THE FREE FONT MOVEMENT

DAVID CROSSLAND

Dissertation submitted in partial fulfilment of the requirements for the
MA Typeface Design, University of Reading, 2008

Typeset on September 15, 2008 at 11:23, containing 13,739 words.

Dedicated to +Fravia

In the study of ideas, it is necessary to remember
that insistence on hard-headed clarity issues
from sentimental feeling, as it were a mist,
cloaking the perplexities of fact. Insistence on
clarity at all costs is based on sheer superstition
as to the mode in which human intelligence functions.
Our reasonings grasp at straws for premises
and float on gossamers for deductions.

— A. N. Whitehead, "Adventures in Ideas." (McLuhan, 1967)

ACKNOWLEDGMENTS

Many thanks to Ben Weiner, Nicolas Spalinger, Richard Stallman, Karl Berry, Benjamin Mako Hill, Jim Blandy, Raph Levien, Femke Pierre and Harrison at OSP, Bruce Perens, Pierre Marchand, Peter Linnel, Louis Dejardin, Miles Metcalfe, Ian Forrester, Bahi Para, Gustavo Ferreira, Andy Fitzsimon, Andy Ellis, Alexandre Prodoukine, Jon Philips, Ed Trager, Liam Quin, Rob Savoye, MJ Ray, Tom Lord, and Canny Kwok.

ABSTRACT

This dissertation examines the emerging free font movement, a small part of the larger free software and free culture movements.

Part A provides an overview of key concepts in the free software and culture movements. It starts by describing the hacker culture of the 1970s, the origins of Richard Stallman's GNU project, and his ethical basis for free software. Business and copyright practices are examined, and the cultural values of projects are described.

This is followed by an account of Stallman's theory of culture, and the Wikipedia and Creative Commons projects that are associated with this theory. Debates within the movement are explored, such as how Wikipedia develops, the role of non-commercial licensings, and the definition of 'free culture.'

Part B explores the implications of the principles of free culture for typeface design, attempting to answer whether typeface designs and fonts ought to be free. To do this it examines what typefaces are, who the users of typefaces are, and how type connects to Stallman's theory of culture.

It then discusses the relation of typefaces to font software, the different forms of digital type, and how font software connects to Stallman's theory. The legal status of typefaces and fonts is also considered.

Part C looks at what it means for fonts to be free, such as what font source code is. It examines how fonts are made free. The effects of various licensing practices and the ways font freedom is exercised are explored, such as collaborative community development processes.

A business model for sustainable commercial typeface design within the free culture movement is suggested, and a motivation for non-commercial typeface design activity is posed. Finally, areas for further research are suggested.

CONTENTS

1 INTRODUCTION

This dissertation examines the emerging free font movement. It starts by providing an overview of key concepts in the wider free software and free culture movements, by examining key figures and moments and explaining their motivations and significance. Having established the premises of the movement, it goes on to examine the implications of this to typeface design and font software. Finally, it discusses how the free font movement is practiced and areas of future development.

There are two chief intended audiences: type designers, and free culture advocates. Type design, like many other professions, has involved working intimately with computers for many years. Such professionals are often aware of the most publicised aspects of the free software and free culture movements—phrases such as 'open source,' popular applications like Mozilla Firefox, and the Wikipedia website. But they are less often familiar with its origins, goals and methods. It is hoped that a clear explanation of the movement will be of interest to type design professionals.

Free software and free culture advocates may also be interested in this author's account of the history of the movements, and the suggestions for progressing the movement in this crucial but obscure area of culture.

Part A

THE FREE CULTURE MOVEMENT

2 THE HACKER CULTURE

2.1 HACKERS, HEROES OF THE COMPUTER REVOLUTION

The free culture movement aims to reinforce a free society in the age of computer networks. It is intimately connected to the technology of the Internet that originated in the 'hacker' culture that formed around computers in the 1970s. Although today the term hacker is popularly understood as slang for computer criminals, it was initially a term of respect for computer programmers (Levy, 1984, p.432).

Levy's *"Hackers: Heroes of the computer revolution"* (1984) documents the origins of this culture in two groups. The earlier group formed around access to networked mainframe computers in science institutions such as at the Massachusetts Institute of Technology's Artificial Intelligence Laboratory (AI Lab). The later group was based around the earliest microcomputers that were used for computers games or simple data processing, and were not networked.

Levy codified an overall set of the culture's values that he labelled the 'hacker ethic':

> - Access to computers—and anything which might teach you something about the way the world works—should be unlimited and total.
> - All information should be free.[1]
> - Mistrust authority—promote decentralization.
> - Hackers should be judged by their hacking, not bogus criteria such as degrees, age, race, or position.
> - You can create art and beauty on a computer.
> - Computers can change your life for the better.
>
> (Levy, 1984, p.26–36)

The mainframe hackers' access to very expensive computers meant they were able to personally and directly advance the state of the art in software. But access was time-constrained and therefore they prized a flat distribution of power over the 'time-sharing' computers they worked with.

Lessig (2004, p.279) suggests that their "unlimited and total" access to software was a contextual characteristic of the computing industry at the time: Programs written for Data General machines would not work on IBM machines, so they were not motivated to exert any control. This context changed as the physical size and monetary cost of powerful computers fell, commoditising them, and their computational power rose geometrically.

1 The ambiguity in the word free is discussed in later sections.

Software developers became distinct from hardware developers, and many programmers left academia to start businesses. The most famous example of this trend today is Bill Gates, who dropped out of Harvard to found Microsoft in 1975, betting on the microcomputer trend (Wallace and Erickson, 1992). He went on to become the richest man in the world and his business model was simple: Each user must pay for permission to use each program on their computer.

This is achieved with 'End User licence Agreement' (EULA) contracts that require an agreement not to share, and with copyright law which also prohibits their unauthorised redistribution. His potential customers disagreed about this and actively redistributed the first Microsoft program, Altair BASIC.

Gates gained notoriety in the microcomputer hacker community when he responded by circulating a letter to popular computing journals throughout 1976, "An Open Letter to Hobbyists." He challenged software sharing as unfair and stated it was hobbling progress:

> Hardware must be paid for, but software is something to share. Who cares if the people who worked on it get paid? Is this fair? ... [You] prevent good software from being written. Who can afford to do professional work for nothing? What hobbyist can put 3-man years into programming, finding all bugs, documenting his product and distribute for free? ... Most directly, the thing you do is theft. (Levy, 1984, p.229)

2.2 DONALD KNUTH, AN EXEMPLARY HACKER

But the mainframe hackers viewed software as not as a product but as a service; being employed by institutions, they frequently did put years into writing, debugging, documenting and distributing a program without prohibiting sharing in order to charge per-user fees.

An example of this is Stanford mathematics professor, Donald Knuth. From 1968 he became famous, as computer scientists go, with his best-selling multi-volume textbook *"The Art of Computer Programming."* During the preparation of the third volume, he became frustrated with the poor quality of the phototypesetting technology that by 1977 had replaced the Monotype line caster metal typesetting technology used for his earlier volumes.

Digital typesetting systems were new in 1976, but Knuth realised that he himself could develop one in order to achieve the quality[2] he desired (Knuth, 1999, p.7). Writing such a large and sophisticated program took years of somewhat solitary work, but eventually Knuth successfully developed the 'TEX' typesetting system.[3] During its development he encountered the nascent culture of developer control,

2 "I had spent 15 years writing those books, but if they were going to look awful I didn't want to write any more." (Knuth, 1999, p.5)

3 TEX is used to typeset this dissertation. The system comprises several different components, including a programming language (`tex`), a program which interprets programs written in that language (TEX), and a format for representing pages in a device independent format which the TEXprogram outputs (DVI).

and rejected it: "A mathematical formula should never be 'owned' by anybody! Mathematics belongs to God" (Knuth, 1999, p.8).

When he completed the initial version in 1979, he published full details (Knuth, 1979), and when its was finished he published complete, documented source code (Knuth, 1986). Knuth's motivation was concern for progress, but in contrast to Gates, he thought that maintaining proprietary control would hinder progress: "The whole business of typesetting was being held back by proprietary interests ... pretty much there would be no progress" (Knuth, 1999; Levien, 2000, p.16).

Knuth valued the quality of his books above the monetary gain from having them published, and he did not assert control over others' use of his program because he was frustrated about the effect of propriety control on progress. Eric Hoffer (1951) identified *frustration* as the originating principle of mass movements and the hacker most famously frustrated by the rise of proprietary control was Richard Stallman.

2.3 RICHARD STALLMAN, THE LAST HACKER

In the early 1980s the microcomputer boom collapsed the once-vibrant AI Lab community. Levy (1984) described the migration of the AI Lab staff to two competing companies, Symbolics and Lisp Machines, as a personally hostile period that politicised Stallman against proprietary control of software; he became the last hacker left at MIT from the 1970s set. However the biography of Stallman by Williams (2002), which is strongly influenced by him, focuses on a less personal and more practical event as that which politicised him.

Stallman had been employed for 15 years at the AI Lab to make incremental improvements to the lab's software. During his final year at MIT in 1983, Xerox donated a prototype network laser printer to the lab, but with only binary copies of the printer's driver software. Programs made proprietary like this were unusual at the AI Lab at the time. Without the source code corresponding to the binary driver software, Stallman was unable to do his job.

This was frustrating because the printer was a prototype and often jammed. While visiting Carnegie Mellon university's computer lab, Stallman met the engineer who had written the software and requested a copy of the source code. His request was refused because the engineer had signed a 'Non Disclosure Agreement' (NDA) with Xerox, another way of maintaining proprietary control that was unusual at that time at the AI Lab (Williams, 2002, p.4–9).

The biography says this event caused Stallman to clarify his thoughts about the values of the hacker culture (Williams, 2002, p.10), but it was the retirement of the mainframe system he had worked on for

the previous 15 years that provided the final impetus to reassert these values with the 'GNU project' (Williams, 2002, p.91).[4]

4 Thus Stallman frames the free software movement as a *reform* movement: "We did not call our software 'free software,' because that term did not yet exist; but that is what it was." (Stallman, 1999b, p.53).

3 THE FREE SOFTWARE MOVEMENT

3.1 THE GNU PROJECT

On September 27th, 1983, Stallman posted a message to the `net.unix-wizards` Usenet newsgroup that proposed writing "GNU," a complete operating system and suite of applications with no proprietary components.[5] GNU would include existing programs that were unrestricted, such as TEX and the X window system,[6] and would create all the other necessary parts.

The announcement explained his motivation for such a large undertaking in terms of ethics:

> I consider that the golden rule requires that if I like a program I must share it with other people who like it. I cannot in good conscience sign a nondisclosure agreement or a software licence agreement.
>
> So that I can continue to use computers without violating my principles, I have decided to put together a sufficient body of free software so that I will be able to get along without any software that is not free. (Stallman, 1983)

The 'golden rule' referred to is the ethical principle of reciprocity, recognised by all major world religions and known in the Occident by the Biblical phrase "do to others what you would have them do to you" (Matthew 7:12).

This shows Stallman's clarification of the informal 'hacker ethic' into a formal rationale based on ethical principle. While the reciprocity principle also underlies Gates' assertion that sharing software is unfair, the other principle that motivates Stallman it that of a free society. He believes it is unjust for software developers to have total power over their users because as computers come to control more and more activities in society, non-free software creates a non-free society:

> Schools should teach their students to be citizens of a strong, capable, independent and free society. And this means teaching them to use free software, not proprietary software. (Rubini, 2007)

5 GNU is an acronym for "GNU's Not Unix," chosen because it explained it was modelled on AT&T's popular but operating system Unix, but was not in fact Unix, since Unix was propriety. First designed in 1969 (Raymond, 1996, p.460), Stallman chose Unix as a model because it was modular and could run on many heterogeneous systems, and users familiar with Unix would be able to easily switch to GNU.

6 The X Window System is a networked graphical user interface system originating from MIT in the early 1980s, and like TEX continues to be developed and used widely today.

Stallman's proposal solicited donations of hardware, money, software and work. All were forthcoming so he quit his job at MIT and the following year published the 'GNU Manifesto' (Stallman, 1984) and set up the 'Free Software Foundation' (FSF) to receive donations in a tax deductible way and administrate the project.

3.2 FREE SOFTWARE DEFINED

In the English language there is some ambiguity inherent in the term "free software," because 'free' can refer to freedom or to price. In the anglophone consumer culture of the late 20th century the latter meaning is the default, though it is unambiguous in other languages, such as "logiciel libre" in French.

Today the FSF website notes that in the initial announcement "several of the details about [the term] free software had not yet been clarified" (Stallman, 1983) and now provides the 'free software definition' that Stallman refers to as 'the four freedoms':

> - The freedom to run the program, for any purpose (freedom 0).
> - The freedom to study how the program works, and adapt it to your needs (freedom 1). Access to the source code is a precondition for this.
> - The freedom to redistribute copies so you can help your neighbor (freedom 2).
> - The freedom to improve the program, and release your improvements to the public, so that the whole community benefits (freedom 3). Access to the source code is a precondition for this.
>
> (Free Software Foundation, 2008a)

All computer users are able to directly exercise freedoms 0 and 2, but studying how a program works and adapting it implies that one must be a programmer to benefit from all of these freedoms. However there are many aspects of programs that are not programmatic, such as the text of the user interface. Free software user interfaces can be freely translated into any language, so monolingual minority language users can run the programs.

In this way all individuals do benefit indirectly when everyone in society has these freedoms. If a modification does require a programmer, non-programmers can hire any programmer they choose to exercise these freedoms on their behalf.

3.3 FREE SOFTWARE BUSINESSES

When software does not come with such freedoms, it is accurately referred to it as 'proprietary' or 'non-free' instead of 'commercial.' Yet free software is often developed on a commercial and for-profit basis (Friedman, 2005, p.97).

Free software is often mistakenly referred to as 'non-commercial' though. Since everyone is free to develop a free program in their leisure time, there is a 'long tail' power law distribution of commercial and hobbyist development, and the hobbyists can be more visible.

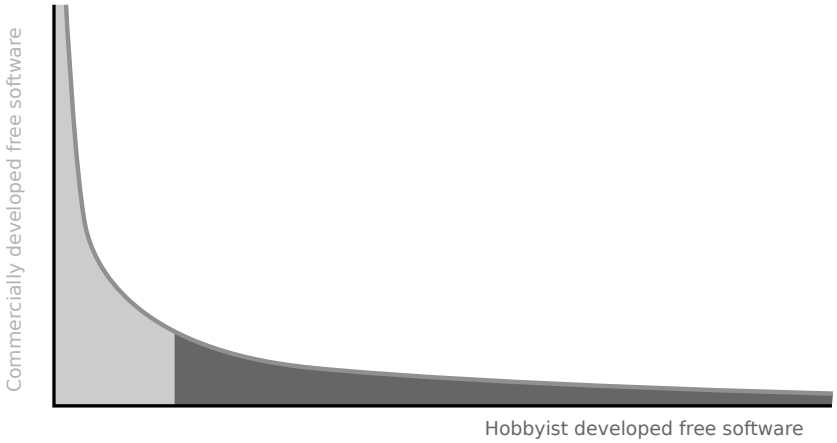


Figure 1: The long tail of commercial and hobbyist free software development

Additionally, freedom to redistribute a program has direct implications for the standard business model of software developers that Gates defended in 1976.

The GNU Manifesto included suggestions about how programmers could make a living while permitting unrestricted redistribution and improvement:

> There are plenty of ways that programmers could make a living without selling the right to use a program. This way is customary now because it brings programmers and businessmen the most money, not because it is the only way to make a living. It is easy to find other ways if you want to find them. Here are a number of examples.
>
> A manufacturer introducing a new computer will pay for the porting of operating systems onto the new hardware.
>
> The sale of teaching, hand-holding and maintenance services could also employ programmers.
>
> People with new ideas could distribute programs as freeware,[7] asking for donations from satisfied users, or selling hand-holding services. I have met people who are already working this way successfully.

7 The term 'freeware' has since become established as redistributable but non-modifiable software and is no longer synonymous with free software.

> Users with related needs can form users' groups, and pay dues. A group would contract with programming companies to write programs that the group's members would like to use. (Stallman, 1984, p.38)

The final item in that list can be seen as the principal activity of the FSF until the late 1990s. The Foundation received large donations annually, starting with $10,000 in 1987 (Goldstein et al., 1988, p.4), and $250,000 in 1989 (Arceneaux et al., 1988, p.5), and used these to funds to hire programmers.[8]

As the project progressed, contributors soon started developing GNU software on a for-profit basis. In 1987 the FSF began supporting this by maintaining a 'service directory' that listed people offering services, such as "answering questions for new users, customising programs, porting to new systems, or anything else" (Rubin, 1987).

In 1989 some of GNU contributors founded a start-up, Cygnus Support, "the first for-profit corporation that provides commercial support *only* for free software" (Kingdon et al., 1990, p.10). This was the maintenance business model Stallman suggested in 1986, founded because there were "so many requests for paid support" of the GNU project's software development tools (Hudson, 2008).

