

OPEN SOURCE SOFTWARE LICENSING: THEORY AND PRACTICE

by John Ackermann N8UR

ira@febo.com

This presentation was prepared on behalf of NCR Corp. and is used with permission.
Copyright 2003-2005 NCR Corp.
All Rights Reserved

I. A BRIEF HISTORY OF OPEN SOURCE SOFTWARE, PART 1

“Open Source” is a philosophy of software development and distribution. Of the two words, “source” is the easy part to explain. It refers to source code, which is the form in which most programs are created. Source code is usually written by and can, at least theoretically, be read and understood by human beings (some source code is actually generated by other computer programs, but we won't go there). Although humans can understand source code, computers can't, and a tool called a compiler is used to convert source code to a binary format (an extremely long string of ones and zeroes) that a computer can digest and execute. The binary version of a program is often called “object code” and an object code version that includes all the pieces necessary to run on a computer is called an “executable.”

Source code is important to lawyers as well as programmers because it discloses the software's secrets. No one can look at the millions of ones and zeroes in an object code file and learn much about how the program works. However, a skilled programmer can understand a program by studying its source code and, by modifying the source and recompiling it, can create a modified version of the original. In the traditional software industry, source code is considered almost synonymous with “trade secret,” and companies go to great lengths to keep their source code away from prying eyes. Lawyers make their living writing license agreements that protect source code.

However, things weren't always that way. In the days when dinosaurs roamed the machine room (through the 1960s), software normally wasn't considered valuable in and of itself – it was just something that made having a computer worthwhile. IBM routinely made available

the source code for its mainframe computer applications.¹ At the time, it was unclear whether computer programs could even be protected by copyright.

It was the development of an independent software industry in the 1970s that led to the idea of keeping source code a secret. Vendors whose profits came from bits rather than nuts and bolts had no desire to let their competitors, or for that matter their customers, dig into the guts of their products. As the industry grew, it became unheard of to distribute source code.

And that is how things stayed until the mid-1990s. It was taken for granted that distributing source was the surest way to commercial ruin. Where business needs required the disclosure of source code (for example to a hardware vendor porting to a new machine), the required license agreement had non-disclosure terms so stringent that just about all the licensee could do with its employees after they had seen the source code was shoot them.

And that brings us to the “open” in Open Source. Open Source (the term is normally capitalized) is the opposite of this proprietary model of software development. As its name implies, Open Source software is developed through an open process made possible in large part by the Internet. It is shared, and licensed, in source code form, and people are free to modify it as they choose. It is normally, but not invariably, made available to users at no charge. Before talking more about Open Source, which is a development philosophy, it’s important to describe the idea of “Free Software” which is a political philosophy. And that political philosophy drives the license that lawyers love to argue about, the General Public License – the “GPL”, on which most of this paper will focus.

¹ It took a long time for that idea to die out. The hardware reference manual for the original IBM PC, published in 1981, included a full source code listing of the computer’s BIOS system. Shortly thereafter, IBM had a change of heart and spent the next decade or more threatening companies that developed BIOSes that were a bit too similar to IBM’s.

II. THE PHILOSOPHY OF FREE SOFTWARE

Before continuing with our history of how Open Source became the force that it is, it is necessary to take a slight detour to explain the first of the two philosophies that underlie Open Source software. Although “free software” or “freeware” is often used generically for any software that is distributed without charge, the Free Software Foundation (“FSF”), an organization we’ll shortly talk much more about, defines it strictly:

Free software is a matter of the users' freedom to run, copy, distribute, study, change and improve the software. More precisely, it refers to four kinds of freedom, for the users of the software:

- 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.

(<http://www.fsf.org/philosophy/free-sw.html>; a page that is well worth your attention if you are interested in the basis of the Free Software movement.)

Free Software by this definition is virtually always Open Source software. But not all Open Source software is Free Software; some Open Source licenses, according to the FSF, do not grant the requisite level of freedom.² An acronym that’s gained currency in the last few years is “FOSS” – Free and Open Source Software – to describe the intersection of the two concepts.

