Michael Hall
Law for Artists and Writers
Dr. Clark
12 June 2003

From Free Software to Open Content: An Overview of Copyleft

For the past two decades, spurred on by the ubiquity of both inexpensive personal computer technology and access to the Internet, a technical movement founded on idealistic communitarian principles and a desire to use copyright law to preserve some measure of artist's rights in the distribution of software has grown. Known as the "Free Software" or "Open Source" movement, it has driven such notable successes as the Linux computer operating system, the Apache Web server, and myriad less visible but equally sophisticated pieces of software. Though frequently characterized as an "anti-copyright" movement, Free Software exists almost solely because of the protections conventional copyright law offers authors. Since gaining public visibility in the late 1990s, the Free Software movement has inspired others in non-technical fields to bring communal values to other creative endeavors, such as writing and photography, providing a response to the growing restrictiveness of the entertainment industry. The most recent culmination of the Free Software spirit, often referred to as "copyleft," is the Creative Commons license, which provides an accessible and legally sound way for authors to release their creative efforts into the public domain while ensuring a certain amount of moral control remains in place.

1. Technological Roots of the Copyleft Ethos

In 1984, a computer programmer working for MIT's Artificial Intelligence Lab named Richard M. Stallman founded an software authoring project he referred to as "GNU," which stood for the recursive title "GNU's Not UNIX." Since the 1970s, UNIX was the standard-

bearing computer operating system (the software responsible for letting a computer run programs and interact with its users), for academics and companies. Much of the software that currently allows the Internet to function, for instance, was initially conceived and authored on UNIX systems. A property of Bell Labs, UNIX the frequently subject of litigation as a result of Bell's desire to maintain proprietary control of its software. Stallman's goal was to recreate the UNIX operating system as a series of freely available work-alike programs that would provide a familiar environment for computer programmers used to the way UNIX worked without requiring expensive licenses from Bell. Stallman was heavily influenced by his experiences at MIT during the '70s, a period during which many academic programmers were slowly hired away from universities by new private sector companies that insisted on treating software less as a collection of communal recipes and more as important intellectual property handled with the care of trade secrets in any other industry. Though Stallman has been derided by critics for his "anti-property" views, there's a strongly Jeffersonian streak in his thinking when he likens software to cooking recipes:

> " . . . recipes are the closest thing in everyday pre-computer life to a computer program. It is a set of things to be done in a particular order with certain inputs and rules for how to tell when a step is done, and sometimes you have to go back and repeat a step. It's like an algorithm except that it's carried out by a cook and not by a computer. And of course, if you look at the things that people want to do with a recipe, it is much like software. They want to be free to use a recipe, to pass it along copies to other cooks. They want to be free to modify the recipe--you know, not put in so much salt--

> depending on how good a cook you are; a better cook can make bigger modifications and get something interesting." (Stallman, "Free Software and Beyond")

Stallman called the software he sought to create for the GNU project "Free Software," and he eventually founded the Free Software Foundation. An important semantic point should be noted at this point. In modern English, "free" is often used to denote an item or service that's available *gratis*, and less to denote that an item or person is unrestricted. Stallman meant for his software to be "free" in the sense that it would be unencumbered by the sort of restrictions proprietary software companies placed on their products.

As explained by Leaffer, software commonly appears in two forms: as source code, which is readable by humans; and as object code, which is readable by computers (Leaffer, 102). Object code is easily copied and shared, but provides little insight to how a piece of software has been written: it acts as a sort of black box end users can interact with that reveals little of the alogrithms or programming techniques that drive it. To an academic programmer like Stallman, access to source code was much more important, since it represented access to the solutions fellow developers applied to programming challenges. Consequently, he focused much less on distributing object code *gratis* to other computer enthusiasts and much more on seeing to it that the source code for the programs comprising the GNU project would be freely available to all. To that end, he created the GNU Public License (GPL), which is nearly ten printed pages long, but can be distilled to a simple set of rules:

> 1. Users may run the GNU licensed program for any purpose
> 2. Users have the freedom to modify the program to suit their needs (which effectively requires access to the source code, since making

> meaningful changes to object code is exceedingly difficult and beyond the knowledge of even a skilled programmer).
>
> 3. Users have the freedom to distribute modified versions of the program so the developer community at large may benefit from the improvements. They may do so gratis, or for payment that doesn't exceed the cost of the media on which the software is distributed.
>
> 4. Users must make changes to GNU-licensed software freely available if they distribute that software to others in binary (object code) format. (Hall, 212)

The GPL came to be known colloquially as a "copyleft license." To the minds of a programming community that usually encountered copyright law as a tool of proprietary interest, the GPL must have seemed like the antithesis of copyright since it sought to force improvements to software into the public domain where they could be enjoyed by all instead of being locked up for exclusive gain. The GPL is not, of course, any such thing: its conditions may be novel, but they represent the terms of a license like any other, stipulating three distinct elements of a non-exclusive license: the manner in which a piece of software may be used, circumstances under which the license may be terminated, and the name to be carried on the copyright notice (Leaffer, 215).

