technological "lock" or barrier to access without authorization from the copyright holder of the computer program, the circumvention would not be excused by the doctrine of reverse engineering under the DMCA or by relevant caselaw defining the same doctrine unless the defendant could persuade the court that her purpose for breaking the access control was to determine how two programs interoperate.

In *Reimerdes*, the plaintiffs, seven motion picture studios, brought suit under the DMCA, *inter alia*, seeking an injunction against the magazine and website, known as 2600 and 2600.com, (and several other defendants) to prohibit the defendants from publishing, disseminating, trafficking in, and posting on or linking to a website with the source code to a decryption program that allowed computer users of the Linux operating system (OS) to play lawfully acquired movies digitized by the copyright holder onto digital video discs (or DVDs).

The source code was alleged to provide a way of circumventing a technological access barrier; namely, encryption software called Content Scramble System Software (or CSS). This was accomplished by use of a decryption software program called, cleverly enough, DeCSS, which was developed after reverse engineering CSS. Ostensibly, at the time DeCSS was developed, DVDs were sold with CSS to allow users to play the content on DVD-ROM drives connected to computers running only one operating system, namely, the Windows operating system developed by Microsoft Corporation.

Reimerdes, both narrowed the scope of the defendants' conduct that could come within the statutory exception permitting reverse engineering, and limited the range of protection allowing public access to a work for the purpose of reverse engineering. In this regard, the court took the unprecedented step of restricting the reverse engineering exception to circumstances, involving the dissemination of information, **solely** for the purpose of achieving interoperability, among disparate computer programs -- which the court summarily, but mistakenly, determined was unlikely to be among the defendants' actual purposes -- given that, in the court's view, interoperability is not likely to have been the goal of the reverse engineer, if the immediate fruits of reverse engineering lead to the development of a decryption program designed to operate on the same OS, rather than immediately interoperate on a different OS.

In addition, the Court, rather conspicuously, failed to consider whether obtaining access to unprotectable ideas had been the basis of defendants' efforts of reverse engineering. Instead, the district court's analysis followed a lock-step argument presented by the plaintiffs concerning the copyright in the motion picture content on the DVD, which had nothing at all to do with reverse engineering the Content Scramble System Software (also known as CSS).

The district and appellate courts in *Reimerdes* is not the only court recently to question or doubt a defendant's claim to be engaged in legitimate or credible reverse engineering conduct in the context of software and, in this respect, there is a troubling trend toward increasing statutory and judicial approval of a software developer's prerogative to block public access to source code without exception, despite the appealing logic of the notion that copyright protection of the expressive quality of source code should result in its exposure, not its secrecy.

In a case related to *Reimerdes*, *DVDCAA v. Bunner*, a trial court rejected the argument that a 15 year old Norwegian programmer had engaged in lawful reverse engineering for purposes of interoperability. The court's skepticism seemed to have emanated from its concern that the 15 year old was an alleged hacker who had publicly declared his disrespect for the law of copyright. Although the plaintiff had sought injunctive relief under a state law trade secret misappropriation claim, rather than a claim under the DMCA, the court's application of the doctrine of reverse engineering, generally, would retard the public's access to source code, if applied to the DMCA in *Reimerdes*.

Reimerdes, in particular, so far has had the perverse effect of invalidating reverse engineering, when occurring in the context of the public dissemination of source code that contains information concerning how reverse engineering might be undertaken.

Ostensibly, section 1201(f) of the DMCA authorizes the circumvention of an access control by reverse engineering a copyright-protected work if: [a] the reverse engineering does not otherwise constitute copyright infringement, [b] the statutory definition of reverse engineering is met, which permits reverse engineering for the sole purpose to achieve interoperability, and [c] the defendant does not otherwise have a readily available alternative manner to access the information sought through reverse engineering.

Since the DMCA is silent on whether the reverse engineering exception applies to copy controls, a computer user or software developer may circumvent a copy control, if the control is identifiable as such, for the purpose of reverse engineering and need not rely upon the DMCA's narrowly drawn statutory definition of reverse engineering. It is difficult to distinguish an access control from a copy control, however. Even in *Reimerdes*, the plaintiffs claims alleged copy control violations and access control violations as if the two were interchangeable and treated similarly by the DMCA; of course, they are not, and the Act does not. If one were attempting to obtain the benefit of the DMCA's lack of a prohibition or silence on circumventing copy controls -- assuming they had the skill to create their own tool, since the market for these tools is precluded by section 1201(b) -- they would not be limited to the exception provided in section 1201(f), which presumably only applies to section 1201(a)(1)(A) issues (unless the defendant could identify the distinction between access and copy controls).

More to the point, if one could distinguish an access control from a copy control, the question remains whether the DMCA's prohibition from circumventing access controls would chill users from the practice of reverse engineering. In other words, the failure of the DMCA to provide specific guidance on how an access control differs from a copy control ostensibly renders the reverse engineering exception in section 1201(f) inaccessible to software developers and computer users in circumstances where the copyright holder does not make it evident what type of control has been established.

Indeed, there is no incentive for the right-holder to make it apparent what type of control is used to "protect" their work. Although some right-holders have found it useful to notify users that a technological barrier exist to protect the copyright interest in the work sold or licensed to the user, oftentimes, the description used to provide the notice is carelessly drafted so as to refer to access and copy control interchangeably.

More important, it is quite easy for a defendant to lose section 1201(f) protection if a court summarily determines that the reverse engineering was not done for the purpose set out in section 1201, which was the result of the courts' analysis in *Reimerdes.*

The reverse engineer has to show that there is no infringing conduct in the act of reverse engineering (other than the excused reverse engineering, itself), and that the reverse engineering does not infringe the copyright holder's interest in the access or copy control. Since some copyright holders may claim that a decryption program created for the purpose of overriding an encryption program's digital key lock is itself an infringement of the encryption program (i.e. the decryption program was derived from the encryption program), section 1201(f) would be unavailable to the defendant since the reverse engineering was not accomplished without otherwise infringing a copyright

Second, many software contracts purport to prohibit reverse engineering of the licensed software. These terms may conflict with a user's apparent right under copyright law to reverse engineer-copyrighted works for certain purposes. This is perhaps the most common example in the software industry of a conflict between contractual terms and copyright policy.

Third, software and digital information contract terms often seek to prohibit the licensee from moving a program to an upgraded computer or from altering, upgrading, or "debugging" the program. Such requirements may conflict with at least the spirit, and arguably the letter, of section 117, which gives users the right to copy and adapt the program to the extent necessary to run it on a particular machine.

In particular, section 117 was intended to give users the right to upgrade programs themselves, and to transfer software programs to newer hardware or operating systems, even if the transfer requires translation of the code.