The Free Software idea arose from the small group of computer enthusiasts in the academic and hobbyist worlds – let’s call them “hackers” in the positive sense of that term – who

² The FSF web site (<http://www.fsf.org/licenses/license-list.html>) lists licenses which are “GPL-compatible free software licenses,” “GPL-incompatible free software licenses,” and “non-free software licenses.”

during the 1980s and 1990s began espousing the libertarian ideas that “information wants to be free”³ and that source code should be freely distributed. As shown by the FSF’s definition, the “free” in “Free Software” isn’t about price, but about freedom.

A commonly used catchphrase is “free as in speech, not free as in beer” – in other words, there’s much more to Free Software than its price. The fact that the English word “free” has both an economic and a political meaning concerns Free Software advocates, because it confuses what are to them two very distinct ideas; in fact, many advocates prefer the Spanish “software libre” as a more precise way to describe the essential nature of their beliefs.

In the late 80’s a group of these hackers organized the FSF with the stated goal of developing a complete Unix operating system clone that could be freely distributed with source code. They called their project “GNU,” which stands for “GNU is Not Unix” (the hacker sense of humor is a mysterious thing). The FSF proceeded to develop some very powerful tools that would be necessary components of a free operating system. They developed the concept of Free Software and wrote the GPL to implement it for the code they wrote.

The FSF encouraged other developers to use the GPL, and it has become the predominant license agreement in the Open Source world. Because of its importance, I’ll dissect the GPL, and one of its cousins, in much greater detail later in this paper. Prior to that, it’s important to understand why the GPL is what it is – a political as much as a legal document.

The GPL and similar licenses are designed with one key concept in mind: all users must have the freedom to modify and redistribute software licensed under its terms. You may build upon the contributions of others to the community, but you cannot hijack them. No one can

³ I’m not using this quote in the proper context and this use doesn’t really reflect what the author intended, but it has become common shorthand for the view that information should tend toward community, rather than private, ownership.

create a proprietary version of a GPL'd program by distributing only the object code, while keeping the source code locked up.

The Free Software Foundation coined the term “copyleft” to describe the philosophy of the GPL: “Copyleft says that anyone who redistributes the software, with or without changes, must pass along the freedom to further copy and change it. Copyleft guarantees that every user has freedom” (<http://www.fsf.org/copyleft/copyleft.html>).

III. A BRIEF HISTORY OF OPEN SOURCE SOFTWARE – PART 2

The GNU project remained, at least to the commercial software world, little more than a curiosity until about 1994, when two other groups almost simultaneously developed clones of the key Unix component called the “kernel” and made them available under free-software licenses. These were the “FreeBSD”⁴ and “Linux” projects. Both continue today, though Linux is by far the most popular and gathers the most publicity (and, until the crash, the most venture capital).⁵ The Linux kernel is licensed under the GPL, and that fact has had a major impact on the FOSS world.

The Linux kernel, coupled with the work FSF had already done on GNU, meant that hobbyists and students could put together a complete Unix-like system running on a PC computer that rivaled the power of commercial machines. And, because all this code was under the GPL or similar licenses, no one could divert it down a proprietary development path.⁶

⁴ There are also at least two other Open Source kernels derived from the same underlying BSD code – OpenBSD and NetBSD. Both are highly respected, but neither commands a major part of the Open Source market.

⁵ For the last couple of years, the Open Source world has been following the claims made by The SCO Group that Linux infringes copyrights on Unix that SCO now holds. That case is so complicated, and things are moving so rapidly, that it's hopeless to try to explain it here. If you're interested in the bizarre twists and turns of SCO vs. The World, Groklaw (<http://www.groklaw.net>) is the one and only place you need to go.

⁶ Another reason that it's difficult to co-opt Open Source projects is that most of them are the creation of so many authors that it would be virtually impossible to obtain suitable licenses from everyone who owns a piece of the copyright.