The GPL doesn't stop at simple creation of a license, and Stallman was interested in more than creating a novel way to share software, as he has noted:

> "With a free operating system, we could again have a community of cooperating [programmers] – and invite anyone to join. And anyone would be able to use a computer without starting out by conspiring

to deprive his or her friends." (Hall, 21)

There was, in other words, a strong moral component to Stallman's project as he sought to use his software to recreate a sharing community driven by generosity and mutualism. In a way, the GPL not only provided a way to release software into the public domain in such a way that its value as a public good was enhanced with each improvement, it also provided Stallman a way to exercise lasting control over his software in a manner US copyright law would not tend to recognize, by making Stallman's own distinctive values, sharing knowledge for the betterment of a community, compulsory as a precursor to a developer's full enjoyment of GNU software. Every time a software programmer borrows code from a GNU program then dutifully passes her improvements along to others, that software has become even more of an "extension of the author's personality" (Leaffer, 361), by virtue of the very communal nature of its improvement.

The GPL isn't the only such license that seeks to ensure the availability of source code to the public. Another variant, originally devised at the University of California at Berkeley is called the "Berkeley Software Distribution" (BSD) license, which lays no claim on the creative efforts of its licensees, and instead merely offers software under the license free to all takers, who are under no compulsion to publish their changes. Both BSD- and GNU-licensed code is present in the Apple Macintosh operating system, and Apple regularly publishes its improvements to several GPL-licensed software projects (Searls). Yet another variant, the Perl Artistic License, named for a popular programming language used to program Web sites and perform routine system administration tasks by technology professionals, requires either redistribution of modified source code *or* simply renaming modified programs so users don't confuse them with the original.

In the twenty years since the GNU project and Free Software Foundation were launched, many software programmers have followed Stallman's lead and released their software under the GPL or similar licenses. In conjunction with the United Nations Education, Scientific and Cultural Organization, the FSF maintains a database of such software that reports a total of 2,230 software packages released under Free Software licenses. The software runs the gamut from specialized programming tools to a program known as "the GNU Image Manipulation Program," which provides many of the capabilities of Adobe's popular Photoshop software (FSF, "FSF/UNESCO Free Software Directory"). It should also be noted that Free Software isn't limited to UNIX systems. GNU software is available for Macintoshes, PC's running Microsoft Windows, and even Palm handheld computers.

2. Corporate Interaction with Free Software

There are two compelling problems faced by Free Software advocates. First is the issue of enforceability, second is the conflict between the proprietary impulse of most software houses and the mandatory openness of the GPL and its cousins.

Object code, the machine-readable software computers use when they run a software program, provides little hint of the source code that constituted it before it was compiled into binary form. For that reason, developers can easily borrow source code from a program, compile that source code into object code, and few would have any way of determining any sort of copyright infringement had taken place. In fact, the Free Software community is largely dependent on the honesty of the public and the occasional whistleblower working within a corporate firewall to ensure improvements to its source code are shared.

Free Software Foundation chief legal counsel Eben Moglen, a Columbia University law professor, reports that software companies are aware of the GPL and that their legal teams

regularly examine their internal use of GPL-licensed software to ensure no infringement of the license is taking place. In cases where infringement has occurred and the infringing company has been confronted, "I have secured nonjudicial enforcement of the GPL numerous times [...] never failing of cooperation even from reluctant entities once the legal situation had been explored by them," he's said (Powell).

One widely discussed case in 2001 involved a software company that appropriated GPL-licensed code then failed to release its changes and improvements, as the GPL stipulates. A developer at the company blew the whistle (Tai, Advogato), Moglen confronted the company with a letter explaining his understanding of the situation and suggested how to bring the infringing software into compliance with the provisions of the GPL, and they complied, to the extent the company republished the letter on its site (Moglen, “Letter to Vidomi”) and continues to offer an assertion of its “good citizenship” on its corporate site to this day (Vidomi, homepage).