Fourth, contract terms commonly prohibit licensees from transferring or assigning their particular copy of a work. Such provisions may conflict with the "first sale" doctrine in copyright law, which gives the owner of a particular copy of a copyrighted work the right to dispose of that copy without the permission of the copyright owner. Whether this is actually a conflict depends on whether the copyright owner "sold" or "licensed" the copy in question; the first sale doctrine does not prevent restrictions on the transfer of licensed items.

One interesting question that arises as a result of the enactment of the DMCA, at least for those developers within the jurisdiction of the United States, is whether an open source license that provides free and open access to source to downstream licensees should prohibit licensees from including access or copy controls in their open source software distributions.

This issue might be of particular concern as open source finds increasing support among developers of embedded systems and tethered device distributors. Under circumstances where an open source license carries a copyleft provision, the licensor may find it useful to include a clause the either directly prohibits subsequent licensees from adding access or copy controls to their software distribution or, less directly, offers a safe harbor from potential section 1201 liability in the event that an access control or copy control need to be circumvented to access the source code.

There are many circumstances where a copyleft clause will not be necessary, but, if its likely that a particular open source software program may be distributed as part of an embedded system or tethered device, either the anti-access control or anti-copy control provision will help ensure that both the source code remains unavailable to end-users and that the software slips into a proprietary product; tethered devices include software that is usually distributed as part of a hardware product, content package or media device such as; DVD-ROM or CD-ROM, which includes an encryption program, digital footprint, watermark, or similar technology; E-books with access controls and digital copyright management systems; streaming audio content; handheld or pocket devices with open source software or open content; and open source software distributed on proprietary media containing access controls.

Copyright Fundamentals

To profit from open source licensing, you must share the code

COPYRIGHT LAW

Copyright is a form of intellectual property protection provided to works that an author creates. Copyright protects ***original works of authorship*** that are fixed in a tangible form of expression. The fixation need not be directly perceptible so long as it may be communicated with the aid of a machine or device. Copyrightable works include the following categories:

(1) literary works;

(2) musical works, including any accompanying words

(3) dramatic works, including any accompanying music

(4) pantomimes and choreographic works

(5) pictorial, graphic, and sculptural works

(6) motion pictures and other audiovisual works

(7) sound recordings

(8) architectural works

These categories have been viewed broadly by courts. For example, computer programs and most "compilations" may be registered as "literary works"; maps and architectural plans may be registered as "pictorial, graphic, and sculptural works." In the United States, the Copyright Act generally gives the owner of copyright the exclusive right to do and to authorize others to do the following:

To reproduce the work in copies or phonorecords;

To prepare ***derivative works*** based upon the work;

To distribute copies or phonorecords of the work to the public by sale or other transfer of ownership, or by rental, lease, or lending;

To perform the work publicly, in the case of literary, musical, dramatic, and choreographic works, pantomimes, and motion pictures and other audiovisual works;

To display the copyrighted work publicly, in the case of literary, musical, dramatic, and choreographic works, pantomimes, and pictorial, graphic, or sculptural works, including the individual images of a motion picture or other audiovisual work; and

In the case of ***sound recordings, to perform the work publicly*** by means of a ***digital audio transmission.***

Despite recent news accounts of the Sonny Bono Copyright Act, copyright is not unlimited. The Copyright Act establishes limitations on copyright, and in some cases, those limitations are specified exemptions from copyright liability.

One major limitation, for example, is the doctrine of ***fair use***, which is given a statutory basis in *section 107* of the 1976 Copyright Act. In other instances, the limitation takes the form of a compulsory license under which certain limited uses of copyrighted works are permitted upon payment of specified royalties and compliance with statutory conditions.

Similarly, the Copyright Act limits the reach of Copyright protection by subject matter, wherein several categories of material are generally not eligible for federal copyright protection. These include among others: works that have not been fixed in a tangible form of expression (for example, choreographic works that have not been notated or recorded, or improvisational speeches or performances that have not been written or recorded) Titles, names, short phrases, and slogans; familiar symbols or designs; mere variations of typographic ornamentation, lettering, or coloring; mere listings of ingredients or contents Ideas, procedures, methods, systems, processes, concepts, principles, discoveries, or devices, as distinguished from a description, explanation, or illustration Works consisting entirely of information that is common property and containing no original authorship (for example: standard calendars, height and weight charts, tape measures and rulers, and lists or tables taken from public documents or other common sources).

Fair Use of Copyright-protected Works

Despite what might rightly be taken to be the well-known sentiments of those who opposed music file-sharing on Napster, not all unauthorized reproduction or copying of a copyright-protected work constitutes infringement. The Copyright Act explicitly provides for fair use in section 107. Since the Act essentially contains the language of the fair use doctrine developed by courts, the principle means of understanding how the doctrine might be applied to an actual issue is by study of court precedent. The doctrine appears to be under persistent evolution. The Copyright Act makes it clear, however, that:

"*Notwithstanding* the provisions of sections 106 and 106A, the fair use of a copyrighted work, including such use by reproduction in copies or phonorecords or by any other means specified by that section, for purposes such as criticism, comment, news reporting, teaching (including multiple copies for classroom use), scholarship, or research, is not an infringement of copyright. In determining whether the use made of a work in any particular case is a fair use the factors to be considered shall include-

(1) the purpose and character of the use, including whether such use is of a commercial nature or is for nonprofit educational purposes;

(2) the nature of the copyrighted work;

(3) the amount and substantiality of the portion used in relation to the copyrighted work as a whole; and

(4) the effect of the use upon the potential market for or value of the copyrighted work.

The fact that a work is unpublished shall not itself bar a finding of fair use if such finding is made upon consideration of all the above factors."

Reverse engineering is a defense to copyright infringement that frequently arises in the context of fair use. By use of the term "reverse engineering," we are referring to the reverse of the process wherein, a program written in source code is translated into object code using a computer program, often called a compiler, and then rendered suitable for both commercial distribution and execution in run-time. Reversing the process of compilation is less exacting than compiling, but essentially results in a decompiler reading the binary code of 0s and 1s and producing a more readable approximation of the original source code.

Indeed, courts often mistakenly conflate the statutory distinctions between the fair use privilege with the affirmative rights established as a part of reverse engineering in section 117. In this regard, courts have described the reverse engineering of a computer program for the purpose of determining its underlying ideas as a matter protected by fair use, rather than simply relying on the affirmative right established by section 117.

Today, copyright is secured automatically upon creation or fixation of the work. No publication or registration or other action in the Copyright Office is required to secure copyright. There are, however, certain definite advantages to registration. Copyright is secured automatically when the work is created, and a work is "created" when it is fixed in a copy or phonorecord for the first time. If a work is prepared over a period of time, the part of the work that is fixed on a particular date constitutes the created work as of that date.