These tools were centered around the GNU C Compiler (GCC), a large and complex program that converts the source code that people write into binary code that computers run (Puzo, 1987, p.5).

In 1986 Stallman wrote the first complete version of GCC on his own, having never written a compiler before. It was considered excellent quality by other hackers, cementing his reputation in the hacker community as "the greatest hacker who ever lived" (Moody, 2001, p.30).

In 1987 Michael Tiemann extended the C compiler into a C++ compiler, while working at "the Microelectronics and Computer Technology Corporation (MCC), a consortium established in 1982 to do long range research for around 20 shareholder companies"—the same business model as the FSF (Goldstein et al., 1988, p.9).

Tiemann was particularly aware of the business potential of free software, stating that for many the GNU manifesto "read like a socialist polemic, but [he] saw something different ... a business plan in disguise" (Tiemann, 1999).

Cygnus took off when they began additionally pursuing the first business model Stallman outlined in 1986, porting GNU software to new hardware. Cygnus developed marketing material around the concept of free software to appeal to large corporate customers; the company slogan was "making free software affordable." As part of these efforts Cygnus published a 'Free Software Report' and the first issue explained free software in economic terms:

> Nobel laureate John Maynard Keynes proposed the theory that free markets were more efficient than artificial markets. The

8 The FSF's income in 2006 was $832,175 (Charity Navigator, 2008) and it no longer hires programmers because of the commercial free software development market.

> balance sheet of what is left of the Soviet Union tends to bear this out. The inherent economics of software are that software is expensive to produce and maintain, but cheap to duplicate. This is not well represented by traditional software offerings, where development costs are rarely cited, and duplication is severely restricted.
>
> By charging for production and maintenance and offering the freedom to duplicate, free software behaves like an ideal commodity in the free market. In contrast, proprietary software depends on a state-supported monopoly (copyright) to support prices, and thus behaves like a controlled commodity in an artificial market. (Thatcher, 1992, p.1)

Cygnus was successful at engaging lucrative corporate contracts and became the largest free software company of this period, easily comparable to most successful proprietary software businesses.

In 1998 it had a compound annual growth rate greater than 65% since 1992; had been listed on the San Jose Business Journal's Top 100 Fastest Growing Private Companies list three years in a row; ranked on the "Software 500" list of highest revenue software businesses in the world (Tiemann, 1999, p.71).

That year it had annual revenues over $20,000,000, employed more than 120 people, and was sold to Red Hat for $600,000,000 making all the early employees into millionaires (Gilmore, 2006).

Tiemann has stated that a key aspect of Cygnus' success was the innovative approach to licensing GNU software, 'copyleft' (Wayner, 2000, p.194).

3.4 GPL AND STRONG COPYLEFT LICENCES

Prior to the GNU project, software was typically made free by dedicating it to the public domain. But GNU software was instead distributed under a 'copyleft' copyright licence, the GNU General Public licence (GPL, Appendix B.1). The June 1988 newsletter explained what both were, and why they were used:

> [Public domain software] allows anyone to copyright and restrict its use against the author's wishes, thus denying others the right to access and freely redistribute it. This completely perverts the original intent.
>
> A copyleft is a legal instrument that makes everybody free to copy a program as long as the person getting the copy gets with it the freedom to distribute further copies, and the freedom to modify their copy (which means that they must get access to the source code). Typical software companies use copyrights to take away these freedoms; now we software sharers use copylefts to preserve these freedoms.
>
> The copyleft used by the GNU project is made from a combination of a copyright notice and the GNU General Public licence. The copyright notice is the usual kind. The General Public licence is a copying licence which basically says that you have

> the freedoms we want you to have and that you can't take these freedoms away from anyone else. (Tower et al., 1988, p.4)

When a software developer plans to include a new feature in their program, they are faced by a 'build or buy' scenario; they can either write the new feature internally, or they can integrate existing 'library' code from a third party.[9]

Licensing a proprietary library can be cheaper and faster in the short term, but cedes some control over a business to its suppliers which may be unacceptable. Over the long term suppliers can become less cost effective or even go out of business. Costs can be cut and development hastened while retaining full control by integrating free software.

But if the developer's business model depends on prohibiting sharing and further modification, the GPL blocks them. Although from a lay-person's perspective the developer's software is separate from the integrated component, under copyright law the whole combination is considered a 'derived work' of the GPL part.

Therefore either it must be licenced under the GPL, or it must not be distributed at all. Many programmers find this contentious (Kennington, 2006) and Microsoft Senior Vice President Craig Mundie has criticised it:

> This viral aspect of the GPL poses a threat to the intellectual property of any organization making use of it. It also fundamentally undermines the independent commercial software sector because it effectively makes it impossible to distribute software on a basis where recipients pay for the product rather than just the cost of distribution. (Mundie, 2001)

Mundie's comments contrast with the experience of Cygnus, who were fundamentally supported by the GPL: It meant that their customers would benefit from improvements commissioned by Cygnus' other customers, and that when their competitors won a contract, the client would publish the source code for their version when their product shipped, so that "all the new features and insights developed by competitors would flow directly back to Cygnus" (Wayner, 2000, p.194).

3.5 BSD AND NON-COPYLEFT LICENCES

The GPL acts as a lever to "ensure cooperation," something of paradox (Ray, 2008b), which leads to a philosophical contention: Copyleft is a restriction. Although the GPL is intended "to protect your rights

9 In this context a library is a program designed for use as a component in a larger program, although any program can be integrated into any other if its source code is available.

[and to] forbid anyone to deny you these rights or to ask you to surrender the rights" (Appendix B.1), some say that despite that intention the GPL is restrictive so it is not truly free. The figurehead for those with this view is the BSD project.

Programmers at The University of California Berkeley's Computer Systems Research Group had vigorously improved AT&T's Unix into the 'Berkeley Standard Distribution' (BSD), which included early Internet access software.

Stallman visited California to persuade the Berkeley programmers like Keith Bostic to make the non-AT&T parts free software,[10] so that they could be used in the GNU system (Williams, 2002, p.129).

In 1989 this happened and the programmers at Berkeley decided to rewrite all the AT&T parts to release their own free software Unix operating system.[11] However, they used the 'BSD licence,' (Appendix B.2) not the GPL.

The BSD licence permitted modification and redistribution in source code or binary form, and the only conditions were to give attribution without endorsement.[12] This meant that such software could be used as part of a proprietary program.

Therefore proprietary software developers prefer simple all permissive licences like the BSD licence. For example the X Window System developed at MIT in the early 1980s was always made available under the 'MIT X11' licence because it was funded by a consortium of proprietary Unix vendors.

By the late 1980s many programs for proprietary Unix systems were being released as free software, often related to Internet services: 'BIND' underpinned the domain names system that ties human address names like `www.gnu.org` to machine address numbers like `199.232.41.10`; 'Sendmail' underpinned the Internet's email system; and later 'Apache' would become the most popular web-server program.[13]

New 'scripting' programing language systems such as Perl and Python were released as free software, too. But all used their own non-copyleft licences, signalling their independence from the GNU project (Raymond, 2001, p.70).

10 "We'd go to dinner afterward and continue arguing about copyright over dinner." (Williams, 2002, p.129)

11 A complete free software Unix system was developed at Berkeley, but was not publicly available until the mid 1990s because of legal disputes with AT&T.

12 These conditions required any advertising for derived works to say "this product includes software developed by the University of California, Berkeley and its contributors." This meant that, just as the GPL blocked proprietary software developers from integrating GPL programs, the BSD licence blocked developers using the GPL from integrating those programs. Thus the MIT X11 licence, which is similar but without that clause, is used by many free software projects. In 1999 this was resolved when that clause was removed (Appendix B.2).

13 The Apache licence was also not compatible with the first two versions of the GPL. Despite being a non-copyleft licence, it contained legal requirements that responded to rising the threat to free software posed by software idea patent litigation. However the third version of the GPL has provisions for becoming compatible with more licences (Appendix B.5).

3.6 LGPL AND WEAK COPYLEFT LICENCES

In 1991 the second version of the GPL was published, followed by a 'Library GPL' (LGPL) (Appendix B.3; B.4). The second GPL contained only small adjustments,[14] but the LGPL was substantially different.

If a proprietary software developer wants to add a feature to their program, they may be unwilling to use a library licenced under the GPL. However, they might be willing to use a free software library if their parts of the combination remain under their own licence.

Libraries can be enhanced to better integrate with specific programs. The LGPL requires that if an improved library is distributed, developers must also make the source code for these improvements available to their users. It also requires developers to provide the means for users to run the program with their own modified versions of the library.

Thus the 'weak' copyleft of the LGPL is a tactical compromise to encourages proprietary software developers to contribute partially to the free software movement.[15]

At this point Stallman had succeeded at conserving the 1970s hacker culture as the 'free software movement,' a social movement to write free software larger than the GNU project and independent of it.

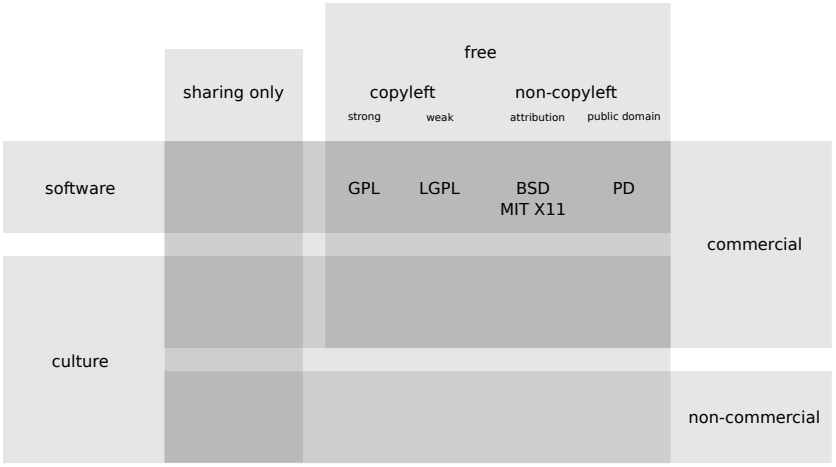


Figure 2: The spectrum of copyleft in major free software licences

14 The largest differences were the additions of new legal requirements that responded to rising the threat to free software posed by software idea patent litigation.

15 This tactical nature was highlighted in 1999 when the licence was renamed the 'Lesser GPL' because "it does Less to protect the user's freedom than the ordinary General Public licence" Free Software Foundation (1999).

4 THE OPEN SOURCE MOVEMENT

4.1 THE LINUX KERNEL

After quitting his job at MIT to start the GNU project, Stallman had earned a living by distributing GNU software and manuals; with GNU underway the FSF took over his distribution business and he did commercial development of GNU programs. GNU contributors saw this as the way they would earn a living, as Stallman could earn $100/hour (Lord, 2007).

In 1991 Stallman was awarded a MacArthur Fellowship "genius grant" income award that totalled $240,000 (Moody, 2001, p.28). The award was invested in mutual funds to give him a secure lifetime income so he could dedicate himself fully to the movement. It also enabled him to travel more to give lectures about free software around the world.

By 1991 the GNU project had developed working programs for all the necessary parts of an operating system, except a one—a 'kernel.'[16] But the GNU kernel, 'HURD,' suffered numerous problems and never became compelling enough to attract intensive development as other GNU programs had done (Appendix A.1).

In 1991 Linus Torvalds, a computer science student at the University of Helsinki, attended one of Stallman's lectures (Torvalds, 2001, p.58). That year he became frustrated that Unix was priced outside the reach of an undergraduate student, so in order to run Unix on his microcomputer, he started developing a kernel in his spare time[17]: 'Linux' (Williams, 2002, p.136).

Torvalds developed Linux with a simple design that was complemented by Torvald's unusual style of project management: He was willing to accept any 'patch'[18] emailed to him if it worked (Moody, 2001, p.80). This meant a loose knit community quickly formed around the project. Many patches were sent to him and there was unusually rapid progress.

The licence of the Linux kernel was not initially free software because it included an anti-commercial restriction, but Torvalds adopted the GPL in 1992. This meant a complete operating system that was totally free software was finally possible. With the commercial restriction removed, dozens of individuals in the Linux community combined the GNU system with the Linux kernel and other programs that would run atop the combination.

16 A kernel is a core operating system program that connects all other programs to the hardware, and is particularly complex.

17 "Linux is ... just a hobby, [not] big and professional like gnu" (Torvalds, 1991)

18 A patch is a text file that represents the differences between two version of the same source code file; ideal for contributing improvements.

4.2 GNU/LINUX DISTRIBUTIONS

A cottage industry formed around selling these complete systems, which the distributors called 'Linux distribution' (Moody, 2001, pp.88–100). This is synecdochal since Linux is an operating system kernel, not a complete operating system.

Stallman asked for the complete system to be called 'GNU/Linux' to give a fair representation of the systems origin, and call attention to the GNU project's goal of opposing proprietary software (Stallman, 2008). But this was generally ignored since the developers of most distributions *wanted* to include proprietary programs, not reject them.

With many Internet service programs already existing as free software, GNU/Linux quickly matured into a fully-featured Internet server system. This shifted the distribution cottage industry from selling copies of GNU/Linux for a small distribution fee to selling installation support and long term maintenance contracts. The largest company to emerge from this activity was Red Hat, which today has 2,200 employees worldwide, $400,000,000 annual revenue and $60,000,000 annual profit (Red Hat, 2007).

Ian Murdock was frustrated that the initial distributions did not reject proprietary software like the GNU project, and were not developed collaboratively at a fast pace like the Linux project.

In 1993 he started the Debian project to do so.[19] Collaboration was facilitated by Debian's 'package management' software,[20] where individual contributors took ownership of a particular program and created a package that would fit together cleanly with the rest.

Creating a wholly-free GNU/Linux distribution would finally reach Stallman's goal of a complete operating system with no proprietary programs, so the FSF funded Ian to work full time on Debian for 6 months, delegating the job of shipping GNU (Appendix A.1).

But the FSF and the early Debian developers fell out over a technical disagreement and Debian set up its own charitable foundation, 'Software in the Public Interest, Inc' (SPI).

Debian has since operated entirely independently of the FSF in a democratic manner (Moody, 2001, p.92). The processes for democratic collaboration at such a large scale were developed throughout the 1990s, and the principles were chartered in the 'Debian social contract' by Bruce Perens in 1997 SPI (2004). A key part of the social contract is the 'Debian free software guidelines,' which make explicit many of the implications of the FSF's free software definition.

Today Debian is the largest project within the free software movement, billing itself as 'the universal operating system.' Over 18,000 packages are maintained by over 1,000 voting members and these

19 Debian distributes proprietary software packages, but does not consider those packages part of Debian (Debian, 2004).

20 Package management software allows users to upgrade a system and make incremental updates to individual programs with ease.

packages that work in a uniform way from mobile phones to supercomputers (Debian, 2008).[21]

This size and reach is achieved with little money compared to corporate distributions. SPI (2008) reports Debian spent $194,495.81 and received $258,490.94 in 2007. This is because Debian is created by volunteers who earn a living disconnected from Debian, ranging from self-employed consultancy[22] to employment at large technology corporations like Pixar and Hewlett-Packard.

The most visible and popular distribution with individuals today is Ubuntu. In 2004 dot-com millionaire Mark Shuttleworth founded Canonical and the Ubuntu project to make a derivative of Debian focused only on common laptop, desktop and server hardware.

Canonical sells support services for Ubuntu to individuals and corporations, and Ubuntu is released every 6 months and focuses on providing a operating system competitive with the latest versions of Windows and Mac OS X thanks to the graphical desktop systems, KDE and GNOME (Canonical, 2008; Hill, Bacon, Burger, Jesse and Krstic, 2006).

4.3 KDE AND GNOME

In contrast to its server capabilities, GNU/Linux desktop software took a long time to mature. The GNU project had always planned to develop a full graphical desktop environment akin to Windows 95 and Mac OS. But like the HURD, efforts in 1990 and 1994 had stalled (Stallman, 2001).

In 1995 the first large GNU program with a graphical user interface (GUI), the GNU Image Manipulation Program (GIMP), was developed by two Berkeley students in their spare time. Spencer Kimball and Peter Mattis wanted to replicate Adobe Photoshop, and began with a proprietary 'toolkit'[23] but quickly developed their own 'GIMP ToolKit' (GTK), released under the LGPL.

In 1996 a student at the German university of Tübingen, Matthias Ettrich, developed a word processor based on TEX as a course project. For this he used the 'Qt' toolkit from the start-up Trolltech. Despite that the source code was published, it was proprietary software, but Ettrich used it because it was technically advanced and allowed him to develop his program quickly.

This success motivated him to initiate the development of a free software desktop environment based on Qt, 'KDE.' His Usenet announcement attracted contributors and they reached a consensus that they would prefer Qt's practical benefits over less advanced but free alternatives, but for their own programs they would adopt the GPL (Moody, 2001, p262).