Because of Moglen's preference for non-judicial solutions to potential infringement issues, there has yet to be a case where the GPL itself has been upheld by a court, which is a matter of some speculation and occasional concern on the part of Free Software advocates, who have operated under the license's provisions for two decades.

Companies have also come up with some interesting variants on the GPL in order to benefit from Free Software, produce competitive products, and remain in the good graces of Free Software developers.

Netscape Corporation, for instance, released the source code to its Web browser to reinvigorate the faltering software and to harness the collective imagination of the Free Software programming community, which the company anticipated would gladly contribute to a major

commercial software package (Netscape). Faced, however, with the concern that Netscape licensed proprietary, non-GPL-licensed software from a variety of vendors, the company created what it called the "Mozilla Public License" (Mozilla is the internal code-name for Netscape's browser software), which ensures that source code for much of the project is available while respecting proprietary elements of the software (Mozilla Project, "The Mozilla Public License").

A Norwegian software firm called TrollTech offers a "dual license", which allows software programmers to use its proprietary software development programs in commercial products that don't involve source code release only if they pay a fee, while Free Software developers are welcome to use its software *gratis* provided they release their source code under a GPL-like license (TrollTech, "Qt Free Licensing").

Collectively, these license are referred to as "Open Source" licenses, a term coined to alleviate some of the confusion between "free" as in *gratis* and "free" as in "freely redistributable and unencumbered." An organization called the Open Source Initiative maintains a list of licenses it considers in broad compliance with the goals of the Free Software movement, (OSI, "Approved Licenses") though for political reasons it largely describes itself as more interested with the methodological benefits of sharing code and less with the moral issues Stallman found so compelling.

3. The Linux Movement

Most of the Free Software movement was invisible to the public for the first 15 years of its existence. Though computer programmers were familiar with GNU software, which often provided superior functionality to the proprietary software on which it was patterned, the general public had no reason to pay any attention. In the late 1990s, that changed with the growing buzz surrounding the Linux computer operating system.

Developed in the early 1990s by a Finnish computer science student named Linus Torvalds, Linux provided an entire standalone software environment that could be used to run an IBM-style personal computer in a manner very similar to UNIX, which had traditionally run only on much more expensive and sophisticated systems. Linux was heavily dependent on software from the GNU project, and was licensed under the GPL as well, but succeeded where GNU had not in terms of popularity because of its ability to run on home computers, which made it popular with student computer scientists who were familiar with UNIX systems from their studies and wanted to use it away from the classroom. Amateur computer enthusiasts were also fond of Linux because it allowed them to experiment with home networking and Web publishing for free. As more and more enthusiast programmers wrote software to help Linux become usable by everyday people, more and more of the public came in contact with the ideas of Free Software and Copyleft. This, in turn, led to a number of efforts to contribute to the Free Software movement by non-programmers.

Initially, these efforts were unsophisticated. Some artistically inclined Linux enthusiasts created art meant to be used as "wallpaper" on computer screens (Propaganda) and playfully "licensed" the art for Linux computers only (Microsoft users were forbidden to download the images). Small fan Web sites and "advocacy groups" liberally applied "copyleft" to their published efforts. Some groups of technical writers contributed their skills to writing documentation for various Linux-based programs, which they released under a document license the GNU Project put together (FSF, "GNU Free Documentation License"), but the fit was never very comfortable for a variety of reasons: the GPL is very specifically oriented toward computer source code, and it leaves some cultural issues between artists and programmers unanswered. While a programmer may be perfectly happy to release her software to the world and welcome

any optimization or improvement that befalls it, writers and other artists may be less eager to see their artistic expression "improved." While they may enjoy the thought that their poetry is widely shared, and a communitarian impulse little different from Richard Stallman's may drive that sharing, they may not want to have modified copies of their poetry credited to them if "improvements" render their art atrocious.

A founding member of the Open Source Initiative, Eric Raymond, noted the difference between software and other sorts of content:

> "Music and most books are not like software, because they don't generally need to be debugged or maintained, [...] I do not want to weaken the winning argument for open sourcing software by tying it to a potential loser." (Lawton, "The Great Giveaway")

Nonetheless, there were several attempts during the late 1990s to develop a non-software version of the GPL, including the aforementioned GNU Documentation License (GDL) and the "Design Science License," which made provisions for requiring the documentation of changes to a source text and the renaming of derivative works to avoid confusion with their source (Stutz, "The Design Science License") . Within the relatively rarefied world of computer enthusiasts, these licenses were embraced and used frequently, but they were generally *ad hoc* creations from within the enthusiast community, and they were addressed to the concerns and needs of enthusiasts. Within the context of the general public, copyleft was still an obscure concept applicable to little but software.