PUBLICATION

Publication is no longer the key to obtaining federal copyright as it was under the Copyright Act of 1909. However, publication remains important to copyright owners. The 1976 Copyright Act defines publication as follows: "Publication" is the distribution of copies or phonorecords of a work to the public by sale or other transfer of ownership, or by rental, lease, or lending. The offering to distribute copies or phonorecords to a group of persons for purposes of further distribution, public performance, or public display constitutes publication. A public performance or display of a work does not of itself constitute publication.

Before 1978, federal copyright was generally secured by the act of publication with notice of copyright, assuming compliance with all other relevant statutory conditions. U. S. works in the public domain on January 1, 1978, (for example, works published without satisfying all conditions for securing federal copyright under the Copyright Act of 1909) remain in the public domain under the 1976 Copyright Act.

The use of a copyright notice is no longer required under U. S. law, although it is often beneficial. Because prior law did contain such a requirement, however, the use of notice is still relevant to the copyright status of older works. Notice was required under the 1976 Copyright Act. This requirement was eliminated when the United States adhered to the Berne Convention, effective March 1, 1989. Although works published without notice before that date could have entered the public domain in the United States, the Uruguay Round Agreements Act (URAA) restores copyright in certain foreign works originally published without notice.

Use of the notice may be important because it informs the public that the work is protected by copyright, identifies the copyright owner, and shows the year of first publication. Furthermore, in the event that a work is infringed, if a proper notice of copyright appears on the published copy or copies to which a defendant in a copyright infringement suit had access, then no weight shall be given to such a defendant's interposition of a defense based on innocent infringement in mitigation of actual or statutory damages, except as provided under copyright law. Innocent infringement occurs when the purported "infringer" did not realize that the work was protected. The use of the copyright notice is the responsibility of the copyright owner and does not require advance permission from, or registration with, the Copyright Office.

Form of Notice for Visually Perceptible Copies

The notice for visually perceptible copies should contain all the following three elements:

1. The symbol (c) (the letter C in a circle), or the word "Copyright," or the abbreviation "Copr."; and

2. The year of first publication of the work. In the case of compilations or derivative works incorporating previously published material, the year date of first publication of the compilation or derivative work is sufficient. The year date may be omitted where a pictorial, graphic, or sculptural work, with accompanying textual matter, if any, is reproduced in or on greeting cards, postcards, stationery, jewelry, dolls, toys, or any useful article; and

3. The name of the owner of copyright in the work, or an abbreviation by which the name can be recognized, or a generally known alternative designation of the owner.

Example: (c) 2001 Jane Doe

The "C in a circle" notice is used only on "visually perceptible copies." Certain kinds of works-for example, musical, dramatic, and literary works-may be fixed not in "copies" but by means of sound in an audio recording. Since audio recordings such as audio tapes and phonograph disks are "phonorecords" and not "copies," the "C in a circle" notice is not used to indicate protection of the underlying musical, dramatic, or literary work that is recorded.

Period of Copyright Protection

A work that is created (fixed in tangible form for the first time) on or after January 1, 1978, is automatically protected from the moment of its creation and is ordinarily given a term enduring for the ***author's life plus an additional 70 years*** after the author's death. In the case of "a joint work prepared by two or more authors who did not work for hire," the term lasts for 70 years after the last surviving author's death. For ***works made for hire***, and for anonymous and pseudonymous works (unless the author's identity is revealed in Copyright Office records), the duration of copyright will be 95 years from publication or 120 years from creation, whichever is shorter.

Works Originally Created before January 1, 1978

These works have been automatically brought under the statute and are now given federal copyright protection. The duration of copyright in these works will generally be computed in the same way as for works created on or after January 1, 1978: the life-plus-70 or 95/120-year terms will apply to them as well. The law provides that in no case will the term of copyright for works in this category expire before December 31, 2002, and for works published on or before December 31, 2002, the term of copyright will not expire before December 31, 2047.

Regarding works originally created and published or registered before January 1, 1978, under the law in effect before 1978, copyright was secured either on the date a work was published with a copyright notice or on the date of registration if the work was registered in unpublished form. In either case, the copyright endured for a first term of 28 years from the date it was secured. During the last (28th) year of the first term, the copyright was eligible for renewal.

The Copyright Act of 1976 extended the renewal term from 28 to 47 years for copyrights that were subsisting on January 1, 1978, or for pre-1978 copyrights restored under the Uruguay Round Agreements Act (URAA), making these works eligible for a total term of protection of 75 years. Public Law 105-298, enacted on October 27, 1998, further extended the renewal term of copyrights still subsisting on that date by an additional 20 years, providing for a renewal term of 67 years and a total term of protection of 95 years. Public Law 102-307, enacted on June 26, 1992, amended the 1976 Copyright Act to provide for automatic renewal of the term of copyrights secured between January 1, 1964, and December 31, 1977. Although the renewal term is automatically provided, the Copyright Office does not issue a renewal certificate for these works unless a renewal application and fee are received and registered in the Copyright Office.

Finally, congress recently made renewal registration optional. Thus, filing for renewal registration is no longer required in order to extend the original 28-year copyright term to the full **95** years. However, some benefits accrue from making a renewal registration during the 28th year of the original term

Transfer of Copyright

Any or all of the copyright owner's exclusive rights or any subdivision of those rights may be transferred, but the transfer of exclusive rights is not valid unless that transfer is in writing and signed by the owner of the rights conveyed or such owner's duly authorized agent. Transfer of a right on a nonexclusive basis does not require a written agreement. A copyright may also be conveyed by operation of law and may be bequeathed by will or pass as personal property by the applicable laws of in testate succession.

Since copyright is viewed as a personal property right, it is subject to the various state laws and regulations that govern the ownership, inheritance, or transfer of personal property as well as terms of contracts or conduct of business. For information about relevant state laws, consult your attorney. A transfer of copyright, which is generally called an assignment, is made by a written contract. The law does provide for the recordation in the Copyright Office of transfers of copyright ownership. Although recordation is not required to make a valid transfer between the parties, it does provide certain legal advantages and may be required to validate the transfer as against third parties.

Under the previous law, the copyright in a work reverted to the author, if living, or if the author was not living, to other specified beneficiaries, provided a renewal claim was registered in the 28th year of the original term. The present law drops the renewal feature except for works already in the first term of statutory protection when the present law took effect. Instead, the present law permits termination of a grant of rights after 35 years under certain conditions by serving written notice on the transferee within specified time limits.

The copyright in works eligible for renewal on or after June 26, 1992, will vest in the name of the renewal claimant on the effective date of any renewal registration made during the 28th year of the original term. Otherwise, the renewal copyright will vest in the party entitled to claim renewal as of December 31st of the 28th year. For works already under statutory copyright protection before 1978, the present law provides a similar right of termination covering the newly added years that extended the former maximum term of the copyright from 56 to 95 years.

International Copyright

There is no such thing as an "international copyright" that will automatically protect an author's writings throughout the entire world. Protection against unauthorized use in a particular country depends, basically, on the national laws of that country. However, most countries do offer protection to foreign works under certain conditions, and these conditions have been greatly simplified by international copyright treaties and conventions.