21 While not referring to Debian specifically, a list of supercomputer operating systems gives GNU/Linux 5.6% market share in 2000 and 84.6% in 2007 (Top500.org, 2007).

22 Phil Hands is been a long-time Debian consultant whose `http://www.hands.com` homepage states his daily rate is "£1075 + VAT + expenses."

23 A toolkit is a library for constructing GUIs.

As the KDE project became popular, the GNU project contributors objected to KDE's dependency on Qt because it was proprietary, and responded in three ways: Stallman approached Trolltech and suggested that a free software business model could work for them;[24] a direct replacement for Qt was initiated; and a third GNU desktop environment based on GTK was developed: GNOME.[25]

4.4 OPEN SOURCE

The KDE project's decision to accept a proprietary dependency, most distributions' inclusion of proprietary software, and the many free software programs released under non-copyleft licences, are all indicative of split in the free software movement: Many do not agree with the FSF that proprietary software ought to be rejected and opposed.

In 1997 Eric Raymond published a paper "The cathedral and the bazaar" and presented it at one of the first large free software conferences, in Germany. Brooks (1978) famously wrote that large programming projects achieve the highest quality and are done fastest in small teams. He stated this was because complex programs require planning and coordinating programmers takes time. Raymond noted the way that the Linux kernel was developed by a large team and yet was a complex, high quality program that was progressing rapidly, discrediting Brooks.

Tim O'Reilly is the founder of technical publisher O'Reilly Media. The company had been successful in the 1980s publishing third party documentation for Unix programs, such as the best selling O'Reilly manual for the Perl programming language. He invited Raymond to present his paper at the first O'Reilly Perl conference, and shortly after convened a 'summit' of influential hackers, including Raymond and the primary developers of the popular free Internet server programs—but excluding Stallman (Moody, 2001, p.166). The summit sought to coin an alternative term for free software: 'Open source."

A charitable foundation, the 'Open Source Initiative,' was formed, and a trademark registration application was filed. In practice both terms describe exactly the same kind of software; the term was defined by Perens for the Open Source Initiative's 'open source definition' by adapting the 'Debian free software guidelines' without making any major changes (Moody, 2001, p.168) (Perens, 1999).

The new term was presented to the free software community as a necessary clarification of the ambiguity in the word 'free'[26] (Raymond,

24 In 2000 Qt was released under the GPL for GNU/Linux (Stallman, 2002a, p.28) and today is also available for Windows and Mac OS X. GTK is available under the LGPL for GNU/Linux and Windows and while the Mac OS X version is under development as a prototype.

25 The GNOME project established its own charitable foundation, The GNOME Foundation, and operates somewhat separately from the GNU project.

26 This was immediately criticised as vulnerable to co-option and countered with the suggested term 'software freedom' (Tower, 1998).

1998). But it was also intended for those who do not view proprietary software as unjust "to distance themselves from the FSF" and its political motivation (Torvalds, 2006). This meant free software could be presented to the corporate business community without sociopolitical dissent over proprietary control.[27]

Shortly after the summit, the proprietary web browser developer Netscape made a radical announcement: To compete with Microsoft's 'Internet Explorer' browser, it would publish the source code of its browser as 'open source.' This eventually resulted the most popular alternative to Internet Explorer, Mozilla Firefox (Refsnes Data, 2008).

With Netscape in the press and O'Reilly's influential support, the term soon became established and the 'open source movement' emerged as distinct from the free software movement. The difference is in their motivations, since they work towards the same aim. The open source movement is motivated by Raymond's ideas that through peer review, free software inevitably leads to reliable, secure, fast, powerful and convenient software.

27 "It was about giving businesses a better story to tell their customers than 'We faithfully contribute all our patches back to some guys who are out to smash proprietary software."' (Lord, 2007)

5 THE FREE CULTURE MOVEMENT

5.1 FREE SOFTWARE MANUALS

The GNU project had from the start been writing documentation for its programs and making it available under the GPL or a simple copyleft licence (Stallman, 1983, 1999b). O'Reilly benefited from establishing the open source movement because its rhetoric avoids the implication that if a program is changed, the manual accompanying it ought to take into account those changes.

In 1996 Stallman had published an article promoting the writing of free manuals, and while he did not name O'Reilly Media, he did mentioned proprietary Perl manuals as his primary example (Stallman, 1996). In 2000, he continued this effort to encourage the publication of free software manuals by developing a copyleft licence specifically for commercial manual publishers (Stallman, 2000b): The 'GNU Free Documentation licence' (GFDL) (Free Software Foundation, 2000).

Since the GFDL accommodated publishers, its terms are controversial in the free software community. The primary criticism is that it allows for 'invariant' sections which cannot be modified or removed (Nerode, 2003).[28]

Stallman's rationale for this was that works "should be distinguished not by media, but by the way the works contribute to society" (deVilla, 2007). GNU manuals contained functional information about programs that ought to be modifiable, and also political essays promoting the GNU project's goals. In Stallman's view it would not contribute to society if his essays were modified because that would misrepresent him.

5.2 STALLMAN'S THREE CATEGORIES

At this time Stallman also began to give speeches about how the principles of software freedom applied to other kinds of works. Since starting the GNU project, he had often been asked how software freedom might apply to other areas, such as hardware. Since physical objects cannot be copied without factories, and since modification of your own property is limited only by feasibility, Stallman states that the principles of free software do not apply to physical objects.

However, many kinds of works are not physical, and subject to copyright. He proposed a broad theory of compromises between the

28 In 2002 version 1.2 of the licence was published in response to some of this feedback (Free Software Foundation, 2002) and in 2006 the Debian project voted that GFDL works are only free if they do not have invariant sections (Debian, 2006).

'no rights reserved' public domain and the 'all rights reserved' copyright default, starting the 'free culture movement' (Stallman, 2000a).

Copyright law already distinguishes between different kinds of works and arranges different copyright bargains for each kind. Stallman characterises copyrighted works into three categories, based on a work's purpose for its users: to function, to witness, and to entertain. These categories can be explained by example:

Works that are practical and functional: Software programs and corresponding manuals, culinary recipes, reference works such as encyclopedias, dictionaries, textbooks and species taxonomies, geographic maps, educational material, designs of equipment and buildings.

Works that witness the thoughts of certain parties: Essays, scientific papers, political manifestos, personal diaries and memoirs.

Works that are artistic, aesthetic, or entertainment: Novels, theatrical scripts, music, cinema, television drama, paintings,

Stallman believes that the general public should always be able to non-commercially redistribute works verbatim (Appendix A.4); that functional works should always be commercially redistributable and modifiable; that witnessing works should not be modifiable; and that artworks should only be restricted in their commercial use, redistribution and modification for a short period of 10 years (Stallman, 2002b; deVilla, 2007).

5.3 WIKIPEDIA

In 1999 Stallman begun promoting the idea of a free encyclopedia that would be written by school teachers (Severance, 1999; Stallman, 1999a). Two people contacted Stallman for advice on their projects to do this, GNUpedia and Nupedia, and the projects eventually merged. Nupedia was the larger effort and had an expert review panel process, but was proceeding slowly with less than 24 articles in the first year. To hasten development, the chief editor Larry Sanger adopted a new kind collaborative authoring software, a 'wiki' (Sanger, 2005).

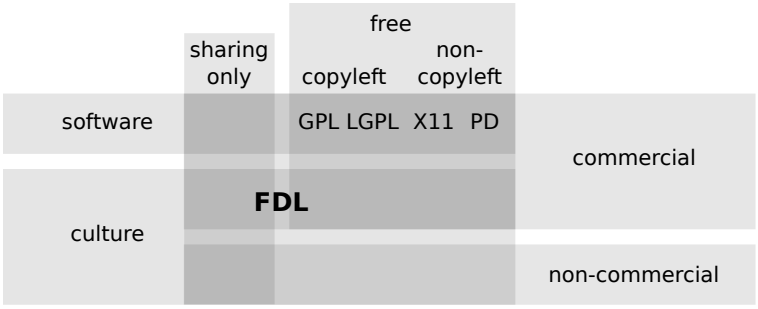


Figure 3: The GNU Free Documentation licence in relation to free software licences.

A wiki is a website revision control system for radical collaboration: They allow all visitors to edit the text of pages, even anonymously. This anarchy allows fast progress because it allows visitors to immediately correct errors of omission or fact. It is not degenerative because all edits are saved incrementally, so vandalism can be instantly reverted.

The wiki part of Nupedia was licenced under the GFDL and in January 2001 it was split from Nupedia and renamed Wikipedia. Wikipedia grew quickly, with over 13,000 English articles within one year (Sanger, 2005). Today it has more than 2,534,000 English articles and millions more across 322 languages (Wikimedia Foundation, 2008) and is the 8th most popular website globally (Alexa Internet, 2008); Wikipedia is the largest and most visible project within the free culture movement.

Sanger's (2005) account of the origins attempts to explain "why Wikipedia started working." However Clay Shirky (2008) has explained the success of Internet-based collaboration overall in terms of radically collapsed 'transaction costs.' Shirky states that while Coase (1937) explained how effective management lowered the transaction costs of doing business to create successful firms, management itself is a transaction cost. Since computer networks reduce transaction costs, the cost of management can in some situations outweigh its benefit:

> Loosely coordinated groups can now achieve things that were previously out of reach for any other organizational structure, because they lay under the Coasean floor ... Prior to the current era, the alternative to institutional action was usually no action. Social [network] tools provide a third alternative: Action by loosely structured groups, operating without managerial direction and outside the profit motive. (Shirky, 2008)

Like Debian, despite its size Wikipedia is operated with relatively little money; the Wikimedia Foundation (2006) reported its expenses were $56,666 in 2004, $201,418 in 2005 and $716,132 in 2006, of which $107,122 were salaries and wages that year.

5.4 CREATIVE COMMONS

In 2002 a law professor at Stanford, Lawrence Lessig, tried to challenge the US Copyright Term Extension Act of 1998 in the Supreme Court. The amendment retroactively extended copyrights by 20 years, ensuring that Micky Mouse did not lapse into the public domain—substantially reducing it (Levy, 2002). Lessig lost the case, but the William and Flora Hewlett Foundation gave Lessig $1,000,000 to launch Creative Commons (CC) (Lessig, 2008).

Creative Commons aims to bridge the public domain by providing a range of general purpose copyright licences for all kinds of works that authors could easily apply. Today there are six licences

that all require attribution and permit worldwide non-commercial verbatim redistribution, but vary the terms of modification:

- Attribution (CC-BY) permits any modification
- Attribution–NonCommercial (CC-BY-NC) permits any modification, non-commercially
- Attribution–ShareAlike (CC-BY-SA) permits any modification with a weak copyleft akin to the LGPL
- Attribution–NonCommercial–ShareAlike (CC-BY-NC-SA) permits any modification with a weak copyleft akin to the LGPL but all use must be non-commercially
- Attribution–No Derivations (CC-BY-ND) permits no modification
- Attribution–NonCommercial–No Derivations (CC-BY-NC-ND) permits no modification and verbatim redistribution can only be done non-commercially
  (Creative Commons, 2008)

These effectively implemented Stallman's theoretical framework, and he initially endorsed the project; the licences soon became massively popular and by 2007 tens of millions of works had been published with them (Cheliotis, 2007). However, the project also came to be criticised.

5.5 FREE CULTURE DEFINED

In 2004 Lessig published a book titled "Free Culture" (Lessig, 2004). The book examined the way new media technologies disrupted existing media businesses historically, and compared that to the contemporary situation where existing businesses seek to disrupt computer network technology. But it left the precise meaning of the term 'free culture' undefined.

The widest possible definition appeared to encompass anything permitting non-commercial redistribution, since the Creative Commons licences had that in common. But even this came into question when Creative Commons began promoting additional licences that did not permit non-commercial verbatim sharing worldwide in 2004 (Haughey, 2006a). Stallman found this unacceptable and withdrew his support until they were retired (Stallman, 2005). He also criticised the way the branding strategy promoted the overall brand ambiguously.

The non-commercial licences are the most popular, but also the most controversial for the commercial free software community.[29]

29 The Debian project also considered many small terms of the licences problematic (Haughey, 2006b) but these were resolved with the current 3.0 release of licences.

Myers (2008a,b,c,d) suggests copyleft mitigates exploitation and non-commercial restrictions are too limiting; Ray (2008a) said they create a "creative flowerbed" rather than a commons. But Lessig (2006) defended them, stating that allowing people to choose terms that they believed were best without being prescriptive was crucial.

Benjamin Mako Hill (2005) criticised the lack of a definition[30] and in 2006 co-initiated the development of the 'Definition of Free Cultural Works' (DFCW). This defined the label "free culture" in terms equal to the FSF free software definition, generalised away from software and with more detail akin to the Debian free software guidelines (Hill, Möller et al., 2006).

Creative Commons responded positively to all these criticisms. It developed licence-specific branding, retired the non-redistribution licences, and endorsed the DFCW. Thus today it is clear that only the CC-BY and BY-SA licences contribute to free culture (Linksvayer, 2008).

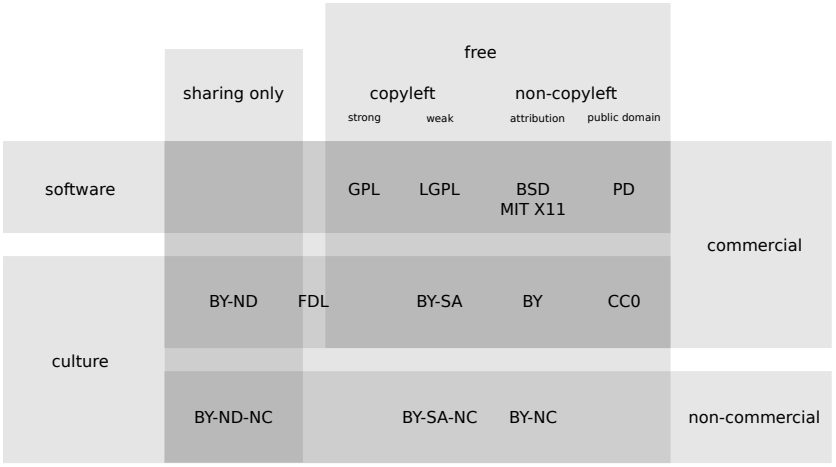


Figure 4: The relation of free software and Creative Commons licences.

30 "CC sets no defined limits and promises no freedoms, no rights, and no fixed qualities. Free software's success is built upon an ethical position. CC sets no such standard." (Hill, 2005)

Part B

IMPLICATIONS FOR TYPEFACE DESIGN

6 SHOULD TYPEFACE DESIGNS BE FREE?

6.1 WHAT ARE TYPEFACES?

To explore the implications of free culture for typeface design, what a typeface design is must be established. The etymology of the word 'typeface' comes from the 'Gutenberg' printing technology of medieval Europe, where a 'type' was a small long metal cuboid with the shape of a letter in relief on one 'face' of the cuboid. These types were arranged and bound into blocks, ink spread on the typefaces, and the block pressed onto paper to create a print.

'Calligraphy' is stylised hand-writing that harmoniously relates the shapes of letters to make them look like they belong together. The shapes of the letters initially mimicked the calligraphy of the time and were created through a process known as 'punch cutting' (Smeijers, 1996). As printing technology developed, the letter shapes on type faces diverged from calligraphic shapes, although they tended to retain elemental forms derived from the structural characteristics of human hand-writing such as contrast (Noordzij, 2005).

Printing technology has developed using a variety of media and processes, so the precise definition of a typeface can be unclear. In the community surrounding Knuth in the early period of digital typography, Richard Southall (1985, p.5–9) provided a precise set of definitions.

A 'character' is the abstract essence of a letter, separated from its visual shape. A 'glyph' is the inverse, a shape of a letter separated from its abstraction; the Latin character 'a' has two glyphs, 'a' and 'A.' A script is "a set of characters used to write one or more languages" and a typeface as "a set of distinctive, visually related shapes that represent some or all of the characters of a script."

These sets of shapes are abstracted from any particular medium; the same typeface can be expressed in many different kinds of media: metal, wood, or digitally. Thus Southall distinguishes a font from a typeface; a font is "a set of character shape specifications, on one or more image carriers, that corresponds to the character shapes of a particular typeface." 'Image carriers' are "the means by which the shapes ... are specified."[31] That is, a font is the concrete implementation of a typeface in a particular technology. Since typefaces and fonts are theoretically discrete, fonts are considered separately in the next chapter.

31 Metal types are 'specified' in a direct physical way with punches, but since Linn Boyd Benton's pantographic punch-cutting machine, patented in 1885, specifications have existed as drawings that are mechanically processed (Kinross, 2004, p.41)

6.2 STALLMAN'S THREE CATEGORIES

Stallman proposes three categories of copyrighted works that are distinguished by their qualitative contribution to society. In which category do typefaces belong?

Is a typeface a kind of witness statement? Existing historic designs are often revived. A typeface designer may develop a typeface with the intention that it witness their thoughts about another design (private communication with professional typeface designer Erik van Blokland, 2007).