4. The Creative Commons

In the late 1990s, the sort of intellectual property issues the Free Software movement formed to address became more prominent in the minds of everyday, non-technical

members of the public with the rise of Internet file sharing and the attendant backlash from the software and entertainment industries, which responded to widespread and casual copyright infringement by the likes of Napster users by pushing for legislation such as the Digital Millenium Copyright Act (Leaffer, 372) and the even more restrictive Security Systems Standards and Certification Act (SSSCA), which, had it passed, would have made sale of computer equipment that didn't include built-in copy protection technology illegal (McCullaugh). As the music industry fought for tougher sanctions against file sharing, other elements of the entertainment industry pushed for legislation such as the Sonny Bono Copyright Term Extension Act of 1998, which extended the term of copyright from seventy-five years to ninety-five years in the case of works of corporate authorship (Thomas).

These efforts came to the attention of a burgeoning movement within the law community, most prominently Stanford Professor of Law and the court-appointed "special master" in Microsoft's antitrust trial Lawrence Lessig, who argued :

> " . . . the reason perfect control has not been our tradition's aim is that creation always involves building upon something else. There is no art that doesn't reuse. And there will be less art if every reuse is taxed by the appropriator. Monopoly controls have been the exception in free societies; they have been the rule in closed societies." (Lessig, "May the Source Be With You")

Lessig was no stranger to the field of computer technology or the Free Software movement when he wrote that. In the same article, he noted:

> "Copyright law doesn't require the release of source code because it is believed that software would become unprotectable. The open

source movement might throw that view into doubt," (ibid)

He echoed Stallman's concerns, expressed decades earlier, when he said

> "While the creative works from the 16th century can still be accessed and used by others, the data in some software programs from the 1990s is already inaccessible. Once a company that produces a certain product goes out of business, it has no simple way to uncover how its product encoded data. The code is thus lost, and the software is inaccessible. Knowledge has been destroyed." (ibid)

Stallman and Lessig have, in fact, developed deep affinities, to the extent Lessig likened Stallman to Moses of the Old Testament, calling him "leader of a critically important aspect of modern culture." (Lessig, *The Future of Ideas,* 270).

Lessig was also a vocal opponent of the Sonny Bono Act, arguing against it before the Supreme Court in Eldred vs. Ashcroft, a case that would have had the act struck down as unconstitutional had the plaintiff prevailed. As "Eldred" winded its way through court, Lessig became the chair of the Creative Commons project, which took on as its first project the creation of the Creative Commons Licenses.

According to the Creative Commons organization, the GNU project's GPL was an inspiration for the effort, but the organization also acknowledged that it was ill-suited for the sort of creative work for which the licenses had been designed:

> "Unlike the GNU GPL, Creative Commons licenses are not be designed for software, but rather for other kinds of creative works: websites, scholarship, music, film, photography, literature,

> courseware, etc. We hope to build upon and complement the work of others who have created public licenses for a variety of creative works. Our aim is not only to increase the sum of raw source material online, but also to make access to that material cheaper and easier," (Creative Commons, "About Us").

Whereas the GPL is fairly inflexible in how it defines the terms of the license it provides, the Creative Commons (CC) Licenses were created as a suite of complementary licenses authors could mix and match in three broad categories: attribution, derivative works, and commercial use:

> **Attribution:** Authors may either require attribution or not when their work is used by a licensee.
>
> **Commercial use:** Authors may forbid commercial use of their work by licensees.
>
> **Derivative Works:** Authors may permit derivative works without qualification, with the requirement that derivative works be CC-licensed, or not at all.

As with the GPL, hints of preserving some of the moral right again came into play in the composition of the CC licenses: authors who find commercial use of a work repugnant may forbid it while still "sharing" their content, and authors inspired by the "share and share alike" ethos of the Free Software movement may require that behavior from their licensees where derivative works are involved. At the same time, however, the licenses explicitly stipulate that once the author has released a work under a Creative Commons License, the work remains so

licensed indefinitely. To a certain extent, the concept of the author's moral right is undermined by this provision, since the author may never "recall" a work in wide use.