Until recently, the thought of using Open Source software didn't even occur to most IT organizations. There was a strongly held view that the only reliable software was commercial software, and that "free" software by definition was shoddy and amateurish. The fact that there was no commercial support available for non-commercial software sealed its fate in the corporate world.

As IT folks with the hacker nature began playing with Linux and FreeBSD at home, they noticed that, contrary to received wisdom, the software was actually very stable. In fact, there's evidence that these free upstarts are more reliable than most of their commercial counterparts. They also noticed that Open Source software tended to have frequent updates, rich features, and good performance. They began to question why such software wouldn't work in the business environment as well as the home and academic ones.

One of the most powerful arguments made for Open Source is that it may be the best path to high reliability. The open development process means that source code goes through a peer review and quality checking process that's far beyond the capability of any closed development shop. There are two reasons for this. First, the test process is open to the public, so the code is exercised under a broader range of conditions than a private company could create. Second, the availability of source code means that many eyes, not just a privileged few, are examining it and looking for flaws.

The combination of rapid development cycles and peer review has shown its value in the security area. Like all software, Open Source programs have bugs, and some of those bugs create security vulnerabilities. However, unlike proprietary products, which have long development cycles that discourage frequent updates, Open Source bugs are found and fixed

very rapidly. And, using the Internet as a distribution channel means that updates are available virtually instantly.

Gradually, Open Source began to penetrate the business world. Internet servers were the first toehold, and today a very large proportion of web traffic is hosted on Open Source systems. In fact, as of July, 2005, 70% of the web servers on the Internet use Apache, an Open Source product (compared to 21% for Microsoft),⁷ and the vast majority of all Internet email travels through Open Source mail servers. There are numerous other examples of Open Source software at the heart of the web.

Although Open Source on the desktop is not nearly as widespread as its use in in the server farm, Linux-based desktops are popping up, and the OpenOffice.org applications suite is gaining popularity as an Open Source alternative to Microsoft Office. In 2005, the Firefox web browser gained a lot of publicity and won 8% of the browser market from Microsoft – a small number in absolute terms, but a huge wake-up call to Microsoft. In early 2004, a leaked IBM memo challenged the IBM IT organization to move to a Microsoft-free infrastructure by the end of 2004 (it appears that IBM has backpedaled on that challenge, however).

⁷ Source: Netcraft Web Server Survey (<http://www.netcraft.com>).

IV. THE PHILOSOPHY OF OPEN SOURCE SOFTWARE

While the philosophy of Free Software is a political one, the philosophy of Open Source software is a practical one. In the brief history above, for practical purposes I left the impression that all non-commercial software is Open Source. That's not true. Not all free (as in beer) software is developed in an open environment, and much of it, particularly in the DOS/Windows world, is distributed only in object code form. Sometimes, source code is available, but a small group maintains tight control over development and user contributions aren't welcomed.

Open Source software is a distinct species of free-as-in-beer software characterized by development philosophy rather than license terms. It relies on development in the public eye, using the Internet as a medium to let many developers contribute to the work, and to allow testing by anyone and everyone. Actually, the distinction between "testing" and "using" Open Source programs is often blurred – Linux was widely used long before it reached the magical "version 1.0" that symbolizes a production release.

Open Source development is inherently a cooperative venture. Developers are usually volunteers, doing the work because they enjoy it, or perhaps because the product under development solves a problem that they are currently facing. A number of projects – notably the Linux kernel – have generated sufficient commercial interest that companies have assigned paid programmers to work on them, but that's still the exception rather than the rule. In an Open Source project, the group leader may serve more as a project manager than as a programmer. Depending on the origin of the project, the group leader may provide the overall architecture of the program, ensure that contributors are developing code that fits into that scheme, and make the final decision about what goes into the code base. Other projects have a more democratic focus, and these decisions may be made by committee, or even by a voting process.