Is a typeface a kind of artwork? A typeface designer may create a design based on intrinsic artistic motivations, and a typeface can be appreciated for its aesthetic beauty. Charles Bigelow has said, "typefaces are part of fashion" (Rubinstein, 1988, p.25).

Is a typeface a kind of functional work? Design is distinguished from engineering and art by a focus on users' needs rather than technological implementation or personal expression. A typeface designer may develop a design to meet the needs of users in a specific scenario, such as road signage.

Thus it is difficult to categorise typefaces based on typeface designers' intended contribution to society. However Stallman recommends evaluating a work based on how it is used in society instead, giving users primacy over developers.

While this is objectionable to typeface designers, Fred Smeijers (2003, p.25) echoed the primacy of users over typeface designers in judging a typeface aesthetically: "A type designer cannot escape this responsibility of judgement … In the end, people—the society—either accept it or they don't."

6.3 READERS ARE THE PRIMARY USERS

There are three kinds of typeface users, whose roles overlap: typeface designers, typographers and readers.

Readers are the 'end users' of type, and for them the purpose of a typeface is primarily to enable their reading of a text; Stanley Morison (1951, p.5) said "typography is the efficient means to an essentially utilitarian and only accidentally aesthetic end, for enjoyment of patterns is rarely the reader's chief aim."

That is, the primary use of a typeface is that it represents the characters of a script; readers must be able to read with it. A typeface that did not enable reading would be a paradox, by definition not a typeface. Reading also involves non-lexical activity, such as parsing the visual hierarchy of a text and distinguishing its parts (Unger, 2006).

Readers also have secondary purposes for type, inferring emotional values visually. This aspect of a typeface is akin to artwork and entertainment, and happens somewhat subliminally. For example, readers recognise identity through type, distinguishing company brands. If a typeface represents the opinion of a typeface designer on

another design or was created as art, those may be visible aspects for some readers but in such cases those aspects are tertiary.

6.4 TEXT AND DISPLAY FACES

These two kinds of purposes are reflected in typeface classification systems that distinguish between 'display' and 'text' typefaces (Dixon, 2002). Text typefaces are used to set paragraphs of text with complex lexical structures. Display typefaces are used for short amounts of text with simple lexical structures. For short texts such as advertisements, the secondary aspects of a typeface become primary for readers.

(a) Darkgarden (b) Palatino

Figure 5: Comparison of display and text typefaces.

Stallman has suggested that this could form the basis of a compromise between the free culture movement and the traditionally proprietary typeface design culture; text types must be considered functional since they are primarily about reading, but display types may legitimately be considered non-functional if the emotion conveyed in their visual forms is primary and their readability secondary (personal communication, 2007).

However Stallman is not a specialist in this field, and Smeijers (2003, p.23) has said that "often now the line between artistic practice and design is difficult to draw." Typeface classification expert Catherine Dixon has also questioned the distinction:

> Distinctions between text and display are now increasingly irrelevant, with the greater subtlety that has been introduced into sans serifs and slab serif designs leading to a wider application of such types for text purposes. (Dixon, 2002)

If no clear distinction can be made between the two, Stallman says that since the functional aspect of a typeface is inherent, all typefaces should be considered functional. Therefore all typefaces ought to be free (personal communication, 2008).

6.5 OTHER USERS OF TYPEFACES

However, even if a typeface is in the public domain, readers cannot directly exercise that freedom because printed type is a kind of physical object that cannot be modified. Instead typographers act on behalf

of readers to select and adjust type in anticipation of readers' needs and in response to their commentary.

In turn, typeface designers act on behalf of typographers to develop typefaces in anticipation and in response to typographers. Therefore the free status of typefaces is an indirect concern of readers, but a direct concern of typographers and typeface designers.

All typesetting technologies allow typographers to adjust some aspects of type, such as increasing the inter-letter spacing, but physical limits gave typeface designers full control of typefaces.

With the introduction of photo-typesetting in the 1970s, modifying glyph shapes with coarse optical distortions became possible, such as condensing them horizontally. But physical limits still blocked typographers from easily making fine changes to a typeface.

This changed when desktop publishing arrived in 1985 with the combination of the Apple Macintosh microcomputer, Apple LaserWriter printer, Adobe PostScript page description system, and Aldus PageMaker typesetting program (Kinross, 2004, p.169). Digital type removed all physical limitations so that all typographers could potentially take on the role of typeface designers.

Desktop publishing also blurred the division between authors and typographers, since all computer users with access to laser printers and print shops became de facto typographers. Thus since the 1980s technological advances have brought the issue of freedom in typefaces to the general public, not as readers but as de facto typographers.

With the shift from print to screen-based reading, it is possible for readers to become their own typographers and adjust the typographic layouts—and even the typefaces—they read with. The 'Cascading Style Sheets' (CSS) technology used to specify the design of web pages explicitly rejected that "the author ultimately had to be in charge of deciding how documents were presented ... The user, whose eyes and ears ultimately have to decode the presentation, should be given the last word" (Lie and Bos, 1999).[32]

For these reasons, typeface designs ought to be free as part of a free culture in a free society. The distinction between typefaces and fonts can be seen as a theoretical one; for practical purposes they are the same now that digital typography has eclipsed all other technologies in the printing industry.

32 The issue of 'web fonts' is very much a contemporary development, and as such is outside the scope of this dissertation.

7 SHOULD FONT SOFTWARE BE FREE?

7.1 FONTS AS PROGRAMS

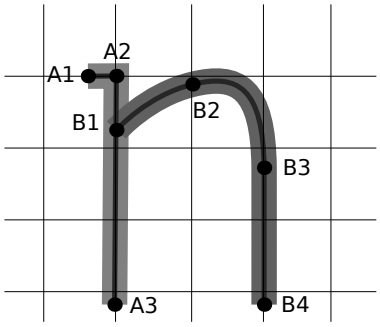
Fonts are generally accepted as a kind of software; but 'software' typically refers to programs, and fonts are not obviously programs, but more akin to data files like illustrations, documents or music files.

Typefaces can be specified digitally as either data, or as programs, or as both. The extreme cases are straightforward.

A program can describe the way a simple glyph is drawn from point to point with a pen in a predefined coded *language*. This is the basis of the METAFONT program that Knuth developed as part of the TEX typesetting system (figure 6a). These symbolic instructions are 'intelligent,' in that they describe *how* to draw the glyph shape, instead of describing *what* the glyph shape is. This means producing a range of variants of a typeface design, a 'typeface family,' can be done in a systematic and convenient way.

```
 1 (glyph "n"
 2  (point A1 ( 0.6, 3.0))
 3  (point A2 ( 1.0, 3.0))
 4  (point A3 ( 1.0,-0.2))
 5  (point B1 ( 1.0, 2.2))
 6  (point B2 ( 2.0, 2.9))
 7  (point B3 ( 3.0, 2.8))
 8  (point B4 ( 3.0,-0.2))
 9  (pickup square-pen)
10  (pen-down A1)
11  (pen-stroke A2 A3)
12  (pen-up B1)
13  (pen-stroke B2 B3 B4)
14  (label points)
15 )
```

(a) Sourcecode



(b) Output

Figure 6: A program describing a glyph and its output.

7.2 FONTS AS DATA

The simplest way of describing a glyph shape is with a table of numeric 'bitmap'[33] data in a predefined *format* (figure 7a) that correspond directly to mosaic dots of an output device—either high reso-

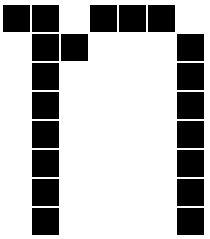
33 'Bitmap' literally means a map of bits; bits are binary digits; 'pixel' is a truncation of 'pixel element' and expands the idea with detailed colour information for each element to describe photographs.

lution printers with 300 dots per inch (DPI), or screens with less than 100 DPI (figure 7b).