The project not only formulated the licenses, but it made them slightly less forbidding to implement than the GPL, which requires a careful reading of its ten pages to understand its full impact. Authors visiting the Creative Commons Web site may step through a "license wizard" of sorts that allows selection of each possible license element and produces a finished Web page to which the author may link, in which the terms are explained in the simplest manner possible (Appendix 1). The project also provides hidden, machine-readable "metadata" search engines can use to provide indexes of Creative Commons-licensed material on the Web (Appendix 2).

The extent to which the Creative Commons licensing scheme will take hold remains to be seen. The licenses were released late in 2002, and the project has begun to index Web sites where the licenses are in use (Creative Commons, "Features"). One notable Creative Commons licensor, author Cory Doctorow, released a science fiction novel through Tor Publishing entitled *Down and Out in the Magic Kingdom*, which was made available both as a print copy and as a freely downloadable electronic edition. At the time of this writing, Doctorow hasn't commented on overall sales, but did report the title, which sold for $16.95 on Amazon.com, was ranked 304th in sales at the online bookstore (Doctorow) for a period. It currently resides around 700th place over six months after its release.

5. Concluding Notes

In a period dominated by copyright extension acts and rhetoric about the absolute sanctity of intellectual property, it's easy to lose sight of the foundations of copyright in the US

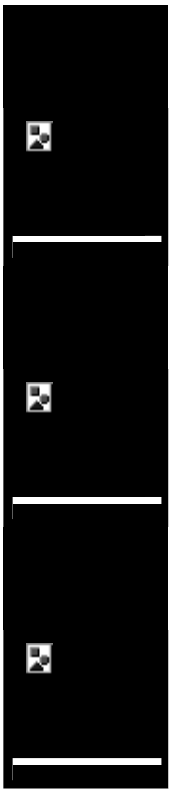
Constitution. This is so much the case, in fact, that even advocates of copyleft in the Free Software movement have been impassioned in their defense of Napster and other infringing technologies on the specious grounds that copyright was formulated solely for the enrichment of the individual, not for “the progress of science and the useful arts.”

As demonstrated here, though, “copyleft” isn't so much an inversion of copyright law as it is an affirmation of the value of copyright, and an attempt to use licenses to reinject some notions of the moral right back into the common conception of intellectual property. A Richard Stallman or Lawrence Lessig may appear, in the current climate, to be opposed to intellectual property. Their efforts, however, are reflective of an abiding concern for society's future as more and more intellectual property is held by large corporations with the influence and longevity necessary to ensure that this generation's Shakespeare (or Edison, as much as he'd hate being used in such an illustration) will remain locked behind a corporate firewall until no further profit can be extracted from them.

Appendix 1: A Creative Commons License Web Page

Attribution-NonCommercial-ShareAlike 1.0

Key License Terms:



Attribution. The licensor permits others to copy, distribute, display, and perform the work. In return, licensees must give the original author credit.

Noncommercial. The licensor permits others to copy, distribute, display, and perform the work. In return, licensees may not use the work for commercial purposes -- unless they get the licensor's permission.

Share Alike. The licensor permits others to distribute derivative works only under a license identical to the one that governs the licensor's work.

Whoever has associated this Commons Deed with their copyrighted work licenses his or her work to you on the terms of the Creative Commons License found here:

Legal Code (the full license)

This is not a license. It is simply a handy reference for understanding the Legal Code (the full license) - it is a human-readable expression of some of its key terms. Think of it as the user-friendly interface to the Legal Code beneath. This Deed itself has no legal value, and its contents do not appear in the actual license.

Appendix 2: Creative Commons “Metadata”

In addition to the HTML code allows Web browsers to present pages to readers in human readable form, there's a kind of HTML called “RDF” (for “Resource Description Framework”) that is not seen by human readers but is indexable by search engines and software programmed to look for it. The Creative Commons organization provides sample metadata that could allow a search engine to eventually produce an index of all the CC-licensed content available on the Internet. Below is an example of this metadata, which can be copied and pasted into any document without affecting the way it looks or reads to the casual visitor:

```
<!-- Creative Commons License -->
<a href="http://creativecommons.org/{license URL}">
<img alt="Creative Commons License" border="0"
src="http://creativecommons.org/images/public/somerights" /></a>
<br />
This work is licensed under a
<a href="http://creativecommons.org/{license URL}">Creative Commons
License</a>.
<!-- /Creative Commons License -->
<!--
<rdf:RDF xmlns="http://web.resource.org/cc/"
    xmlns:dc="http://purl.org/dc/elements/1.1/"
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
<Work rdf:about="">
<license rdf:resource="http://creativecommons.org/{license URL}" />
</Work>
</rdf:RDF>
-->
```

A page bearing this data may be found at http://www.puddingbowl.org/mph/. Note how a small graphical icon on the lower left of the page links to the Creative Commons site. By using a Web browser's “view source” option, it's possible to look toward the bottom of the HTML source code for the hidden RDF data.