The development of Linux is the prime example of the Open Source model. Originally conceived by Linus Torvalds, a Finnish computer science student, the initial versions of the Linux kernel were an outgrowth of his classwork, but he made them available on the 'net and rapidly gained a large following of programmers who, because it was a cool idea, made major contributions to the code. The current Linux kernel incorporates the work of literally thousands of developers, and Linus serves as coordinator, spokesman, and keeper of the faith, ensuring that the overall project remains true to its goal.

Certain contributors, based on their proven mastery of their subject matter, have become de facto owners of parts of the Linux kernel, and they make decisions within their sphere (subject always to Linus having the final say). Developers communicate via email lists, and a strong culture of consensus-building exists. All beta versions of the code are immediately available on the Internet. A bug reporting mechanism allows all users to report problems and suggest enhancements.

And, of course, the “official” development group is only part of the story of an Open Source project. Since the source code is always available, anyone can make their own modifications, some of which may even be improvements, and both use and distribute them. Thus, if the original development team loses interest, or no longer meets the needs of users, they hold no monopoly on the code. An offspring group can (and often has) taken over the project and moved it forward after all the original contributors are gone.

One of the leading proponents of the Open Source development model, Eric S. Raymond, has written a seminal article called “The Cathedral and the Bazaar,” (www.kde.org/food/cathedral/cathedral-paper.html⁸). In it, he analyzes through the example of

⁸ I hope this link works; the official URL for the paper seems to have disappeared.

an actual Open Source software development project why the noisy and “promiscuous” development style typified by Linux seems to work better than the quiet crafting of elegant systems in a closed environment (in other words, the vision, if not the reality, of typical commercial development). After noting that his relatively small project ended up with over one thousand individual contributors, he closes with the insight that:

Perhaps in the end the Open Source culture will triumph not because cooperation is morally right or software ‘hoarding’ is morally wrong (assuming you believe the latter, which neither Linus nor I do), but simply because the closed-source world cannot win an evolutionary arms race with Open Source communities that can put orders of magnitude more skilled time into a problem.”

To summarize the philosophical underpinnings of Open Source, I can’t do better than quote from www.opensource.org/intro.html:

The **basic idea behind Open Source** is very simple. When programmers on the Internet can read, redistribute, and modify the source for a piece of software, **it evolves**. People improve it, people adapt it, people fix bugs. And this can happen at a speed that, if one is used to the slow pace of conventional software development, seems **astounding**.

We in the Open Source community have learned that this rapid evolutionary process produces **better** software than the traditional closed model, in which only a very few programmers can see source and everybody else must blindly use an opaque block of bits. (Emphasis in original.)

V. THE COMMERCIAL BASIS OF OPEN SOURCE SOFTWARE

Linux and FreeBSD, and other lesser-known Open Source projects, were curiosities at first, but gradually people began to realize that it might be possible to make money from this unique licensing model. Prior to the dot-com crash, Linux in particular became a favorite of the investment world, and numerous companies went public to massive stock-price pops by having the word “Linux” in their names. Although the investment bubble has burst, Open Source has continued its infiltration, and it is no longer something the commercial world can ignore.

How does an Open Source company survive if it can’t charge big bucks for software licenses? Most follow one of two business models:

1. Providing support services. Just because source code is available doesn’t mean that you have the skill to do anything with it. Many, perhaps most, Open Source companies focus on software support, including activities such as publishing documentation, conducting training, providing technical support, and performing substantial customization to meet a customer’s unique requirements.

2. Providing Linux “distributions.” Linux itself isn’t a complete solution. It needs to be packaged with hundreds of other programs to form a complete computing environment. Several groups, both commercial and non-profit, have developed Linux distributions (which are usually made available on CD-ROM) that include these programs as well as tools to install and manage the system. The most famous distribution is RedHat, which although currently trading at a small fraction of its IPO price, has managed to be profitable since 2002.

A third model, based on the idea that anyone who knew enough buzzwords could float an IPO, has proven itself to be of limited viability, and most of the companies built around it have disappeared since 2000.