In order to be rendered visible, programmatic descriptions of shapes are put through a 'rasterising process' that convert them into bitmap descriptions of shapes. In the early period of digital typography this was a slow process, so bitmap fonts were developed directly as the basis of the earliest 'visual display unit' (VDU) computer screen interfaces (Karow, 1987, p.71).

```
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 1 1 0 1 1 1 0 0 0 0 0 0
0 0 0 0 1 1 0 0 0 1 0 0 0 0 0
0 0 0 0 1 0 0 0 0 1 0 0 0 0 0
0 0 0 0 1 0 0 0 0 1 0 0 0 0 0
0 0 0 0 1 0 0 0 0 1 0 0 0 0 0
0 0 0 0 1 0 0 0 0 1 0 0 0 0 0
0 0 0 0 1 0 0 0 0 1 0 0 0 0 0
0 0 0 0 1 0 0 0 0 1 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

(a) Bitmap data



(b) Bitmap rendering

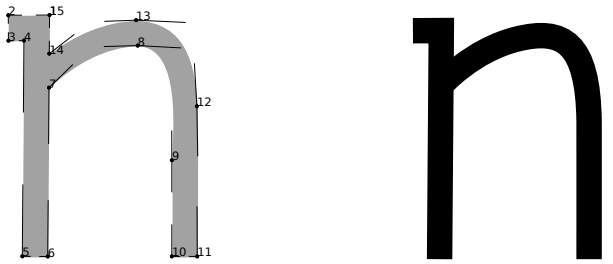
Figure 7: Data describing a glyph and its output.

Vector graphics are an intermediate way of representing a glyph, but whether they are programs or data can seem ambiguous. The program in figure 6a describes the skeletal path of a pen, which the nib shape fleshes out as it passes along that path (figure 7b).

While this is ideally suited for describing type forms that are strongly calligraphic, an abstract lexical program has an indirect connection to the concrete graphical shapes it outputs. This means it can be hard to create shapes as intended, especially less calligraphic ones (Southall, 1985, p.31).

Instead, the path of the outline can be described directly, which is much more simple conceptually and technically. This loses the 'intelligence' of the program in figure 6a, because it uses a pen program to describe only *what* the shape is, not *how* it is drawn. Yet such simple drawing programs are advantageous because direct control of the shape can happen through interactive GUIs where font developers set the points along the path in an entirely visual process (figure 8a).

Large typeface families can be developed relatively conveniently with interpolation techniques. Since users interact with vector graphics in these ways without seeing the programmatic representation of the outlines that underlies their interactive representation on screen, they appear to be data instead of programs. Indeed, they are referred to as being in *formats* instead of *languages.* But technically, they are still programs that describe shapes.



(a) Source (b) Output

Figure 8: A vector program describing a glyph outline, and its output.

7.3 INVISIBLE ASPECTS

A typeface design is not only the black shapes of letters though, and the inter-letter spacing is intimately connected to the letter shapes because "a free standing letter on a small rectangle is a rarity ... maintaining the equilibrium in the white shapes makes all the difference" (Noordzij, 2005, p.14). 'Metrics' are the glyph widths or 'bounding boxes' that form general inter-glyph spacing values, and 'kerns' are their exceptions for specific glyph combinations. These values are simply numerical data.

A font places these shapes and spacings in a technological context. During the rasterising process of display, the typeface may not fit the bitmap grid of an output device crisply.

The Adobe PostScript 'Type 1' font format introduced 'hinting instructions' which were simple because they were intended for laser printers like the 1985 Apple LaserWriter (Haralambous, 2007, p.507). Their simplicity means they are as amenable to being developed visually and without appearing as programs, just as vector outlines can seen as data.

But it also means they are unsuitable for instructing coarse grained computer screens. A complex hinting system based on a full 'Turing complete' programming language (Haralambous, 2007, p.518) was the key patented innovation of the Apple–Microsoft 'TrueType' font format released in 1991 (Kinross, 2004, p.170).

This means TrueType hints can have unintended effects—the 'bugs' inherent in all programs—but when implemented correctly do improve legibility (Sheedy et al., 2008). This is especially important for type at small sizes on low resolution cathode ray tube displays (CRTs).

However as those screens have been replaced by liquid crystal displays (LCDs). These have increased real resolutions of computer displays slightly, but 'sub pixel rendering' techniques have made a dramatic improvement in the effective resolution of such screens through software.

The different approaches to this in Apple and Microsoft's rendering systems has been controversial: Microsoft ClearType is "crisp" but slightly distorts the glyph shapes, while Apple Mac OS X auto-hinting rendering ignores hinting[34] and does not distort glyph shapes, but is "blurry" (Spolsky, 2007). The free rendering software 'Freetype' implements both approaches, although the hinting functionality is only legally used in countries where Apple's patent is not valid (Haralambous, 2007, p.546).

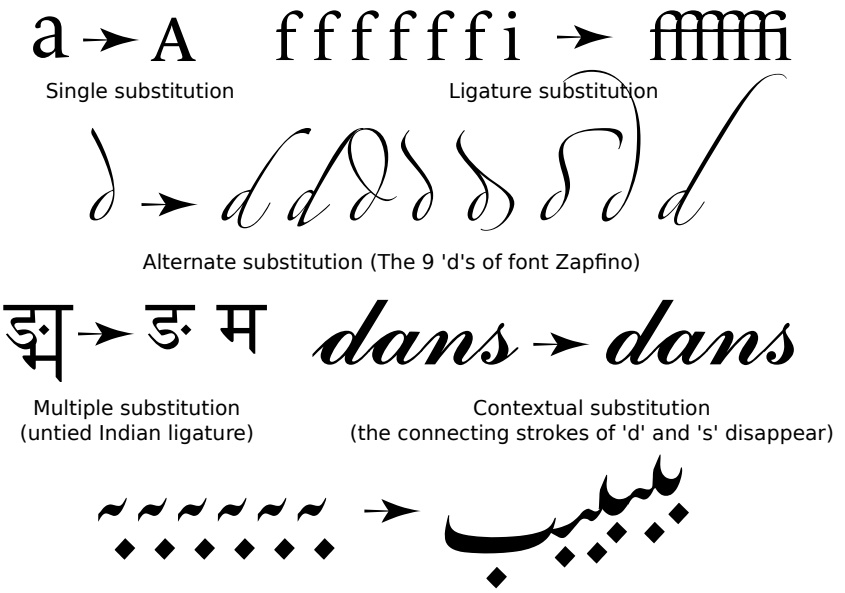


Figure 9: OpenType glyph substitutions, reproduced from Haralambous (2007, p.551).

Contemporary fonts also include additional logic to describe the specific behavior of a writing system for a typeface. This can involve altering glyph positions or substituting glyphs in appropriate character contexts; this is essential for many non-Latin scripts such as Urdu (figure 9).

The most popular 'smart font' format is 'OpenType,' which supports most major languages. It was developed by Microsoft and Adobe and in 2007 became an ISO standard (ISO, 2007) but is partially encumbered by Apple's hinting patents as it can envelop both Type 1 and TrueType fonts' outlines and hints.

OpenType implements complex layout logic through "lookup tables" and despite not being described in a Turing complete programming language, these are programmatic instructions. The more ad-

34 That Apple systems ignore hinting is ironic because Apple is the patent holder for TrueType hinting.

vanced SIL Graphite font format uses Graphite Description Language, "a rule-based programming language" (Correll, 2003).

Although fonts can appear like data both in their parts and overall, they are in fact an unusual kind of program. To the extent a font can be seen as data, this data generates a typeface for reading, itself a functional work. Therefore fonts ought to be free as part of a free culture in a free society.

8 LEGAL STATUS OF FONTS AND TYPEFACES

8.1 TYPEFACES

The US copyright office states that typeface designs are not subject to copyright (USCO, 2008). US courts have explained the reasoning behind this—in their view, the artwork aspect cannot be separated from the functional aspect:

> If the sole intrinsic function of an article is its utility, the fact that the article is unique and attractively shaped will not qualify it as a work of art. However, if the shape of a utilitarian article incorporates features, such as artistic sculpture, carving, or pictorial representation, which can be identified separately and are capable of existing independently as a work of art, such features will be eligible for registration ... It is patent that typeface is an industrial design in which the design cannot exist independently and separately as a work of art.
> (The US court of appeals for the forth circuit, 1978)

However, in the UK designs, including typeface designs, are subject to copyright. This is for a period of 25 years (OPSI, 1988), much shorter than the copyright term for literary works of 70 years after the death of an author (OPSI, 1995). Perhaps this is in recognition of their functional aspects. But it is an anomaly and most legal systems exclude typeface designs from their copyright bargains (Stack, 2008).

Instead variants of the patent system are used, requiring registration and with a much shorter term than artistic copyrights. Copyrights restrict expressions of ideas and not ideas themselves, so that similar but independently developed works that are not exact of derived copies cannot be restricted by copyright holders. These are very similar to copyrights because they are granted automatically and can not be used to restrict independently created designs.

A 'design patent' in the USA lasts for 14 years (USPTO, 2005) and a 'registered community design right' in the EU lasts for 5 years but can be renewed for a maximum term of 25 years (EP, 1998). Additionally the EU grants designers 'unregistered community design rights' for 3 years.

Thus all typefaces older than 25 years are free because they are in the public domain worldwide.

8.2 FONTS

Since typefaces are not subject to copyright in the USA, and design patent registrations are "relatively rare because of the cost and effort involved" (Gaultney, 2003), the copyright status of fonts has been hotly contested there.

Referring to bitmap fonts in 1988, the US Copyright Office said that "A typeface as such is not registrable ... data that merely represents an electronic depiction of a particular typeface or individual letterform is also not registrable" (Walsh, 1996).

In 1998 this changed when United States District Judge Ronald Whyte ruled that vector outline data representations of typeface designs are subject to copyright because they are programs, and programs are subject to copyright as literal expressions. He recognised that while the vector point data is separate from the drawing program, it is also separate from the non-copyright typeface design and the placement of the points involves "some creativity" and is therefore subject to copyright (Whyte, 1998).

Typefaces are discrete from their implementations as software, and like all software fonts are subject to copyright as a kind of literary work for a term of the life of the author plus 70 years.

Part C

THE FREE FONT MOVEMENT

9 HOW FONTS ARE MADE FREE

9.1 THE FOUR FREEDOMS

Together the FSF free software definition and Debian free software guidelines clearly define what freedom means for regular programs, while the definition of free cultural works is similar but phrased more broadly (Free Software Foundation, 2008a; SPI, 2004; Hill, Möller et al., 2006). What does freedom mean for fonts?

Use: Fonts should be free for use in any purpose, in private or in public, by any party, in any field of endeavour. Restrictions on commercial or political or religious use, on how many devices a party can use a font, or on what formats a font can be converted to for use in unforeseen typesetting systems are not acceptable.

Study and private modification: Fonts should be privately modifiable, in any way. This requires users to be able to study them unhindered, so fonts should be available to their users in complete 'source code' form. Complete font source code includes and is not limited to: Glyph outlines; 'master' glyph outlines interpolated to generate outlines; skeletal paths used to design outlines; guidelines and guideline shapes; and features such as hinting and layout instructions. Proprietary, incomplete or completely missing source forms are problematic, but acceptable.

Redistribution: Any party must be able to sell, swap or give the font away at no charge. Requirements for royalties or other fees for such sale are not acceptable. Redistribution must be permitted in all forms; as an independent work, assimilated into another font, as part of a collection of fonts, on in part. Fonts should not place restrictions on other works that are distributed in combination with the font, such as the text of documents typeset with the font.

Redistribution of modifications: Derived fonts must be distributable under the same terms as the licence of the original. Requiring derived fonts to carry a different name or version number from the original, and to retain attribution notices of other contributors, is acceptable.

9.2 FONT SOURCE CODE

What 'source code' means for a font may seem ambiguous since fonts are not lexical programs. However, the version three of the GPL defines source code broadly:

> The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work. The "Corresponding Source" for a work in object code form means all the source code needed

> to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities ... The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source. The Corresponding Source for a work in source code form is that same work. (Appendix B.5)

This concepts apply directly to fonts: Fonts are developed with font editor programs that store them in 'source code' formats for development, and generate or 'compile' fonts in 'object code' formats for use.

The free software font editor FontForge has a 'spline font database' (SFD) format (Williams, 2008), while the proprietary editor FontLab has VFB format, and both can generate Type 1 'PFB,' TrueType 'TTF' or OpenType 'OTF' files (FontLab, 2006).

The GFDL deals with the concept of source for non-program works, and goes one step further, defining them as 'transparent' and 'opaque' formats: the former is "a machine-readable copy, represented *in a format whose specification is available to the general public,* that is suitable for revising the [work] straightforwardly with ... some widely available drawing editor ... A copy that is not "Transparent" is called "Opaque" ... Opaque formats include proprietary formats that can be read and edited only by proprietary [programs]"

(Free Software Foundation, 2002).

FontLab's VFB format is such an opaque format. This suggests fonts developed with FontLab are precluded from being free. But the differences between object code and source code for fonts are much smaller those for lexical programs[35]: The object code must contains all essential shape spacing and layout information, and all font editors can open and modify font object code because all the major formats are publicly documented.

So a free font developed with a proprietary editor is not a disaster, since the font object code is akin to incomplete source code. Users can still exercise their freedom to use and modify the most important aspects of the font with access only to the object data, unlike programs. Also, it is unlikely but conceivable that the VFB format might eventually be published so that a free software font editor could implement support for it.

However, such fonts do seriously disadvantage users of free editors as they cannot access all aspects of the font in "the preferred form for making modifications to it." The DejaVu project has encountered this problem. It is a community project that develops the Bitstream Vera font, which is free because the GNOME Foundation licenced it from Bitstream under a non-copyleft licence that requires derivatives to be renamed (Gettys, 2003).

35 The 'compilation' process that converts a human-readable source code program into a computer-processable creates object data that is useless to humans and practically unmodifiable.

However, only the font object code was licenced, so the source code for the hints are not available and the project does not modify them (personal communication with DejaVu project administrator Ben Laenen, 2008).

10 FREE FONT LICENCES

Fonts are typically redistributed in part when they are embedded into a digital document such as PDF. This is called 'sub-setting,' and means only the parts of the font needed for the characters in the document are included.

Strong copyleft licences like the GNU GPL make this situation complicated. When a GPL work is combined with another work, the whole must be licenced under the GPL or it cannot be distributed. This means that if GPL font is embedded in a PDF, it requires all other works in the PDF to also be licenced under the GPL—including the text of the document—or the document cannot be distributed.

Weak- and non-copylefts, such as CC-BY-SA or MIT X11 licences, do not cause this problem. The SIL Open Font licence (OFL, Appendix B.6) is a free software licence written specifically for fonts. Drafts were reviewed and refined at various free software and type design conferences (Spalinger and Gaultney, 2007).

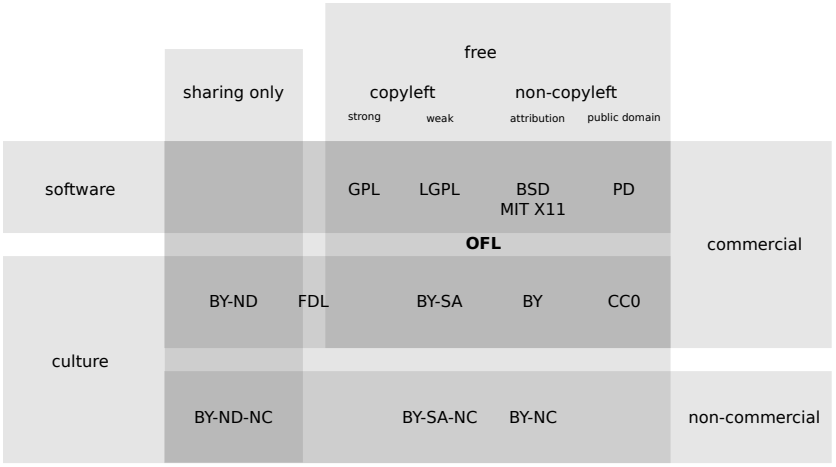


Figure 10: The SIL Open Font licence in the context of other free software and free culture licences.

The OFL accommodates type designers by being very simple, with the only major requirement being that derivatives must be renamed. For example, the 'Gentium' font published under the OFL by one of its authors, Victor Gaultney, can be freely modified and redistributed by anyone, but no one other than Gaultney may call their font 'Gentium.'

This means that like the CC-BY-SA licence it does not require distributors to make source code available on request, like the GPL (Appendix B.3). Neither of these licences or the GPL version 2 mention the issue of transparency, though the GPL requires source code to be made available on request. The other author of the licence, Nicolas Spalinger, explained the reasoning for this:

> One reason was to take into account the fact not all the designers were ready to release everything or would be able to make use of certain types of sources. Making it a cultural best practise to encourage releasing as much useful font sources as possible is the best way forward IMHO. Requiring that would be going too far for many designers.(Spalinger, 2007)

Therefore the OFL, CC-BY-SA, CC-BY or MIT X11 licences are suitable for fonts developed with proprietary programs such as FontLab or FontShop's FontStruct, which has a feature to allow users to apply Creative Commons licences to their FontStruct font object data conveniently (FontShop FSI, 2008).

If a font developer creates a new font with a free editor, they may wish to use copyleft to ensure that they are able to fully edit any derivative versions of the font. Version three of the GPL enables this since requires source code of published versions to "be in a format that is publicly documented (and with an implementation available to the public in source code form)" (Appendix B.5).

Version 3 of the GPL also permits the inclusion of 'additional terms,' similar to the renaming requirement of the OFL. It allows "prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version" (Appendix B.5).

The document embedding problem can also be avoided, by making exception to copyleft terms by granting additional permission to distribute the fonts in that way. The licensing section of the GNU website suggests the legal terms to eliminate the problem in this way, known as the 'font exception' (Free Software Foundation, 2008b).

11

BENEFITS OF FREEING FONTS

The primary benefit of freeing type is freedom itself, which is intangible but intrinsically valuable; it contributes to maintaining a free culture in a free society. Yet there are secondary tangible benefits that are derived from people in society exercising the freedom of free fonts.