Works Cited

Creative Commons. “About Us”. *Creative Commons Home Page.* <http://creativecommons.org/learn/aboutus/>

Creative Commons. “Features”. *Creative Commons Home Page.* <http://creativecommons.org/works/>

Doctorow, Cory. “Over 20,000 Downloads So Far”. *Personal Web Site*. <http://www.craphound.com/down/archives/2003_01.php#000029>

Free Software Foundation. “FSF/UNESCO Free Software Directory”. *FSF Home Page.* <http://www.gnu.org/directory/>

Free Software Foundation. “GNU Free Documentation License”. *FSF Home Page.* <http://www.gnu.org/copyleft/fdl.html>

Hall, Michael and Proffitt, Brian. *The Joy Of Linux: A Gourmet Guide to Open Source*. Roseville, CA: Prima Publishing. 2001

Lawton, Graham. “The Great Giveaway”. *The New Scientist*. <http://www.newscientist.com/hottopics/copyleft/copyleftart.jsp>

Leaffer, Marshall. *Understanding Copyright Law*. Third Ed. New York: Matthew Bender and Co. 2001

Lessig, Lawrence. “May the Source Be With You”. *WIRED*. Dec 2001. <http://www.wired.com/wired/archive/9.12/lessig.html>

Lessig, Lawrence. *The Future of Ideas: The Fate of the Commons in a Connected World*. New York: Random House. 2001

McCullaugh, Declan. “New Copyright Bill Headed to D.C.”. *WIRED News*. 7 Sep 2001. <http://www.wired.com/news/politics/0,1283,46655,00.html>

Moglen, Eben. Letter to Vidomi. *Vidomi Home Page*. <http://www.vidomi.com/fsfpop.php>

Mozilla Project, The. “Mozilla Public License, Version 1.1”. <http://www.mozilla.org/MPL/MPL-1.1.html>

Netscape Corporation. “NETSCAPE ANNOUNCES PLANS TO MAKE NEXT-GENERATION COMMUNICATOR SOURCE CODE AVAILABLE FREE ON THE NET”. *Netscape Home Page.* 22 Jan 1998. <http://wp.netscape.com/newsref/pr/newsrelease558.html netscape>

Open Source Initiative (OSI). “Approved Licenses”. *OSI Home Page.* <http://www.opensource.org/licenses/>

Powell, Dennis. "Judgment Day for the GPL?". *Linux Planet.* <http://www.linuxplanet.com/linuxplanet/reports/2000/1/>

Propaganda. *Propaganda Home Page.* <http://www.ibiblio.org/propaganda/>

Searls, Doc. “Surprise: Apple's New Browser Is a Sister to Konqueror”. *Linux Journal.* Jan 11 2003. <http://www.linuxjournal.com/article.php?sid=6565>

Stallman, Richard. “Free Software and Beyond.” Transcript. <http://mikro.org/Events/OS/ref-texte/stallman.html>

Stutz, Michael. “Design Science License”. *Michael Stutz Home Page.* 1999. <http://www.dsl.org/copyleft/dsl.txt>

Supreme Court of the United States. Eldred vs. Ashcroft. Jan 2003. <http://www.supremecourtus.gov/opinions/02pdf/01-618.pdf>

Tai, Andy. “Possible GPL violation by Vidomi”. *Advogato.* 23 May 2001. <http://www.advogato.org/article/282.html>

Thomas: Legislative Information on the Internet. Text of the Sonny Bono Copyright Term Extension Act. 27 Jan 1998. <http://www.loc.gov/copyright/legislation/s505.pdf>

Trolltech. “QT Free Licensing”. *TrollTech Home Page.* <http://www.trolltech.com/products/qt/freelicense.html qpl>

Vidomi. “Credit Where Credit's Due”. *Vidomi Home Page.* <http://www.vidomi.org/index.php?artid=6>