Copyright Registration

In general, copyright registration is a legal formality intended to make a public record of the basic facts of a particular copyright. However, registration is not a condition of copyright protection. Even though registration is not a requirement for protection, the copyright law provides several inducements or advantages to encourage copyright owners to make registration; among these advantages includes establishment of a public record of the copyright claim. Before an infringement suit may be filed in court, registration is necessary for works of U. S. origin. If made before or within 5 years of publication, registration will establish prima facie evidence in court of the validity of the copyright and of the facts stated in the certificate. If registration is made within 3 months after publication of the work or prior to an infringement of the work, statutory damages and attorney's fees will be available to the copyright owner in court actions. Otherwise, only an award of actual damages and profits is available to the copyright owner.

Registration may be made at any time within the life of the copyright. Unlike the law before 1978, when a work has been registered in unpublished form, it is not necessary to make another registration when the work becomes published, although the copyright owner may register the published edition, if desired.

Although a copyright registration is not required, the Copyright Act establishes a mandatory deposit requirement for works published in the United States. See the definition of "publication." In general, the owner of copyright or the owner of the exclusive right of publication in the work has a legal obligation to deposit in the Copyright Office, within 3 months of publication in the United States, two copies (or in the case of sound recordings, two phonorecords) for the use of the Library of Congress. Failure to make the deposit can result in fines and other penalties but does not affect copyright protection. In litigation, for the copyright holder to establish "ownership" of the copyright in software, the copyright holder must be careful to comply with all of the statutory formalities for copyright registration.

A copyright holder has complied with all statutory formalities for copyright registration when the Copyright office receives the application for registration, fee, and deposit. Although the 1976 amendment to the copyright law extinguished a number of rigid formalities, some formal requirements persist. For example, even if the software developer has submitted a deposit of the appropriate number of lines of source code with the application and payment of fee, the registration ultimately may be deemed defective or not complete if the developer failed to submit for deposit the *original* source code for the software; for the initial filing, later versions of source code will not meet this requirement.

Certain categories of works are exempt entirely from the mandatory deposit requirements, and the obligation is reduced for certain other categories. There is no requirement that applications be prepared or filed by an attorney. Copyright protection subsists from the time the work is created in fixed form. The copyright in the work of authorship ***immediately*** becomes the property of the author who created the work. Only the author or those deriving their rights through the author can rightfully claim copyright. In the case of works made for hire, the employer and not the employee is considered to be the author. Section 101 of the copyright law defines a "work made for hire" as:

(1) a work prepared by an employee within the scope of his or her employment; or

(2) a work specially ordered or commissioned for use as:

a contribution to a collective work

a part of a motion picture or other audiovisual work

a translation

a supplementary work

a compilation

an instructional text

a test

answer material for a test

a sound recording

an atlas

if the parties expressly agree in a written instrument signed by them that the work shall be considered a work made for hire....

The authors of a joint work are co-owners of the copyright in the work, unless there is an agreement to the contrary.

Copyright in each separate contribution to a periodical or other collective work is distinct from copyright in the collective work as a whole and vests initially with the author of the contribution.

Notably, neither mere ownership of a book, manuscript, painting, or any other copy of a work, nor the transfer of ownership of any material object that embodies a protected work conveys any rights in the copyright.

Copyright Infringement for Software

To establish copyright infringement, a plaintiff must prove "(1) ownership of a valid copyright, and (2) copying of constituent elements of the work that are original." There is an inherent tension in the need to protect copyrighted material as well as allow others to build upon that material. Copyright has the bold binary task of both protecting the financial interests of authorship and ensuring public access to the works of authorship. To show ownership of a valid copyright, the author or developer must prove that the work as a whole is original and that the author complied with applicable statutory formalities such as, in a judicial proceeding, presenting a court with a copy of a certificate of copyright registration, which constitutes prima facie evidence of copyrightability and shifts the burden to the alleged infringer to demonstrate why the copyright is not valid.

Software copyright infringement cases are ostensibly disputes over one party's monopoly over the commands it uses to operate a computer. Courts have attempted to resolve cases involving copyright infringement allegations regarding software by either determining whether similarities between two computer programs are due merely to the fact that the two works share the same underlying idea or whether they instead indicate that the second author copied the first author's literal or nonliteral copyrightable expression.

For some courts, an assessment of copyright infringement for software is nothing more than a mechanical application of a court invented 3-part test often identified as: abstraction, filtration, and comparison. Although the abstraction-filtration-comparison test could be applied to software with an appropriate degree of rigor, courts often use the test as a conclusory classification system at best. Abstraction requires courts to isolate each level of abstraction contained within the program within which to separate protectable expression from unprotected ideas. Filtration requires an examination of the program to determine whether the expression included at a particular level of abstraction exists because it was dictated by considerations of efficiency, required by factors external to the program itself, or taken from the public domain. Comparison occurs using the expression remaining, if any, after the unprotectable expression has been filtered out of the analysis. Comparison involves the protected elements of the infringed work (those that survived the filtration screening) matched up against the corresponding elements of the allegedly infringing work to determine whether there was sufficient copying of protected material to constitute infringement.

Often, in software copyright infringement cases, parties seem to re-argue the legislative debate concerning whether a particular software program can be copyrighted at all; although this is a fundamental question, it is one to which Congress already has provided an answer and courts are without power to reject. Consequently, the critical inquiry in these cases should not be whether the software program as a whole can be copyrighted, but, rather, whether the snippets of source code that genuinely form the basis of dispute constitutes copyrightable expression.

In this respect, courts can give effect to the Copyright Act's limitation under section 102(b), which states: "[i]n no case does copyright protection for an original work of authorship extend to any idea, procedure, process, system, method of operation, concept, principle, or discovery, regardless of the form in which it is described, explained, illustrated, or embodied in such work." Copyright assures authors the right to their original expression, but also encourages others to build freely upon the ideas and information conveyed by a work. This is true of all types of literary works, including software. That the law of copyright assures authors the right to their original expression does not require that courts find that all expression in software is necessarily copyrightable. Notably, the primary objective of copyright is not to reward the labor of authors, but to promote the Progress of Science and useful Arts. To promote the progress of science and art, Copyright must make certain that its protections are not so vast that they overwhelm Copyright, itself, and take on the singular task of protecting a legally privileged, tightly controlled, system of producing works of authorship.

To prevail in an action alleging copyright infringement, the copyright holder must prove ownership of the copyright and that the infringer violated at least one of those exclusive rights. In this respect, the exclusive right most commonly alleged to have been violated is right to reproduce or copy the work. Often, unauthorized reproduction is shown by circumstantial evidence of access to the software program's source code, which usually requires evidence that the computer code was decompiled or unlawfully reverse engineered, if the software was produced for a proprietary software project.