One kind of change is incremental adjustments to improve the readability of a typeface. Readers can fix the bugs in a glyph's hinting instructions for a font they read on-screen with. The typographer of a periodical print publication may slowly develop optical variants for increased legibility at various printed sizes in response to feedback from readers.

In an unrecorded lecture at the Royal Society of Arts (RSA) in December 2007, Matthew Carter described how the 'Vincent' typeface's development occurred in this way; Newsweek was using this typeface for their headlines and requested additional variations to improve its utility.

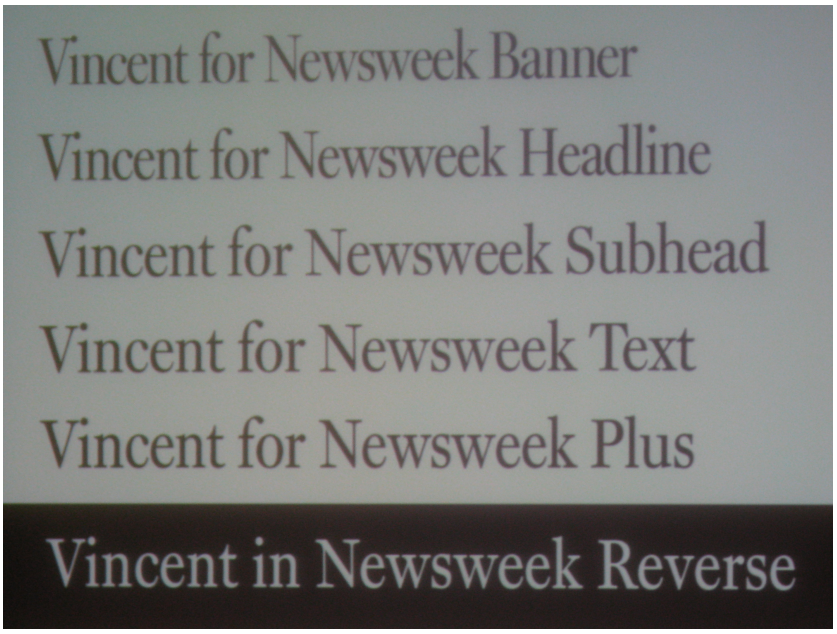


Figure 11: Optical variants of Matthew Carter's 'Vincent' typeface, digital slide projected a lecture at the RSA in December 2007.

Another kind of change is correcting the error of omission in a typeface: Typefaces can omit some characters in a script, and often

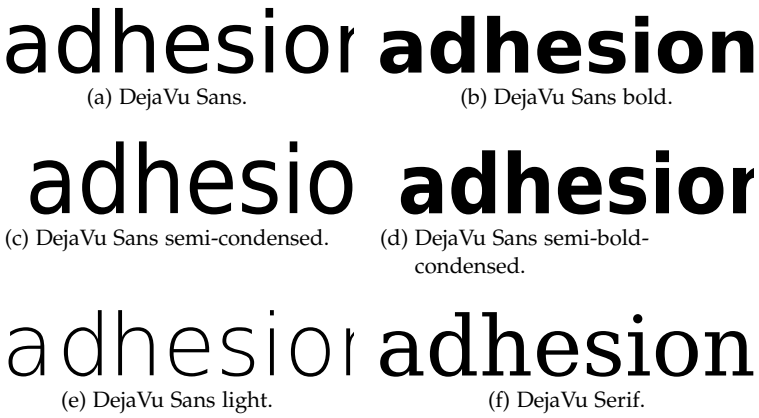
omit characters in closely related scripts. An omission in an encyclopedia or program is a similar kind of error. This is especially the case for non-Latin scripts which include comparatively high numbers of glyphs.

Adobe's policy for the full Cyrillic script is to omit 23 glyphs needed by 12,860,000 people in central Asia (Phinney, 2006). It is unjust that only the American proprietors can correct such errors. This kind of freedom is compelling because it enables 'access to knowledge'—literacy—in those linguistic communities.

Collaborative community development can not happen with proprietary fonts. The 'cathedral' metaphor in Raymond's 'The cathedral and the bazaar' seem an apt metaphor for type design and font software development; they are large undertakings traditionally done monastically in small teams.

Some of the first free fonts had their character range expanded in this way. The company developing GNU GhostScript, Artifex Software, licenced a set of fonts for implementing PostScript Level 2 from URW with the GPL version 2. Valek Filippov in Russia added Cyrillic glyphs, which were later improved by Owen Taylor at Red Hat in the USA. Glyphs for Uzbek and Vietnamese were added by others (Levine, 2005).

'Serial' collaboration contrasts with the 'parallel' collaboration style of the Linux kernel or Wikipedia. The DejaVu project homepage is an example of this, and its homepage states that its "purpose is to provide a wider range of characters while maintaining the original look and feel through the process of collaborative development" and the project lists 41 authors to date (Laenen et al., 2008b,a).



(a) DejaVu Sans. (b) DejaVu Sans bold.

(c) DejaVu Sans semi-condensed. (d) DejaVu Sans semi-bold-condensed.

(e) DejaVu Sans light. (f) DejaVu Serif.

Figure 12: The collaboratively designed DejaVu typeface family.

The project's technical processes involve software engineering practices that are common in the free software programming community but are unusual in type design, such as version control systems.

These are uncommon because the monolithic binary formats of proprietary font editors are not amenable to use in these systems, but free font source code formats are.

There are currently two free font source code formats developed for this purpose. RoboFab first implemented the Unified Font Object (UFO) format for this purpose (van Blokland et al., 2008), and FontForge similarly implements the 'SplineFont Directories' (SFDir) format, which are "sfd files split up into little bits in directories ... to give version control systems a finer granularity and reduce the amount of stuff to download after changes" (Williams, 2008).

The DejaVu project extended Vera with additional weights as well as expanding the range of glyphs, and this suggests collaborative type design can be effective because it is now the user interface font used by most GNU/Linux distributions instead of Vera (Alkalay et al., 2007).

12

THE BUSINESS OF FREEING FONTS

At the 2006 conference of the Association Typographique Internationale in Lisbon, graphic design professor and critic Ellen Lupton presented her view on the free font movement, and published a 'free font manifesto' on her website with a blog (Lupton, 2006a). Lupton presented the movement as non-commercial, and included some example fonts that were freeware rather than free cultural works. However she also identified the impetus for the movement stemming from the need to "serve relatively small or underserved linguistic communities."

Lupton received a negative reaction. In the comments section of her blog Peter Bruhn wrote, "In what way will there be food and shelter for my children" and an anonymous commentator wrote, "You say that type designers have no bread? Then let them eat cake." (Lupton, 2006b).

However, the development of many free fonts has been paid work. Simon Daniels, a program manager in Microsoft's typography group, commented on the blog that "with few exceptions the successful free fonts have had corporate, governmental or other organizational backers that have bankrolled their production" (Daniels, 2006).

For example, UNESCO commissioned Michael Everson to develop the 'Conakry' font for the minority script N'Ko (Everson, 2007) and SIL has employed Victor Gaultney for decades (LISA, 2003), while the GNOME Foundation licenced Vera from Bitstream, and Ascender has been commissioned to develop the 'Liberation' and 'Droid' fonts from Red Hat and Google (Ascender, 2008).

However, there is potential for a free font business using the business model pioneered by Cygnus. A type designer finds a client who wants to commission a new typeface as 'work for hire.' The client is faced with the full cost of the development, but can negotiate a lower price by trading exclusive access to the work.

This is normally so that the type designer can publish the font after some period of exclusivity on proprietary terms to the public. However it could also be so that the type designer can find another client with a similar brief, and both clients can share the costs of development. Professional type designer John Hudson has said that he has worked for clients in this way (personal communication, 2008).

It is also possible for groups of users to collectively raise money to 'bankroll' type design in this way: An initiator estimates the cost of the initial development, $\$\pi$, and estimates the number of people who are will to be patrons, β. The cost per patron is $\$\pi$ divided by β, $\$\Omega$.

The concept of the Pledgebank website is that the initiator can pledge contributing $\$\Omega$ if β other people do (Pledgebank, 2008). When the initiator has publicised the pledgebank enough for β people to

have made a pledge, the money can be collected and use to commission the work. The alternative is holding Ω amounts in escrow, which is administratively costly and risks that β is never reached.

This model is probably not as lucrative as having total power over clients and font users. Stallman has countered such suggestions by saying that "an 'economistic' approach to all these issues [is] a vehicle for unexamined assumptions ... about values, such as that amount of production matters, while freedom and way of life do not" (Stallman, 2006). This business model may generate enough revenue to be sustainable, while maintaining a free way of life.

Such free font businesses do not yet exist, but the free font movement is small. The way that Stallman's GCC was followed by Cygnus, and Torvald's kernel was followed by commercial distributions, suggests a trend of a non-commercial incubation phase, then a cottage industry phase, and then a large corporate commerce phase.

While other business models may be applied to the production of free fonts, perhaps it is impossible to produce them as full time professionals other than under the aegis of a patron. The London-based Demos think-tank published a pamphlet 2004 that described the rise of the 'professional–amateur' or 'pro-am' Leadbeater and Miller (2004).

Computers are omni-competent tools of cultural production and the Internet enables the publishing of any text, audio or video. Both have become extremely cheap in the developed world, so the full process of cultural production is available to anyone there who is not in poverty. The boundary between professionals and amateurs is blurring because both use the same authoring and publishing media.

This is transforming consumptive leisure time into productive leisure time; broadcast television is losing its audience to digital leisure activities. Eben Moglen, legal counsel to the FSF explained this in a speech:

> Michael Faraday first noticed what happened when he wrapped a coil of wire around a magnet and spun the magnet. Current flows in such a wire, but we don't ask what the incentive is for the electrons to leave home. We say that the current results from an emergent property of the system, which we call induction. The question we ask is "what's the resistance of the wire?"
>
> So Moglen's Metaphorical Corollary to Faraday's Law says that if you wrap the Internet around every person on the planet and spin the planet, software flows in the network. It's an emergent property of connected human minds that they create things for one another's pleasure and to conquer their uneasy sense of being too alone. The only question to ask is, what's the resistance of the network?(Moglen, 1999)

If people create cultural artifacts that are valuable, they may find opportunities to exchange that value for money. The popular Creative Commons non-commercial licences are therefore problematic; they risk blocking the development from a non-commercial incubation phase to a cottage industries and beyond.

13 FUTURE DEVELOPMENTS

GNU/Linux distributions include hundreds of fonts (Debian Font task force, 2008; Red Hat, 2008) whose developers have freely licenced the copyrights inherent in the font software. The GPL version 3 is the strongest copyleft licence available for fonts, and it is essential to include the 'font exception' additional permission for document embedding. However, best practices for provisions to require renaming of GPL derivatives have not yet been established.

In addition to the copyright of a font being freely licenced, the typeface it implements much be in public domain or freely licenced too. To make a typeface design freely available in the USA, the designer has to simply not apply for a design patent. However in the EU unregistered design rights are automatically granted for a period of 3 years, and these rights must be disclaimed. There has been no precedent for this in the free font community, though.

Also, free fonts must avoid names that infringe trademarks. It is possible to use the same first two characters for a similar typeface design, so a revival of 'Helvetica' called 'Helopen' would risk trademark infringement while 'Hernan' would not. An exception to this is that people's names and surnames can not be trademarks, so while 'Gill' 'Stone' 'Bell' 'Cooper' 'Zapf' and 'Warnock' are used as names for proprietary fonts, they are not valid trademarks (Stack, 2008).

A methodology or even technology for checking free fonts and type designs to avoid infringing design patents, design rights or trademarks ought to be established.

When people see type that they feel could be improved, if it is in the public domain they are free to do so. With the development of free software font editors, everyone has access to the necessary tools. Now the obstacle is that the knowledge of how to develop and refine type is not freely available.

There is also no central repository from which to easily feed a free font into the various free culture distribution channels, such as each of the popular GNU/Linux distributions. The closest projects to this so far are the 'Open Font Library' and the 'Comprehensive TEX Archive Network' websites (TUG, 2008; Phillips and Prokoudine, 2008). But neither site provides visitors a means of browsing and download free fonts, akin to Bitstream's MyFonts website (Bitstream, 2008). Such a website might also become a central place for free software fonts to be linked from using the upcoming CSS3 web font linking technology, already available in free software browsers like Midori and Apple's Safari.

14 CONCLUSION

Today there are comparatively few free software fonts; the proprietary font development community has produced tens of thousands of typefaces[36] For the free font movement to produce as many, it must engage both the long tail of amateurs that the free software movement enables and for-profit type design professionals.

While today's professionals use proprietary business models, the biggest problem facing the new type designers is obscurity. Treating customers as friends rather than thieves and type design as a service instead of a product may prove an effective way for newcomers to enter the market while reinforcing a free society in the age of computer networks.

> The phrase 'democratization of typography' has become common, referring to the wide availability of the tools of production for type and typographic design. One may take this with some scepticism: after all, for the majority, the generation and production of these tools is still largely in the hands of a few corporations—thought the [free software] movement may provide an alternative. ... The watchwords remain: doubt, critique, reason, hope."
>
> — Robin Kinross (2004, p.182, the final page.)

36 Today Bitstream is one of the largest vendors of proprietary font software, selling 18,018 font families on its website Bitstream (2008).

BIBLIOGRAPHY

Alexa Internet (2008), *Alexa top 500 sites*. Accessed on 2008-09-09 at http://www.alexa.com/site/ds/top_sites?ts_mode=global (Cited on page 22.)

Alkalay, A, Rebbechi, D and Burgiss, H (2007), *Optimal use of fonts on GNU/Linux*. Accessed on 2008-09-14 at http://tldp.org/HOWTO/Font-HOWTO/fonts.html#dejavu (Cited on page 45.)

Arceneaux, J, Chassell, R, Gilmore, J, Tower, L and Stallman, R, editors (1988), *GNU's Bulletin*, number 7, Free Software Foundation, MA, USA. Also available at http://www.gnu.org/bulletins/bull7.html (Cited on page 10.)

Ascender (2008), *Ascender portfolio – custom and bespoke fonts*. Accessed on 2008-09-08 at http://www.ascendercorp.com/portfolio_custom.html (Cited on page 46.)

Bitstream (2008), *Search results for "" : MyFonts*. Accessed on 2008-09-04 at http://www.myfonts.com/search?params[searchtype]=fonts (Cited on pages 48 and 49.)

Brooks, F (1978), *The mythical man-month: Essays on software engineering*, Addison-Wesley Longman, MA, USA (Cited on page 18.)

Canonical (2008), *Ubuntu homepage*. Accessed on 2008-09-08 at http://www.debian.org (Cited on page 17.)

Charity Navigator (2008), *Charity navigator rating Free Software Foundation*. Accessed on 2008-08-31 at http://www.charitynavigator.org/index.cfm?bay=search.summary&orgid=8557 (Cited on page 10.)

Cheliotis, G (2007), *Creative Commons statistics from the CC-Monitor Project*. Accessed on 2008-09-06 at http://hoikoinoi.files.wordpress.com/2007/07/cc-monitor-findings-isummit.pdf (Cited on page 23.)

Coase, R (1937), *The nature of the firm*, in *Economica* 4 number 16, 386–405 (Cited on page 22.)

Correll, S (2003), *Graphite: Frequently asked questions*. Accessed on 2008-09-12 at http://scripts.sil.org/GraphiteFAQ (Cited on page 34.)

Creative Commons (2008), *Creative Commons licenses*. Accessed on 2008-09-06 at http://creativecommons.org/licenses/ (Cited on page 23.)

Daniels, S (2006), *"Who needs free fonts?"*. Accessed on 2008-09-08 at https://www.blogger.com/comment.g?blogID=34853894\&postID=115909186258742186 (Cited on page 46.)

Debian (2004), *General resolution: Status of the non-free section*. Accessed on 2008-09-06 at http://www.debian.org/vote/2004/vote_002 (Cited on page 16.)

Debian (2006), *General resolution: Why the GNU free documentation license is not suitable for Debian main*. Accessed on 2008-09-06 at http://www.debian.org/vote/2006/vote_001 (Cited on page 20.)

Debian (2008), *Debian homepage*. Accessed on 2008-08-31 at http://www.debian.org (Cited on page 17.)

Debian Font task force (2008), *Debian font review*. Accessed on 2008-09-14 at http://pkg-fonts.alioth.debian.org/review/ (Cited on page 48.)

deVilla, J (2007), *Copyright versus community in the age of computer networks*. Accessed on 2008-09-06 at http://www.globalnerdy.com/2007/07/06/richard-m-stallman-copyright-vs-community-in-the-age-of-computer-networks/ (Cited on pages 20 and 21.)

Dixon, C (2002), *Typeface classification*. Accessed on 2008-09-09 at http://www.stbride.org/friends/conference/twentiethcenturygraphiccommunication/TypefaceClassification.html (Cited on page 28.)

EP (1998), *Directive 98/71/EC of the European Parliament and of the Council 13 October 1998*. Accessed on 2008-09-12 at http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:31998L0071:EN:HTML (Cited on page 35.)

Everson, M (2007), *Designing and producing fonts for N'Ko*. Accessed on 2008-09-14 at http://www.evertype.com/fonts/nko/ (Cited on page 46.)

FontLab (2006), *FontLab Studio 5: User's manual for Macintosh*. Accessed on 2008-08-27 at http://www.fontlab.us/downloads/manuals/FLS5MacManual.zip (Cited on page 39.)

FontShop FSI (2008), *FontStruct: An Overview of Licensing*. Accessed on 2008-09-14 at http://fontstruct.fontshop.com/faq/licensing_legal_questions/an_overview_of_licensing (Cited on page 42.)

Free Software Foundation (1999), *Lesser General Public License version 2.2*. Accessed on 2008-09-14 at `http://www.gnu.org/licenses/old-licenses/lgpl-2.1.html` (Cited on page 14.)

Free Software Foundation (2000), *GNU Free Documentation License 1.1*. Accessed on 2008-09-06 at `http://www.gnu.org/licenses/old-licenses/fdl-1.1.html` (Cited on page 20.)

Free Software Foundation (2002), *GNU Free Documentation License 1.2*. Accessed on 2008-09-06 at `http://www.gnu.org/copyleft/fdl.html` (Cited on pages 20 and 39.)

Free Software Foundation (2008a), *The free software definition*. Accessed on 2008-08-27 at `http://www.gnu.org/philosophy/free-sw.html` (Cited on pages 8 and 38.)

Free Software Foundation (2008b), *Frequently asked questions about the GNU licenses*. Accessed on 2008-09-14 at `http://www.gnu.org/licenses/gpl-faq.html#FontException` (Cited on page 42.)

Friedman, T (2005), *The world is flat: A brief history of the globalized world in the 21st century*, Allen Lane–Penguin Group, London, UK (Cited on page 8.)

Gaultney, V (2003), *Font licensing and protection details*. Accessed on 2008-09-12 at `http://scripts.sil.org/UNESCO_Font_Lic` (Cited on page 35.)

Gettys, J (2003), *Bitstream Vera fonts readme*. Accessed on 2008-09-14 at `http://www.gnome.org/fonts/` (Cited on page 39.)

Gilmore, J (2006), *Marketing Cygnus Support – free software history*. Accessed on 2008-08-27 at `http://www.toad.com/gnu/cygnus/index.html` (Cited on page 11.)

Goldstein, O, Chassell, R and Tower, L, editors (1988), *GNU's Bulletin*, number 4, Free Software Foundation, MA, USA. Also available at `http://www.gnu.org/bulletins/bull4.html` (Cited on page 10.)

Haralambous, Y (2007), *Fonts & encodings*, O'Reilly Media, CA, USA (Cited on pages 32 and 33.)

Haughey, M (2006a), *Developing nations copyright license frees creativity across the digital divide*. Accessed on 2008-09-06 at `http://creativecommons.org/press-releases/entry/4397` (Cited on page 23.)

Haughey, M (2006b), *Developing nations copyright license frees creativity across the digital divide*. Accessed on 2008-09-06 at `http://people.debian.org/~evan/ccsummary.html` (Cited on page 23.)

Hill, B (2005), *Towards a standard of freedom: Creative Commons and the free software movement*. Accessed on 2008-09-06 at `http://mako.cc/writing/toward_a_standard_of_freedom.html` (Cited on page 24.)

Hill, B, Bacon, J, Burger, C, Jesse, J and Krstic, I (2006), *The official Ubuntu book*, Prentice Hall PTR, NJ, USA (Cited on page 17.)

Hill, B, Möller, E et al. (2006), *Definition of free cultural works*. Accessed on 2008-09-06 at `http://freedomdefined.org` (Cited on pages 24 and 38.)

Hoffer, E (1951), *The true believer: Thoughts on the nature of mass movements*, Harper & Row, NY, USA (Cited on page 5.)

Hudson, P (2008), *Quick as a Gnash*, in *Linux Format* 1 number 107. Accessed on 2008-08-27 at `http://www.linuxformat.co.uk/pdfs/download.php?PDF=LXF107.iview.pdf` (Cited on page 10.)

ISO (2007), *ISO/IEC 14496-22:2007 Information technology – Coding of audio-visual objects – Part 22: Open font format*. Accessed on 2008-09-12 at `http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=43466` (Cited on page 33.)

Karow, P (1987), *Digital formats for typefaces*, URW Verlag, Germany (Cited on page 31.)

Kennington, A (2006), *GNU General Public Licence. Open source good. GPL bad.* Accessed on 2008-08-31 at `http://www.topology.org/linux/gpl.html` (Cited on page 12.)

Kingdon, J, Chassell, R, Bloom, M, Shein, B, Tiemann, M, Stallman, R and Tower, L, editors (1990), *GNU's Bulletin*, number 8, Free Software Foundation, MA, USA. Also available at `http://www.gnu.org/bulletins/bull8.html` (Cited on page 10.)

Kinross, R (2004), *Modern typography: An essay in critical history*, Hyphen Press, London, UK (Cited on pages 26, 29, 32, and 49.)

Knuth, D (1979), *TEX and METAFONT: New directions in typesetting*, American Mathematical Society, MA, USA (Cited on page 5.)

Knuth, D (1986), *The TEXbook*, Addison-Wesley, MA, USA (Cited on page 5.)

Knuth, D (1999), *Digital typography*, Center for the Study of Language and Information, Leland Stanford Junior University, CA, USA (Cited on pages 4 and 5.)

Laenen, B, Jacquerye, D et al. (2008a), *DejaVuWiki authors*. Accessed on 2008-09-14 at http://dejavu.sourceforge.net/wiki/index.php/Authors (Cited on page 44.)

Laenen, B, Jacquerye, D et al. (2008b), *DejaVuWiki main page*. Accessed on 2008-09-14 at http://dejavu.sourceforge.net/wiki/index.php/Main_Page (Cited on page 44.)

Leadbeater, C and Miller, P (2004), *How enthusiasts are changing our economy and society*, Demos, London, UK. Also available at http://www.demos.co.uk/publications/proameconomy/ (Cited on page 47.)

Lessig, L (2004), *Free culture: How big media uses technology and the law to lock down culture and control creativity*, Penguin Press, NY, USA (Cited on pages 3 and 23.)

Lessig, L (2006), *On free, and the differences between culture and code*. Accessed on 2008-09-06 at http://video.google.com/videoplay?docid=7661663613180520595 (Cited on page 24.)

Lessig, L (2008), *iCommons Summit talk on the history of CCommons*. Accessed on 2008-09-06 at http://blip.tv/file/1163412 (Cited on page 22.)

Levien, R (2000), *Advogato: Interview: Donald E. Knuth*. Accessed on 2008-08-27 at http://www.advogato.org/article/28.html; author confirmed via personal communication (Appendix A.2) (Cited on page 5.)

Levine, R (2005), *Giving it away*. Accessed on 2008-09-14 at http://typophile.com/node/10293 (Cited on page 44.)

Levy, S (1984), *Hackers: Heroes of the computer revolution*, Anchor Press/Doubleday, NY, USA. Accessed on 2008-08-27 at http://www.gutenberg.org/ebooks/729 (Cited on pages 3, 4, and 5.)

Levy, S (2002), *Lawrence Lessig's supreme showdown*, in *Wired Magazine* 10 number 10. Accessed on 2008-08-27 at http://www.wired.com/wired/archive/10.10/lessig.html (Cited on page 22.)

Lie, H and Bos, B (1999), *Cascading style sheets: Designing for the web*, 2 edition, Addison Wesley, MA, USA (Cited on page 29.)

Linksvayer, M (2008), *Approved for free cultural works*. Accessed on 2008-09-06 at http://creativecommons.org/weblog/entry/8051 (Cited on page 24.)

LISA (2003), *Gentium: Providing type to the world*. Accessed on 2008-09-14 at http://www.lisa.org/archive_domain/newsletters/2003/4.3/gaultney.html (Cited on page 46.)

Lord, T (2007), *Happy birthday, cathedral & bazaar (comment)*. Accessed on 2008-09-06 at http://www.roughtype.com/archives/2007/05/happy_birthday.php (Cited on pages 15 and 19.)

Lupton, E (2006a), *Free font manifesto*. Accessed on 2008-09-14 at http://www.designwritingresearch.org/free_fonts.html (Cited on page 46.)

Lupton, E (2006b), *"Why would a typeface designer want to give a font away?"*. Accessed on 2008-09-14 at https://www.blogger.com/comment.g?blogID=34853894\&postID=115909195561855648 (Cited on page 46.)

McLuhan, M (1967), *The medium is the massage*, Ginko Press, CA, USA (Cited on page iii.)

Moglen, E (1999), *Anarchism triumphant*. Accessed on 2008-09-14 at http://firstmonday.org/issues/issue4_8/moglen/ (Cited on page 47.)

Moody, G (2001), *Rebel code: Linux and the open source revolution*, Penguin, London, UK (Cited on pages 10, 15, 16, 17, and 18.)

Morison, S (1951), *First principles of typography*, in *Cambridge authors' and printers' guides* 1 (Cited on page 27.)

Mundie, C (2001), *The commercial software model*. Accessed on 2008-08-31 at http://www.microsoft.com/presspass/exec/craig/05-03sharedsource.mspx (Cited on page 12.)

Myers, R (2008a), *The non-commercial fallacy*. Accessed on 2008-09-06 at http://aboutfoo.com/~robmyers/weblog/2004/08/17/the-noncommercial-fallacy/ (Cited on page 24.)

Myers, R (2008b), *The non-commercial fallacy revisited*. Accessed on 2008-09-06 at http://aboutfoo.com/~robmyers/weblog/2005/10/16/the-noncommercial-fallacy-revisited/ (Cited on page 24.)

Myers, R (2008c), *The non-commercial fallacy three*. Accessed on 2008-09-06 at http://aboutfoo.com/~robmyers/weblog/2005/10/23/the-noncommercial-fallacy-three/ (Cited on page 24.)

Myers, R (2008d), *Why the NC permission culture simply doesn't work*. Accessed on 2008-09-06 at http://aboutfoo.com/~robmyers/weblog/2006/11/08/why-the-nc-permission-culture-simply-doesnt-work/ (Cited on page 24.)

Nerode, N (2003), *Why you shouldn't use the GNU FDL*. Accessed on 2008-09-06 at http://home.twcny.rr.com/nerode/neroden/fdl.html (Cited on page 20.)

Noordzij, G (2005), *The stroke: Theory of writing*, Hyphen Press, London, UK (Cited on pages 26 and 32.)

OPSI (1988), *Copyright, designs and patents act 1998: Duration of copyright*. Accessed on 2008-09-12 at http://www.opsi.gov.uk/acts/acts1988/plain/ukpga_19880048_en_2#pt1-ch1-pb4 (Cited on page 35.)

OPSI (1995), *The duration of copyright and rights in performances regulations 1995: Duration of copyright in literary, dramatic, musical or artistic works*. Accessed on 2008-09-12 at http://www.opsi.gov.uk/si/si1995/Uksi_19953297_en_3.htm#mdiv5 (Cited on page 35.)

Perens, B (1999), *The open source definition*, in *Open sources: Voices from the open source revolution* pp. 171–188 (Cited on page 18.)

Phillips, J and Prokoudine, A (2008), *Open Font Library*. Accessed on 2008-09-04 at http://www.openfontlibrary.org (Cited on page 48.)

Phinney, T (2006), *Extending Cyrillic (and later Latin) character sets*. Accessed on 2008-09-09 at http://blogs.adobe.com/typblography/2006/08/defining_an_ext.html (Cited on page 44.)

Pledgebank (2008), *Pledgebank*. Accessed on 2008-09-14 at http://www.pledgebank.com (Cited on page 46.)

Puzo, J, editor (1987), *GNU's Bulletin*, number 2, Free Software Foundation, MA, USA. Also available at http://www.gnu.org/bulletins/bull2.txt (Cited on page 10.)

Ray, M (2008a), *Creative Commons non-commercial, the creative flowerbed licence*. Accessed on 2008-09-06 at http://mjr.towers.org.uk/blog/2005-1.html#ccnc0520 (Cited on page 24.)

Ray, M (2008b), *Re: [Fsuk-manchester] GNU/Linux and free speech*. Accessed on 2008-09-04 at http://lists.gnu.org/archive/html/fsuk-manchester/2008-01/msg00019.html (Cited on page 12.)

Raymond, E (1998), *Goodbye, "free software"; hello, "open source"*. Accessed on 2008-09-06 at http://web.archive.org/web/20020214222257/www.tuxedo.org/~esr/open-source.html (Cited on page 18.)

Raymond, E (2001), *The cathedral and the bazaar: Musings on Linux and open source by an accidental revolutionary*, revised edition, O'Reilly Media, CA, USA (Cited on page 13.)

Raymond, E, editor (1996), *The new hacker's dictionary*, third edition, MIT Press, MA, USA (Cited on page 7.)

Red Hat (2007), *Financial results for fiscal year 2007*. Accessed on 2008-09-06 at `http://media.corporate-ir.net/media_files/IROL/67/67156/RHAT2007AR.pdf` (Cited on page 16.)

Red Hat (2008), *FedoraProject Category:Packaged fonts*. Accessed on 2008-09-14 at `http://fedoraproject.org/wiki/Category:Packaged_fonts` (Cited on page 48.)

Refsnes Data (2008), *Browser statistics*. Accessed on 2008-09-09 at `http://www.w3schools.com/browsers/browsers_stats.asp` (Cited on page 19.)

Rubin, P, editor (1987), *GNU's Bulletin*, number 3, Free Software Foundation, MA, USA. Also available at `http://www.gnu.org/bulletins/bull3.html` (Cited on page 10.)

Rubini, A (2007), *Transcript: Richard Stallman, honorary degree speech, Pavia 2007*. Accessed on 2008-09-12 at `http://www.fsfe.org/en/fellows/ciaran/ciaran_s_free_software_notes/transcript_richard_stallman_honorary_degree_speech_pavia_2007` (Cited on page 7.)

Rubinstein, R (1988), *Digital typography: An introduction to type and composition for computer system design*, Addison-Wesley Longman, MA, USA (Cited on page 27.)

Sanger, L (2005), *The early history of Nupedia and Wikipedia: A memoir*, in *Open sources 2.0: The continuing evolution* pp. 307–338 (Cited on pages 21 and 22.)

Severance, C (1999), *Internet and web pioneers: Richard Stallman*. Accessed on 2008-09-06 at `http://www.youtube.com/watch?v=ihxGJueWb-I` (Cited on page 21.)

Sheedy, J, Tai, Y, Subbaram, M, Gowrisankaran, S and Hayes, J (2008), *ClearType sub-pixel text rendering: Preference, legibility and reading performance*, in *Displays* 29 number 2, 138–151 (Cited on page 32.)

Shirky, C (2008), *Here comes everybody: The power of organizing without organizations*, Penguin Press, NY, USA (Cited on page 22.)

Smeijers, F (1996), *Counterpunch: Making type in the sixteenth century, designing typefaces now*, Hypen Press, London, UK (Cited on page 26.)

Smeijers, F (2003), *Type now: A manifesto plus work so far*, Hyphen Press, London, UK (Cited on pages 27 and 28.)

Southall, R (1985), *Designing new typefaces with METAFONT*, Stanford University, CA, USA. Accessed on 2008-09-06 at ftp://reports.stanford.edu/pub/cstr/reports/cs/tr/85/1074/CS-TR-85-1074.pdf (Cited on pages 26 and 31.)

Spalinger, N (2007), *[Openfontlibrary] red hat liberation fonts*. Accessed on 2008-08-27 at http://lists.freedesktop.org/archives/openfontlibrary/2007-May/000630.html (Cited on page 42.)

Spalinger, N and Gaultney, V (2007), *SIL Open Font License (OFL)*. Accessed on 2008-09-08 at http://scripts.sil.org/OFL (Cited on page 41.)

SPI (2004), *Debian social contract*. Accessed on 2008-09-06 at http://www.debian.org/social_contract (Cited on pages 16 and 38.)

SPI (2008), *2008-04-16 draft report*. Accessed on 2008-09-06 at http://www.spi-inc.org/treasurer/2008-04-16-draft-report.html (Cited on page 17.)

Spolsky, J (2007), *Font smoothing, anti-aliasing, and sub-pixel rendering*. Accessed on 2008-09-12 at http://www.joelonsoftware.com/items/2007/06/12.html (Cited on page 33.)

Stack, P (2008), *An overview of font IP related law*. Unpublished lecture notes from the Font Designers Rights Coalition *Business of type* summit, 2008-03-05, described at http://www.microsoft.com/typography/links/News.aspx?NID=6121 (Accessed on 2008-09-12) (Cited on pages 35 and 48.)

Stallman, R (1983), *New Unix implementation*. Accessed on 2008-08-27 at http://www.gnu.org/gnu/initial-announcement.html (Cited on pages 7, 8, and 20.)

Stallman, R (1984), *The GNU manifesto*, in *Free software, free society: Selected essays of Richard M. Stallman* pp. 31–41 (Cited on pages 8 and 10.)

Stallman, R (1996), *The GNU manifesto*, in *Free software, free society: Selected essays of Richard M. Stallman* pp. 67–68. The book edition states the article was first published in 2000, but the copyright notice on the web version lists 1996 and personal communication with Stallman confirmed this. Accessed on 2008-09-06 at http://www.gnu.org/philosophy/free-doc.html (Cited on page 20.)

Stallman, R (1999a), *The free universal encyclopedia and learning resource*. Accessed on 2008-09-06 at http://www.gnu.org/encyclopedia/free-encyclopedia.html (Cited on page 21.)

Stallman, R (1999b), *The GNU project*, in *Open sources: Voices from the open source revolution* pp. 53–70 (Cited on pages 6 and 20.)

Stallman, R (2000a), *Copyright versus community in the age of computer networks*. Accessed on 2008-09-06 at `http://www.gnu.org/philosophy/copyright-versus-community.html` (Cited on page 21.)

Stallman, R (2000b), *Why publishers should use the GNU FDL*. Accessed on 2008-09-06 at `http://www.gnu.org/licenses/why-gfdl.html` (Cited on page 20.)

Stallman, R (2001), *GNOME Foundation candidate statement*. Accessed on 2008-09-04 at `http://mail.gnome.org/archives/foundation-announce/2001-November/msg00028.html` (Cited on page 17.)

Stallman, R (2002a), *Free software, free society: Selected essays of Richard M. Stallman*, GNU Press, MA, USA (Cited on page 18.)

Stallman, R (2002b), *Let's share!* Accessed on 2008-09-06 at `http://www.opendemocracy.net/media-copyrightlaw/article_31.jsp` (Cited on page 21.)

Stallman, R (2005), *Fireworks in Montreal (2005-07-01 to 2005-07-05)*. Accessed on 2008-09-06 at `http://www.fsf.org/blogs/rms/entry-20050920.html` (Cited on page 23.)

Stallman, R (2006), *Did you say "Intellectual Property"? It's a seductive mirage*. Accessed on 2008-09-08 at `http://www.gnu.org/philosophy/not-ipr.html` (Cited on page 47.)

Stallman, R (2008), *GNU/Linux FAQ*. Accessed on 2008-09-06 at `http://www.gnu.org/gnu/gnu-linux-faq.html` (Cited on page 16.)

Thatcher, T, editor (1992), *Free software report*, number 1, Cygnus Support, CA, USA. Accessed on 2008-08-27 at `http://www.toad.com/gnu/cygnus/fsr/volume1.1.ps` (Cited on page 11.)

The US court of appeals for the forth circuit (1978), *Eltra Corp v. Ringer, 579 F.2d 294 (4th Circuit)*. Accessed on 2008-09-09 at `http://blogs.adobe.com/typblography/2006/08/defining_an_ext.html` (Cited on page 35.)

Tiemann, M (1999), *Future of cygnus solutions*, in *Open sources: Voices from the open source revolution* pp. 71–89 (Cited on pages 10 and 11.)

Top500.org (2007), *Operating system family share for 11/2007*. Accessed on 2008-09-10 at `http://www.top500.org/stats/list/30/osfam` (Cited on page 17.)

Torvalds, L (1991), *What would you like to see most in minix?* Accessed on 2008-08-27 at http://groups.google.com/group/comp.os.minix/msg/b813d52cbc5a044b (Cited on page 15.)

Torvalds, L (2001), *Just for fun: The story of an accidental revolutionary*, HarperCollins, NY, USA (Cited on page 15.)

Torvalds, L (2006), *Re: GPLv3 Position Statement*. Accessed on 2008-08-27 at http://lkml.org/lkml/2006/9/25/161 (Cited on page 19.)

Tower, L (1998), *Re: "open source" – new term for libre software*. Accessed on 2008-09-06 at http://www.crynwr.com/cgi-bin/ezmlm-cgi?mss:1105:nnkilbfmllojpflppole (Cited on page 18.)

Tower, L, Rubin, P, Chassell, R, Stallman, R and Goldstein, O (1988), *GNU's Bulletin*, number 5, Free Software Foundation, MA, USA. Also available at http://www.gnu.org/bulletins/bull5.html (Cited on page 12.)

TUG (2008), *The Comprehensive TEXArchive Network*. Accessed on 2008-09-12 at http://www.ctan.org (Cited on page 48.)

Unger, G (2006), *While you're reading*, Mark Batty Publisher, NY, USA (Cited on page 27.)

USCO (2008), *Code of regulations, title 37 section 202.1: Material not subject to copyright*. Accessed on 2008-09-12 at http://www.copyright.gov/title37/202/37cfr202-1.html (Cited on page 35.)

USPTO (2005), *A guide to filing a design patent application*. Accessed on 2008-09-12 at http://www.uspto.gov/web/offices/pac/design/ (Cited on page 35.)

van Blokland, E, van Rossum, J and Leming, T (2008), *UFO file format specification 1.0*. Accessed on 2008-08-27 at http://www.robofab.org/ufo/specification.html (Cited on page 45.)

Wallace, J and Erickson, J (1992), *Hard drive: Bill Gates and the making of the Microsoft empire*, John Wiley & Sons, NY, USA (Cited on page 4.)

Walsh, N (1996), *alt.fonts FAQ: Are fonts copyrightable?* Accessed on 2008-09-12 at http://nwalsh.com/comp.fonts/FAQ/cf_13.htm (Cited on page 36.)

Wayner, P (2000), *Free for all: How linux and the free software movement undercut the high-tech titans*, HarperCollins, NY, USA (Cited on pages 11 and 12.)

Whyte, R (1998), *Adobe Systems Inc. v. Southern Software Inc*. Accessed on 2008-09-12 at `http://lw.bna.com/lw/19980303/9520710.htm` (Cited on page 36.)

Wikimedia Foundation (2006), *Wikimedia Foundation financial statements – June 30, 2006, 2005, and 2004*. Accessed on 2008-09-09 at `http://upload.wikimedia.org/wikipedia/foundation/2/28/Wikimedia_2006_fs.pdf` (Cited on page 22.)

Wikimedia Foundation (2008), *Wikipedia*. Accessed on 2008-09-06 at `http://www.wikipedia.org/` (Cited on page 22.)

Williams, G (2008), *Spline font database file format*. Accessed on 2008-08-27 at `http://fontforge.sourceforge.net/sfdformat.html` (Cited on pages 39 and 45.)

Williams, S (2002), *Free as in freedom: Richard Stallman's crusade for free software*, O'Reilly & Associates, CA, USA (Cited on pages 5, 6, 13, and 15.)

Part D

APPENDIX

A EMAILS

This chapter of the appendix comprises copies of emails recieved while undertaking research and confirming details heard in conversation.

A.1 JIM BLANDY ON DEBIAN AND HURD

Jim Blandy has spent his entire career working at major free software companies such as Red Hat, CodeSourcery and Mozilla, and has contributed to famous free software projects such as Project GNU's Compiler Collection (GCC) and debugger (GDB), Subversion and Firefox. His homepage is at `http://www.red-bean.com/~jimb/`