Reverse engineering usually involves a technical process that includes making a so-called "intermediate copy" of the computer program. The intermediate copy is alleged to be an unauthorized reproduction of the software and, hence, constitutes at least one instance of copyright infringement. In the software context, the existence of an intermediate copy arises as a by product of the inherent "use" of software. Since "use" and "copy" seem to closely resemble the same act when applied to software, there have been commentators who have asked whether the copyright conception of "copy" provides a meaningful basis to establish a distinct copyrightable interest. In other words, the question arises whether "copy" consumes too much scope of copyright for software to be worthy of the exclusive interest – when weighed in the balance against the public domain. Why not exclude "copy" from the bundle of interests, but allow wide-ranging exploitation of derivative rights in our recalibration of the scope of copyright protection of software?

Ever since a court erroneously expanded the scope of copyright in software in 1990, when Lotus Corporation succeeded in its copyright infringement litigation over its Lotus 1-2-3 spreadsheet against Paperback Software Company (and its limited success against Borland Corporation), there has been a need for the open source community. Although the harbingers of open source are found in the late 1970s or early 1980s, the need for a movement that might supplant the growing incapacity of courts to thoughtfully assess and evaluate the bold claims of copyright infringement among an increasing number of aggressive software developers was clearly apparent by the early 1990s.

Within a year after Lotus' astonishing success in court, software developers throughout the IT industry seemed set to expand the strength of their copyright, not by producing source code, but by litigating dubious or outright bogus claims copyright infringement. Since many courts were ill-prepared to assess claims involving the copyright protection of software, these courts seemed to developers like Midas when they unwittingly turned many dubious claims to gold. During these years, Lotus sued Borland, Borland sued Lotus, Apple Computer sued Microsoft, Microsoft sued Harmony Computers, Xerox sued Apple, and Apple sued Hewlett Packard, and on went the litigation. Although many of these cases were either settled or did not reach final judgment, the effect of the litigation and the cases that did reach judicial resolution is an expanded expectation by many software developers that they have a thick or wide, rather than thin or narrow, scope of copyright protection for software.

The Digital Millennium Copyright Act (DMCA)

For progress in "Science and useful Arts" to occur, others must be permitted to build upon and refer to the creations of prior thinkers. In this regard, copyright interests -- and their potentially stifling effects upon the ability to build on prior works -- must yield to the public's desire to make fair use of copyrighted works under clearly delineated conditions.

As noted earlier in the chapter, American copyright law aims to strike a balance between the public's right to create works -- by building upon the work of others or works in the public domain -- and the exclusive right of authors of original works to control certain uses of their works. This balance was disrupted in 1998 when the United States Congress enacted the Digital Millennium Copyright Act (DMCA).

Among other things, the DMCA prohibits the circumvention of any technological measure used to protect access to a copyright-protected work. One reason why the DMCA is particularly troubling for the open source community is that the DMCA -- which contains a thicket of legal lumber diminishing fair use rights -- only authorizes an exceptionally narrow use of reverse engineering. Indeed, what emanates from the text of the DMCA and recent judicial interpretations of the reverse engineering doctrine is a shocking complete acceptance, if not reverence, by congress and the courts of the copyright holder's desire to restrict the scope of lawful reverse engineering without accommodating an important access point of the public to public domain material.

Under the authority of the DMCA, the doctrine of reverse engineering is being redefined in a manner that may retard innovation rather than promote it. Reverse engineering, under the DMCA may only protect a limited practice of decompilation of object code into source code authorized solely as an apparent economic efficiency primarily benefiting pertinent intellectual property owners. Apparently, the supporters of the DMCA successfully convinced congress to divorce the reverse engineering doctrine from its historical context in fair use and free expression. In doing so, proponents of the DMCA successfully removed the public's primary gateway of access to the ideas and functional elements embodied in proprietary software.

That this privilege of access is being replaced by a technological barrier sanctioned by the U.S. Congress with the force of civil and criminal liability is demonstrative of how close Americans have come to an ill-fated future for the public domain of information and information products. This does not simply promise a future where the user of digital information will pay an "owner" for its use in each and every instance, but includes a potential guarantee that some owners will be more equal than others. For some, copyright law will become an increasing patchwork of legislative favoritism dislodged from its constitutional purpose.

The DMCA and the Hackers Who Promote Open Source

It is with unfortunate irony that this trend provides an apparent cover for content-based speech restrictions directed toward the activities of some libertarian-oriented Internet-based computer hackers and the open source community, whose activities are focused upon unleashing the locked up ideas of software, instead of hiding them. Perhaps, it is most ironic that it is the open source community, in particular, that demonstrates that a wide-ranging network of programmers can develop robust, popular, and reliable software, where source code is open and freely available to the public, not hidden by access controls or technological barriers of any sort, including computer code made available only in binary or unreadable form. To thwart the most recent legislative and judicial barriers to expanding open source beyond its current successes, the use of open source licensing must undergo a "professionalization" process, whereby the terms of typical open source licenses are bolstered in a manner that is meaningful to prevailing legal norms. This is covered in the next chapter.

SELECTED PROVISIONS OF THE DIGITAL MILLENNIUM COPYRIGHT ACT (DMCA)

112 STAT. 2860 PUBLIC LAW 105–304—OCT. 28, 1998

SEC. 103. COPYRIGHT PROTECTION SYSTEMS AND COPYRIGHT MANAGEMENT INFORMATION.

(a) IN GENERAL.—Title 17, United States Code, is amended by adding at the end the following new chapter:

"CHAPTER 12—COPYRIGHT PROTECTION AND MANAGEMENT SYSTEMS

"Sec.

"1201. Circumvention of copyright protection systems.

"1202. Integrity of copyright management information.

"1203. Civil remedies.

"1204. Criminal offenses and penalties.

"1205. Savings clause.

"§ 1201. Circumvention of copyright protection systems

"(a) VIOLATIONS REGARDING CIRCUMVENTION OF TECHNOLOGICAL

MEASURES.—(1)(A) No person shall circumvent a technological measure that effectively controls access to a work protected under this title. The prohibition contained in the preceding sentence shall take effect at the end of the 2-year period beginning on

the date of the enactment of this chapter.

"(B) The prohibition contained in subparagraph (A) shall not apply to persons who are users of a copyrighted work which is in a particular class of works, if such persons are, or are likely to be in the succeeding 3-year period, adversely affected by virtue of such prohibition in their ability to make noninfringing uses of that particular class of works under this title, as determined under subparagraph (C).

''(C) During the 2-year period described in subparagraph (A), and during each succeeding 3-year period, the Librarian of Congress, upon the recommendation of the Register of Copyrights, who shall consult with the Assistant Secretary for Communications and Information of the Department of Commerce and report and

comment on his or her views in making such recommendation, shall make the determination in a rulemaking proceeding on the record for purposes of subparagraph (B) of whether persons who are users of a copyrighted work are, or are likely to be in the

succeeding 3-year period, adversely affected by the prohibition under subparagraph (A) in their ability to make noninfringing uses under this title of a particular class of copyrighted works.