```
Date: Thu, 28 Jun 2007 10:42:50 -0700
From: Jim Blandy <jimb@red-bean.com>
To: Dave Crossland <dave@lab6.com>
Subject: Re: HURD Development & History of the FSF

On 6/23/07, Dave Crossland <dave@lab6.com> wrote:
> I'm researching the history of the FSF, and I see you were
> (and are! :-) quite involved in development of the GNU system.

Yeah, I've pretty much made a career out of it. :)

> Do you know why the FSF has never shipped a bootable OS?

Well, first, let's clarify the distinction between Project GNU and the
Free Software Foundation.  Project GNU is the software project, which
anyone can participate in.  The FSF is a fundraising organization that
happens to sponsor some programmers working on Project GNU.  So the
question is more, "Why hasn't Project GNU ever shipped a bootable OS?"
(This distinction doesn't really matter, but since you said you were
casually researching the history, I thought you might like to know
this stuff too.)

The answer is that Project GNU chose Debian as its official
distribution, and let Debian ship a Hurd-based GNU system.  I believe
Debian does this.  So the premise of your question --- that GNU/Hurd
has never shipped --- isn't true.  Or, to answer it exactly as posed,
"It's not Project GNU's job to ship it; we delegated to Debian."

> (Was it waiting for HURD to become usable? Why not use Linux like it
> uses X and TeX?)

People do use the HURD, but mostly out of eccentricity or curiousity.
```

If you ask Stallman why Project GNU hasn't endorsed Linux as its official kernel, I think he'll say that "The HURD has technical advantages." I don't know if that's still his line, but it was a few years ago.

The HURD does have technical advantages over Linux; they're just not the technical advantages that people actually care about. :) In the HURD, an ordinary user, with no special privileges, can implement their own filesystem or terminal drivers, without disrupting other users of the system. So you can have a 'tar' filesystem, that simply makes the contents of a tar file appear as a read-only tree of files. You don't need the system administrator's permission to run it; it has no security implications. (Getting that to work that way is quite a trick.) You could have bash-style history and line editing as part of the ordinary tty interface, so that you don't have to build it into every shell and interactive program you want to use.

But nobody actually cares, for two reasons.

The first is political, or historical: back in the days where you had a VAX or some other big shared computer, filesystems and terminal drivers were basically impossible for ordinary people to experiment with. You had to have a whole machine reserved for your own use, which was out of reach for most people, especially if they just wanted to try something. In this context, the kind of flexibility the HURD offers is a real issue of freedom: it opens up new areas for experimentation. (Keep in mind that what's attractive about filesystems isn't so much how to store data on the disk, as that the idea of "here's a hierarchy where the leaves are strings of data" is a cool interface to hack on. So you can play indexing games, version control games, networking games, etc.) But if everyone has their own cheap PC whose kernel they can load modules into at will, then none of this is a really big deal. The freedom came in the form of cheap, powerful hardware, not in the form of software to help you share expensive hardware.

The second reason is technical: the HURD was designed by Thomas Bushnell and Roland McGrath. Both of them, but Thomas especially, are simply way too smart for their own good. They had a boundless supply of clever ideas, and implemented nearly all of them. The result is a system which only very few people can learn to hack on. Linux, by comparison, is rather simple. I think the fact that stack sizes are so limited in the kernel forces the Linux folks to keep call chains small and simple, and the result is that it's actually not that hard to find your way around. Linux has astonishingly complicated constraints on its data structures to get good SMP performance, and there are other hairy areas, but it's obviously something that lots of people are able to write device drivers for.

I hope that helps.

A.2 RAPH LEVIEN ON THE ADVOGATO KNUTH INTERVIEW

Raph Levien maintained important free software programs including 'GhostScript' and parts of the GNOME desktop. He founded 'Advogato,' a community blogging website for hackers that implemented his research on trust metrics. Recently he has developed several free software fonts using his own 'Spiro' tools, notable for implementing a new spline based on the Euler spiral. His homepage is at http://raph.levien.com

Date: Fri, 22 Aug 2008 16:44:35 -0700
From: Raph Levien <raph.levien@gmail.com>
To: Dave Crossland <dave@lab6.com>
Subject: Re: 2 second question: Were you the "Advogato" user on Advogato?

Yup, it's me! I gave permission to Steve Rainwater to use the account after I handed over the reins about 2 years ago, but I think he didn't feel comfortable doing so.

Raph

A.3 BRUCE PERENS ON DEBIAN AND HURD

Bruce Perens was an early contributor to the Debian project and took over leadership of the project from the founder. He co-founded 'Software in the Public Interest' and the 'Open Source Initiative,' and is the primary author of the 'Open Source Definition' and the 'Debian Free Software Guidelines.' Perens started the development of 'busybox,' used in most GNU/Linux distributions today.

Date: Sun, 24 Aug 2008 16:33:50 -0700
From: Bruce Perens <bruce@perens.com>
To: Dave Crossland <dave@lab6.com>
Subject: Re: History question about GNU and Debian

FSF paid Ian Murdock to develop Debian for some time, perhaps about 6 months. Unfortunately, Richard's technical direction was not compatible with Ian's, and Richard - although I admire him tremendously - can be a difficult person to work for.

Debian has indeed made a HURD distribution available. I don't believe this has ever been an official FSF project, because FSF is (reasonably, given FSF's goals) concerned about Debian's inclusion of the "non-free" section in the Debian distribution.

I think it's sufficient that a HURD distribution shipped - it need not have been an "official FSF edition" to realize that goal.

Richard, I'm sure, planned to write the HURD, like Emacs and the compiler, but his fingers burned out with RSI before he could do that. What a tragedy.

Thanks

Bruce

A.4 RICHARD STALLMAN ON COPYRIGHT AND AGREEMENTS TO NOT SHARE

From: Richard M. Stallman <rms@gnu.org>
To: Dave Crossland <dave@lab6.com>
Subject: Re: Copyright Vs Contract To Not Share
Date: Mon, 04 Aug 2008 22:05:19 -0400

If an unjust law forbids sharing, and you disregard it,
you have done nothing wrong. However, making a promise
not to share is wrong even if you do not fulfill it.

One could argue for a boycott of all works that cannot lawfully
be shared, but I think it is not ethically obligatory, and tactically
it is premature. Maybe some day this will be a good tactic
in some area.

B

LICENSES

This chapter of the appendix collects key copyright licenses used in the free software and free culture movements.

B.1 GPL 1, 1989

GNU General Public License Version 1, February 1989
Copyright © 1989 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA

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

Preamble

The license agreements of most software companies try to keep users at the mercy of those companies. By contrast, our 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. The General Public License applies to the Free Software Foundation's software and to any other program whose authors commit to using it. You can use it for your programs, too.
When we speak of free software, we are referring to freedom, not price. Specifically, the General Public License is designed to make sure that you have the freedom to give away or sell copies of free software, 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 a 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 tell them 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.
The precise terms and conditions for copying, distribution and modification follow.

Terms and conditions for copying, distribution and modification

0. This License Agreement 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 work containing the Program or a portion of it, either verbatim or with modifications. Each licensee is addressed as "you."

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 General Public License and to the absence of any warranty; and give any other recipients of the Program a copy of this General Public License along with the Program. You may charge a fee for the physical act of transferring a copy.

2. You may modify your copy or copies of the Program or any portion of it, and copy and distribute such modifications under the terms of Paragraph 1 above, provided that you also do the following:

    a) cause the modified files to carry prominent notices stating that you changed the files and the date of any change; and

    b) cause the whole of any work that you distribute or publish, that in whole or in part contains the Program or any part thereof, either with or without modifications, to be licensed at no charge to all third parties under the terms of this General Public License (except that you may choose to grant warranty protection to some or all third parties, at your option).

    c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the simplest and most usual 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 General Public License.

d) 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.

Mere aggregation of another independent work with the Program (or its derivative) on a volume of a storage or distribution medium does not bring the other work under the scope of these terms.

a) You may copy and distribute the Program (or a portion or derivative of it, under Paragraph 2) in object code or executable form under the terms of Paragraphs 1 and 2 above provided that you also do one of the following:

b) accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Paragraphs 1 and 2 above; or,

c) accompany it with a written offer, valid for at least three years, to give any third party free (except for a nominal charge for the cost of distribution) a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Paragraphs 1 and 2 above; or,

d) accompany it with the information you received as to where the corresponding source code may be obtained. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form alone.)

Source code for a work means the preferred form of the work for making modifications to it. For an executable file, complete source code means all the source code for all modules it contains; but, as a special exception, it need not include source code for modules which are standard libraries that accompany the operating system on which the executable file runs, or for standard header files or definitions files that accompany that operating system.

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

4. By copying, distributing or modifying 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.

5. 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.

6. 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 the 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 the license, you may choose any version ever published by the Free Software Foundation.

7. 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

8. 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.

9. In no event unless required by applicable law or agreed to in writing will any copyright holder, or any other party who may modify and/or redistribute the program as permitted above, be liable to you for damages, including any general, special, incidental or consequential damages arising out of the use or inability to use the program (including but not limited to loss of

DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

End of terms and conditions

How to apply these terms to your new programs

If you develop a new program, and you want it to be of the greatest possible use to humanity, 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 a brief idea of what it does.>
Copyright (C) 19yy  <name of author>

This program is free software; you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation; either version 1, or (at your option)
any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston MA  02110-1301 USA
```

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

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

```
Gnomovision version 69, Copyright (C) 19xx name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type ‘show w’.
This is free software, and you are welcome to redistribute it
under certain conditions; type ‘show c’ for details.
```

The hypothetical commands `show w` and `show c` should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w` and `show c`; they could even be mouse-clicks or menu items—whatever suits your program.
You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here a sample; alter the names:

> Yoyodyne, Inc., hereby disclaims all copyright interest in the program 'Gnomovision' (a program to direct compilers to make passes at assemblers) written by James Hacker.
>
> <signature of Ty Coon>, 1 April 1989
> Ty Coon, President of Vice

That's all there is to it!

B.2 BSD LICENSE, 1989

The 4.4BSD Copyright

All of the documentation and software included in the 4.4BSD and 4.4BSD-Lite Releases is copyrighted by The Regents of the University of California.

Copyright 1979, 1980, 1983, 1986, 1988, 1989, 1991, 1992, 1993, 1994
The Regents of the University of California. All rights reserved.

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

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. 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.
3. All advertising materials mentioning features or use of this software must display the following acknowledgement:

   > This product includes software developed by the University of California, Berkeley and its contributors.

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

The 'new' BSD license does not include term 3, and the FreeBSD project's license does not include terms 3 or 4

THIS SOFTWARE IS PROVIDED BY THE REGENTS 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 REGENTS 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.

B.3 GPL 2, 1991

Preamble

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

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

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

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

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

so that any problems introduced by others will not reflect on the original authors' reputations.
Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.
The precise terms and conditions for copying, distribution and modification follow.

Terms and conditions for copying, distribution and modification

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program," below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification.") Each licensee is addressed as "you."

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

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

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

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

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

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

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

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

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

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a

complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

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

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

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

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license

from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

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

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

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

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

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

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present

version, but may differ in detail to address new problems or concerns.

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

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

    No warranty

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

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

    End of terms and conditions

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.
To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year>  <name of author>

This program is free software; you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation; either version 2 of the License, or
(at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License along
with this program; if not, write to the Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.
```

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

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

```
Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type ‘show w’.
This is free software, and you are welcome to redistribute it
under certain conditions; type ‘show c’ for details.
```

The hypothetical commands `show w` and `show c` should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w` and `show c`; they could even be mouse-clicks or menu items—whatever suits your program.
You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

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

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

B.4 LGPL, 1991

GNU Library General Public License
Version 2, June 1991

This is the first released version of the library GPL. It is numbered 2 because it goes with version 2 of the ordinary GPL.

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 Library General Public License, applies to some specially designated Free Software Foundation software, and to any other libraries whose authors decide to use it. You can use it for your libraries, 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 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 a program 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.

Our method of protecting your rights has two steps: (1) copyright the library, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the library.
Also, for each distributor's protection, we want to make certain that everyone understands that there is no warranty for this free library. If the library is modified by someone else and passed on, we want its recipients to know that what they have is not the original version, 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 companies distributing free software will individually obtain patent licenses, thus in effect transforming the program into proprietary software. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.
Most GNU software, including some libraries, is covered by the ordinary GNU General Public License, which was designed for utility programs. This license, the GNU Library General Public License, applies to certain designated libraries. This license is quite different from the ordinary one; be sure to read it in full, and don't assume that anything in it is the same as in the ordinary license.
The reason we have a separate public license for some libraries is that they blur the distinction we usually make between modifying or adding to a program and simply using it. Linking a program with a library, without changing the library, is in some sense simply using the library, and is analogous to running a utility program or application program. However, in a textual and legal sense, the linked executable is a combined work, a derivative of the original library, and the ordinary General Public License treats it as such.
Because of this blurred distinction, using the ordinary General Public License for libraries did not effectively promote software sharing, because most developers did not use the libraries. We concluded that weaker conditions might promote sharing better.
However, unrestricted linking of non-free programs would deprive the users of those programs of all benefit from the free status of the libraries themselves. This Library General Public License is intended to permit developers of non-free programs to use free libraries, while preserving your freedom as a user of such programs to change the free libraries that are incorporated in them. (We have not seen how to achieve this as regards changes in header files, but we have achieved it as regards changes in the actual functions of the Library.) The hope is that this will lead to faster development of free libraries.
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, while the latter only works together with the library.
Note that it is possible for a library to be covered by the ordinary General Public License rather than by this special one.

Terms and conditions for copying, distribution and modification

0. This License Agreement applies to any software library which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Library 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 compile 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) 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.

   c) 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.

   d) 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 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.

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 to 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 Library 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.

End of terms and conditions

How to Apply These Terms to Your New Libraries

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 a brief idea of what it does.>
Copyright (C) <year>  <name of author>

This library is free software; you can redistribute it and/or
modify it under the terms of the GNU Library General Public
License as published by the Free Software Foundation; either
version 2 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
Library General Public License for more details.

You should have received a copy of the GNU Library General Public
License along with this library; if not, write to the Free
Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301  USA
```

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

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

> Yoyodyne, Inc., hereby disclaims all copyright interest in the library 'Frob' (a library for tweaking knobs) written by James Random Hacker.
>
> <signature of Ty Coon>, 1 April 1990
> Ty Coon, President of Vice

That's all there is to it!

B.5 GPL 3, 2007

GNU General Public License
Version 3, 29 June 2007

Copyright © 2007 Free Software Foundation, Inc. `http://fsf.org/`

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

Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.
The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program–to make

sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. 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 them 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 prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.
For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. 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.
Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.
For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.
Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.
Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.
The precise terms and conditions for copying, distribution and modification follow.

Terms and Conditions

0. Definitions

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you." "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component," in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control

copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law

   No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

   When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies

   You may convey 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; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

   You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions

   You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

   a) The work must carry prominent notices stating that you modified it, and giving a relevant date.

   b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices."

c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.

d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.

b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product," which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in

no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or

b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or

c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or

d) Limiting the use for publicity purposes of names of licensors or authors of the material; or

e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or

f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a)

provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies

   You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients

    Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

    An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

    You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not

initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version."

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others' Freedom

    If 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 convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License

    Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special

requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License

    The Free Software Foundation may publish revised and/or new versions of the GNU 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 that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

    If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

    Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty

    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.

16. Limitation of Liability

    IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS 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.

17. Interpretation of Sections 15 and 16

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

End of Terms and Conditions

How to Apply These Terms to Your New Programs

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

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state 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 a brief idea of what it does.>

Copyright (C) <textyear>  <name of author>

This program is free software: you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation, either version 3 of the License, or
(at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program.  If not, see <http://www.gnu.org/licenses/>.
```

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

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

```
<program>  Copyright (C) <year>  <name of author>

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

The hypothetical commands `show w` and `show c` should show the appropriate parts of the General Public License. Of course, your program's commands might be different; for a GUI interface, you would use an "about box."

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary. For more information on this, and how to apply and follow the GNU GPL, see `http://www.gnu.org/licenses/`

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read `http://www.gnu.org/philosophy/why-not-lgpl.html`

B.6 OFL 1.1, 2007

SIL Open Font License Version 1.1 – 26 February 2007

Preamble

The goals of the Open Font License (OFL) are to stimulate worldwide development of collaborative font projects, to support the font creation efforts of academic and linguistic communities, and to provide a free and open framework in which fonts may be shared and improved in partnership with others.
The OFL allows the licensed fonts to be used, studied, modified and redistributed freely as long as they are not sold by themselves. The fonts, including any derivative works, can be bundled, embedded, redistributed and/or sold with any software provided that any reserved names are not used by derivative works. The fonts and derivatives, however, cannot be released under any other type of license. The requirement for fonts to remain under this license does not apply to any document created using the fonts or their derivatives.

Definitions

"Font Software" refers to the set of files released by the Copyright Holder(s) under this license and clearly marked as such. This may include source files, build scripts and documentation.

Termination

This license becomes null and void if any of the above conditions are not met.

Disclaimer

THE FONT SOFTWARE IS PROVIDED "AS IS," WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF COPYRIGHT, PATENT, TRADEMARK, OR OTHER RIGHT. IN NO EVENT SHALL THE COPYRIGHT HOLDER BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, INCLUDING ANY GENERAL, SPECIAL, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF THE USE OR INABILITY TO USE THE FONT SOFTWARE OR FROM OTHER DEALINGS IN THE FONT SOFTWARE.

COLOPHON

This thesis is written and typeset using only free software: GNU/Linux, GNU Emacs 23, AucTeX, pdfLaTeXand the `classicthesis` package with Paladio and Euler fonts.

ABOUT THE AUTHOR

David Crossland <dave@lab6.com> runs Understanding Limited, a small business that takes care of information design and GNU/Linux systems administration projects. He is currently a committee member of the UK TeX User Group and makes regular public speaking engagements on the topic of this dissertation.