In conducting such rulemaking, the Librarian shall examine—

''(i) the availability for use of copyrighted works;

''(ii) the availability for use of works for nonprofit archival, preservation, and educational purposes;

''(iii) the impact that the prohibition on the circumvention of technological measures applied to copyrighted works has on criticism, comment, news reporting, teaching, scholarship, or research;

''(iv) the effect of circumvention of technological measures on the market for or value of copyrighted works; and

''(v) such other factors as the Librarian considers appropriate.

''(D) The Librarian shall publish any class of copyrighted works for which the Librarian has determined, pursuant to the rulemaking conducted under subparagraph (C), that noninfringing uses by persons who are users of a copyrighted work are, or are likely to be, adversely affected, and the prohibition contained in subparagraph (A) shall not apply to such users with respect to such class of works for the ensuing 3-year period.

"(E) Neither the exception under subparagraph (B) from the applicability of the prohibition contained in subparagraph (A), nor any determination made in a rulemaking conducted under subparagraph (C), may be used as a defense in any action to enforce

any provision of this title other than this paragraph.

"(2) No person shall manufacture, import, offer to the public, provide, or otherwise traffic in any technology, product, service, device, component, or part thereof, that—

"(A) is primarily designed or produced for the purpose of circumventing a technological measure that effectively controls access to a work protected under this title;

"(B) has only limited commercially significant purpose or use other than to circumvent a technological measure that effectively controls access to a work protected under this title;

"(C) is marketed by that person or another acting in concert with that person with that person's knowledge for use in circumventing a technological measure that effectively controls access to a work protected under this title.

"(3) As used in this subsection—

"(A) to 'circumvent a technological measure' means to descramble a scrambled work, to decrypt an encrypted work, or otherwise to avoid, bypass, remove, deactivate, or impair

a technological measure, without the authority of the copyright owner; and

"(B) a technological measure 'effectively controls access to a work' if the measure, in the ordinary course of its operation, requires the application of information, or a process or a treatment, with the authority of the copyright owner, to gain access to the work.

"(b) ADDITIONAL VIOLATIONS.—(1) No person shall manufacture, import, offer to the public, provide, or otherwise traffic in any technology, product, service, device, component, or part thereof, that—

"(A) is primarily designed or produced for the purpose of circumventing protection afforded by a technological measure that effectively protects a right of a copyright owner under this title in a work or a portion thereof;

"(B) has only limited commercially significant purpose or use other than to circumvent protection afforded by a technological measure that effectively protects a right of a copyright owner under this title in a work or a portion thereof; or

"(C) is marketed by that person or another acting in concert with that person with that person's knowledge for use in circumventing protection afforded by a technological measure that effectively protects a right of a copyright owner under this title in a work or a portion thereof.

"(2) As used in this subsection—

"(A) to 'circumvent protection afforded by a technological

measure' means avoiding, bypassing, removing, deactivating, or otherwise impairing a technological measure; and

"(B) a technological measure 'effectively protects a right of a copyright owner under this title' if the measure, in the ordinary course of its operation, prevents, restricts, or otherwise limits the exercise of a right of a copyright owner under this title.

"(c) OTHER RIGHTS, ETC., NOT AFFECTED.—(1) Nothing in this section shall affect rights, remedies, limitations, or defenses to copyright infringement, including fair use, under this title.

"(2) Nothing in this section shall enlarge or diminish vicarious or contributory liability for copyright infringement in connection with any technology, product, service, device, component, or part thereof.

"(3) Nothing in this section shall require that the design of, or design and selection of parts and components for, a consumer electronics, telecommunications, or computing product provide for a response to any particular technological measure, so long as such part or component, or the product in which such part or component is integrated, does not otherwise fall within the prohibitions of subsection (a)(2) or (b)(1).

"(4) Nothing in this section shall enlarge or diminish any rights of free speech or the press for activities using consumer electronics, telecommunications, or computing products.

"(d) EXEMPTION FOR NONPROFIT LIBRARIES, ARCHIVES, AND EDUCATIONAL

INSTITUTIONS.—(1) A nonprofit library, archives, or educational institution which gains access to a commercially exploited copyrighted work solely in order to make a good faith determination of whether to acquire a copy of that work for the sole purpose

of engaging in conduct permitted under this title shall not be in violation of subsection (a)(1)(A). A copy of a work to which access has been gained under this paragraph—

"(A) may not be retained longer than necessary to make such good faith determination; and

"(B) may not be used for any other purpose.

"(2) The exemption made available under paragraph (1) shall only apply with respect to a work when an identical copy of that work is not reasonably available in another form.

"(3) A nonprofit library, archives, or educational institution that willfully for the purpose of commercial advantage or financial gain violates paragraph (1)—

"(A) shall, for the first offense, be subject to the civil remedies under section 1203; and

"(B) shall, for repeated or subsequent offenses, in addition to the civil remedies under section 1203, forfeit the exemption provided under paragraph (1).

"(4) This subsection may not be used as a defense to a claim under subsection (a)(2) or (b), nor may this subsection permit a nonprofit library, archives, or educational institution to manufacture, import, offer to the public, provide, or otherwise traffic in any technology, product, service, component, or part thereof, which circumvents a technological measure.

"(5) In order for a library or archives to qualify for the exemption under this subsection, the collections of that library or archives shall be—

"(A) open to the public; or

"(B) available not only to researchers affiliated with the library or archives or with the institution of which it is a part, but also to other persons doing research in a specialized

field.

"(e) LAW ENFORCEMENT, INTELLIGENCE, AND OTHER GOVERNMENT

ACTIVITIES.—This section does not prohibit any lawfully authorized investigative, protective, information security, or intelligence activity of an officer, agent, or employee of the United States, a State, or a political subdivision of a State, or a person acting pursuant to a contract with the United States, a State, or a political subdivision of a State. For purposes of this subsection, the term 'information security' means activities carried out in order to identify and address the vulnerabilities of a government computer,

computer system, or computer network.

"(f) REVERSE ENGINEERING.—(1) Notwithstanding the provisions of subsection (a)(1)(A), a person who has lawfully obtained the right to use a copy of a computer program may circumvent a technological measure that effectively controls access to a particular portion of that program for the sole purpose of identifying and analyzing those elements of the program that are necessary to achieve interoperability of an independently created computer program with other programs, and that have not previously been readily available to the person engaging in the circumvention, to the extent any such acts of identification and analysis do not constitute infringement under this title.

"(2) Notwithstanding the provisions of subsections (a)(2) and (b), a person may develop and employ technological means to circumvent a technological measure, or to circumvent protection afforded by a technological measure, in order to enable the identification

and analysis under paragraph (1), or for the purpose of enabling interoperability of an independently created computer program with other programs, if such means are necessary to achieve such interoperability, to the extent that doing so does not constitute

infringement under this title.

“(3) The information acquired through the acts permitted under paragraph (1), and the means permitted under paragraph (2), may be made available to others if the person referred to in paragraph (1) or (2), as the case may be, provides such information or means solely for the purpose of enabling interoperability of an independently

created computer program with other programs, and to the extent that doing so does not constitute infringement under this title or violate applicable law other than this section.

“(4) For purposes of this subsection, the term ‘interoperability’ means the ability of computer programs to exchange information, and of such programs mutually to use the information which has been exchanged.

. . . .

“(4) USE OF TECHNOLOGICAL MEANS FOR RESEARCH ACTIVITIES.

—Notwithstanding the provisions of subsection (a)(2), it is not a violation of that subsection for a person to—

“(A) develop and employ technological means to circumvent a technological measure for the sole purpose of that person performing the acts of good faith encryption research described in paragraph (2); and

“(B) provide the technological means to another person with whom he or she is working collaboratively for the purpose of conducting the acts of good faith encryption research described in paragraph (2) or for the purpose of having that other person verify his or her acts of good faith encryption research described in paragraph (2).

….

"(i) PROTECTION OF PERSONALLY IDENTIFYING INFORMATION.—

(1) CIRCUMVENTION PERMITTED.—Notwithstanding the provisions of subsection (a)(1)(A), it is not a violation of that subsection for a person to circumvent a technological measure that effectively controls access to a work protected under this

title, if—

"(A) the technological measure, or the work it protects, contains the capability of collecting or disseminating personally identifying information reflecting the online activities of a natural person who seeks to gain access to the work protected;

Deadline.

"(B) in the normal course of its operation, the technological measure, or the work it protects, collects or disseminates personally identifying information about the person

who seeks to gain access to the work protected, without providing conspicuous notice of such collection or dissemination to such person, and without providing such person

with the capability to prevent or restrict such collection or dissemination;

"(C) the act of circumvention has the sole effect of identifying and disabling the capability described in subparagraph (A), and has no other effect on the ability

of any person to gain access to any work; and

"(D) the act of circumvention is carried out solely for the purpose of preventing the collection or dissemination of personally identifying information about a natural person who seeks to gain access to the work protected, and is not in violation of any other law.

"(2) INAPPLICABILITY TO CERTAIN TECHNOLOGICAL MEASURES.—This subsection does not apply to a technological measure, or a work it protects, that does not collect or disseminate personally identifying information and that is disclosed to a user as not having or using such capability.

"(j) SECURITY TESTING.—

"(1) DEFINITION.—For purposes of this subsection, the term 'security testing' means accessing a computer, computer system, or computer network, solely for the purpose of good faith testing, investigating, or correcting, a security flaw or vulnerability, with the authorization of the owner or operator of such computer, computer system, or computer network.

"(2) PERMISSIBLE ACTS OF SECURITY TESTING.—Notwithstanding the provisions of subsection (a)(1)(A), it is not a violation of that subsection for a person to engage in an act of security testing, if such act does not constitute infringement under this title or a violation of applicable law other than this section, including section 1030 of title 18 and those provisions of title 18 amended by the Computer Fraud and Abuse Act of 1986.

"(3) FACTORS IN DETERMINING EXEMPTION.—In determining whether a person qualifies for the exemption under paragraph (2), the factors to be considered shall include—

"(A) whether the information derived from the security testing was used solely to promote the security of the owner or operator of such computer, computer system or

computer network, or shared directly with the developer of such computer, computer system, or computer network; and

"(B) whether the information derived from the security testing was used or maintained in a manner that does not facilitate infringement under this title or a violation of applicable law other than this section, including a violation of privacy or breach of security.

"(4) USE OF TECHNOLOGICAL MEANS FOR SECURITY TESTING.

—Notwithstanding the provisions of subsection (a)(2), it is not a violation of that subsection for a person to develop, produce, distribute or employ technological means for the sole purpose of performing the acts of security testing described

....

"(b) REMOVAL OR ALTERATION OF COPYRIGHT MANAGEMENT

INFORMATION.—No person shall, without the authority of the copyright owner or the law—

"(1) intentionally remove or alter any copyright management information,

"(2) distribute or import for distribution copyright management information knowing that the copyright management information has been removed or altered without authority of the copyright owner or the law, or

"(3) distribute, import for distribution, or publicly perform works, copies of works, or phonorecords, knowing that copyright management information has been removed or altered without authority of the copyright owner or the law, knowing, or, with respect to civil remedies under section 1203, having reasonable grounds to know, that it will induce, enable, facilitate, or conceal an infringement of any right under this title.

"(c) DEFINITION.—As used in this section, the term 'copyright management information' means any of the following information conveyed in connection with copies or phonorecords of a work or performances or displays of a work, including in digital form, except that such term does not include any personally identifying information

about a user of a work or of a copy, phonorecord, performance, or display of a work:

"(1) The title and other information identifying the work, including the information set forth on a notice of copyright.

"(2) The name of, and other identifying information about, the author of a work.

"(3) The name of, and other identifying information about, the copyright owner of the work, including the information set forth in a notice of copyright.

"(4) With the exception of public performances of works by radio and television broadcast stations, the name of, and other identifying information about, a performer whose performance is fixed in a work other than an audiovisual work.

"(5) With the exception of public performances of works by radio and television broadcast stations, in the case of an audiovisual work, the name of, and other identifying information about, a writer, performer, or director who is credited in the audiovisual work.

"(6) Terms and conditions for use of the work.

"(7) Identifying numbers or symbols referring to such information or links to such information.

"(8) Such other information as the Register of Copyrights may prescribe by regulation, except that the Register of Copyrights may not require the provision of any information concerning the user of a copyrighted work.

"(d) LAW ENFORCEMENT, INTELLIGENCE, AND OTHER GOVERNMENT

ACTIVITIES.—This section does not prohibit any lawfully authorized investigative, protective, information security, or intelligence activity of an officer, agent, or employee of the United States, a State, or a political subdivision of a State, or a person acting pursuant to a contract with the United States, a State, or a political subdivision of a State. For purposes of this subsection, the term 'information security' means activities carried out in order to identify and address the vulnerabilities of a government computer,

computer system, or computer network.

"(e) LIMITATIONS ON LIABILITY.—

"(1) ANALOG TRANSMISSIONS.—In the case of an analog transmission, a person who is making transmissions in its capacity as a broadcast station, or as a cable system, or someone who provides programming to such station or system, shall not be liable for a violation of subsection (b) if—

"(A) avoiding the activity that constitutes such violation is not technically feasible or would create an undue financial hardship on such person; and

"(B) such person did not intend, by engaging in such activity, to induce, enable, facilitate, or conceal infringement of a right under this title.

"(2) DIGITAL TRANSMISSIONS.—

"(A) If a digital transmission standard for the placement of copyright management information for a category of works is set in a voluntary, consensus standard-setting

process involving a representative cross-section of broadcast stations or cable systems and copyright owners of a category of works that are intended for public performance

by such stations or systems, a person identified in paragraph

(1) shall not be liable for a violation of subsection

(b) with respect to the particular copyright management information addressed by such standard if—

"(i) the placement of such information by someone other than such person is not in accordance with such standard; and

"(ii) the activity that constitutes such violation is not intended to induce, enable, facilitate, or conceal infringement of a right under this title.

"(B) Until a digital transmission standard has been set pursuant to subparagraph (A) with respect to the placement of copyright management information for a category or works, a person identified in paragraph (1) shall not be liable for a violation of subsection (b) with respect to such copyright management information, if the activity that constitutes such violation is not intended to induce, enable, facilitate, or conceal infringement of a right under this title, and if—

"(i) the transmission of such information by such person would result in a perceptible visual or aural degradation of the digital signal; or "(ii) the transmission of such information by such person would conflict with—

"(I) an applicable government regulation relating to transmission of information in a digital signal;

"(II) an applicable industry-wide standard relating to the transmission of information in a

digital signal that was adopted by a voluntary consensus standards body prior to the effective date of this chapter; or

"(III) an applicable industry-wide standard relating to the transmission of information in a digital signal that was adopted in a voluntary, consensus standards-setting process open to participation by a representative cross-section of broadcast stations or cable systems and copyright owners of a category of works that are intended for public performance by such stations or systems.

"(3) DEFINITIONS.—As used in this subsection—

"(A) the term 'broadcast station' has the meaning given that term in section 3 of the Communications Act of 1934 (47 U.S.C. 153); and

"(B) the term 'cable system' has the meaning given that term in section 602 of the Communications Act of 1934 (47 U.S.C. 522).

"§ 1203. Civil remedies

"(a) CIVIL ACTIONS.—Any person injured by a violation of section 1201 or 1202 may bring a civil action in an appropriate United States district court for such violation.

"(b) POWERS OF THE COURT.—In an action brought under subsection (a), the court—

"(1) may grant temporary and permanent injunctions on such terms as it deems reasonable to prevent or restrain a violation, but in no event shall impose a prior restraint on free speech or the press protected under the 1st amendment to the Constitution;

"(2) at any time while an action is pending, may order the impounding, on such terms as it deems reasonable, of any device or product that is in the custody or control of

the alleged violator and that the court has reasonable cause to believe was involved in a violation;

"(3) may award damages under subsection (c);

"(4) in its discretion may allow the recovery of costs by or against any party other than the United States or an officer thereof;

"(5) in its discretion may award reasonable attorney's fees to the prevailing party; and

"(6) may, as part of a final judgment or decree finding a violation, order the remedial modification or the destruction of any device or product involved in the violation that is in the custody or control of the violator or has been impounded under paragraph (2).

"(c) AWARD OF DAMAGES.—

"(1) IN GENERAL.—Except as otherwise provided in this title, a person committing a violation of section 1201 or 1202 is liable for either—

"(A) the actual damages and any additional profits of the violator, as provided in paragraph (2), or

"(B) statutory damages, as provided in paragraph (3).

"(2) ACTUAL DAMAGES.—The court shall award to the complaining party the actual damages suffered by the party as a result of the violation, and any profits of the violator

that are attributable to the violation and are not taken into account in computing the actual damages, if the complaining party elects such damages at any time before final judgment is entered.

"(3) STATUTORY DAMAGES.—(A) At any time before final judgment is entered, a complaining party may elect to recover an award of statutory damages for each violation of section 1201 in the sum of not less than $200 or more than $2,500 per act of circumvention, device, product, component, offer, or performance of service, as the court considers just.

"(B) At any time before final judgment is entered, a complaining party may elect to recover an award of statutory damages for each violation of section 1202 in the sum of not less than $2,500 or more than $25,000.

"(4) REPEATED VIOLATIONS.—In any case in which the injured party sustains the burden of proving, and the court finds, that a person has violated section 1201 or 1202 within 3 years after a final judgment was entered against the person for another such violation, the court may increase the award of damages up to triple the amount that would otherwise be awarded, as the court considers just.

“(5) INNOCENT VIOLATIONS.—

“(A) IN GENERAL.—The court in its discretion may reduce or remit the total award of damages in any case in which the violator sustains the burden of proving, and the court finds, that the violator was not aware and had no reason to believe that its acts constituted a violation.

“(B) NONPROFIT LIBRARY, ARCHIVES, OR EDUCATIONAL INSTITUTIONS.—In the case of a nonprofit library, archives, or educational institution, the court shall remit damages in any case in which the library, archives, or educational institution sustains the burden of proving, and the court finds, that the library, archives, or educational institution

was not aware and had no reason to believe that its acts constituted a violation.

“§ 1204. Criminal offenses and penalties

“(a) IN GENERAL.—Any person who violates section 1201 or 1202 willfully and for purposes of commercial advantage or private financial gain—

“(1) shall be fined not more than $500,000 or imprisoned for not more than 5 years, or both, for the first offense; and

“(2) shall be fined not more than $1,000,000 or imprisoned for not more than 10 years, or both, for any subsequent offense.

“(b) LIMITATION FOR NONPROFIT LIBRARY, ARCHIVES, OR EDUCATIONAL

INSTITUTION.—Subsection (a) shall not apply to a nonprofit library, archives, or educational institution.

“(c) STATUTE OF LIMITATIONS.—No criminal proceeding shall be brought under this section unless such proceeding is commenced within 5 years after the cause of action arose.

About the Author

Rod Dixon teaches and writes about the intersections of law and technology, particularly in the context of the legal, cultural, and political dimensions of computer-mediated communications. Mr. Dixon's published scholarship has focused primarily upon an examination of electronic privacy, intellectual property and software, and Internet governance. He has served as chief counsel for FreeBuyer's Net, an Internet service provider; as a law professor at Rutgers University Law School in Camden, New Jersey; and as an attorney at the U.S. Department of Education in Washington, D.C. He is a native of Philadelphia, Pennsylvania, and a graduate of the University of Pittsburgh, where he received his B.A. at the age of 19. Mr. Dixon also received his M.A. from the University of Pittsburgh, his J.D. from George Washington University Law School, and his LL.M. from Georgetown University Law Center. He resides in Silver Spring, Maryland. Quite apart from vocational or scholarly pursuits, Mr. Dixon is a genuine computer hobbyist; he enjoys digital photography and amateur robotics; and maintains a Web log at Cyberspaces.org. He also enjoys painting and traveling, and views himself as a potential novelist.

Index