Having described the environment of Open Source, I'll now go into painful detail about the primary license documents used to implement that model. I will focus first and foremost on the GNU General Public License and its rather odd but very important offspring, the "Lesser GPL," as these are the most common, the most complex, and the most troublesome of the free software licenses. I will also provide brief descriptions of the BSD License, the MIT License, and the Artistic License. Copies of all these licenses are included at the end of this paper.

VI. THE GPL

In 1988, the Free Software Foundation published the first version of the General Public License. The second, and current, version was published in 1990.⁹ The GPL has been the subject of much heated discussion among both computer lawyers and hacker-lawyer-wannabes. Much of the heat stems from the fact that the GPL isn't a particularly clearly written document, and it requires careful parsing to make sense.

Another reason for controversy is the fact that the GPL approaches the world from a perspective totally different than that of traditional licenses. The software licenses that most lawyers are familiar with focus primarily on limiting the rights of the licensee to use the code; distribution to third parties is usually flatly prohibited in a very few unambiguous words. The GPL takes the opposite tack. It focuses exclusively on distribution and modification, not on use. And the way distribution is handled in the GPL is likely to be unfamiliar: its focus is on encouraging, not limiting, distribution of the licensed program's source code.

Commercial entities wishing to use GPL'd software in their own products need to be aware of one thing above all others. The GPL is sometimes referred to as the "General Public Virus" because it "infects" derivative works. In simple terms, *incorporating any GPL'd code in*

your software makes the entire derivative work subject to the GPL, with the corresponding requirement to provide full source code, and allow unlimited distribution, to anyone who asks, for only the cost of reproduction. However, not every use of GPL'd code creates a derivative work. GPL opponents (including Microsoft) have tended to gloss over this fact. Use of GPL'd code together with proprietary code in a way that does not create a derivative work does not pose any copyright risk. In any event, you must think carefully and fully understand the relationship of the individual components before mixing GPL'd and proprietary code, because the end result may not be compatible with a closed-source business model.

With that preliminary warning given, let's walk through the GPL. Starting at the beginning, Section 0 (hackers count the way computers do) establishes the GPL's scope. The 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" (and "Program" is defined as any such program or work). Thus, a mere reference to the GPL in a program's source code files or the splash screen of an object code version is enough to make it a Program subject to the GPL.

The first departure from standard software licenses also appears in Section 0 with the revelation that the GPL has nothing to do with the *use* of a Program: "[a]ctivities 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 . . ." ¹⁰

Section 1 allows unrestricted copying and distribution of the Program's source code in unmodified form. However, the distributor must "conspicuously and appropriately" include a

⁹ Version 3 of the GPL has long been talked about, but never seen. The latest word from the FSF is that an initial draft will be published in 2006.

copyright notice and disclaimer of warranty, and must keep all references to the GPL intact. Contrary to myth, the GPL doesn't require that software be given away for free. Section 1 expressly states that "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." On the other hand, the reality of the GPL makes it highly unlikely that any attempt to charge a significant fee to transfer a GPL'd program will succeed; anyone else has the right to undercut your transfer fee.

In apparent contradiction of Section 0's statement that a mere reference to the GPL is enough to bring a program under its terms, Section 1 requires that a copy of the GPL be included with any distributed copy. This inconsistency can be explained by reading Section 1 as a condition of distribution, not as a limit on the GPL's applicability. In other words, failure to include the GPL with a distributed copy is a violation of Section 1, but does not remove the Program from the GPL's coverage.

Section 2 addresses the right to modify the Program and is in many ways the heart of the GPL. Because the most practical way to modify a computer program is to work with the human-readable source code, this section focuses on source rather than object code. Section 2 begins with a simple and straightforward declaration: "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 . . ." However, this right is subject to compliance with three conditions.

The first and third conditions are fairly straightforward. Section 2(a) requires modified files to carry a prominent notice stating that they were changed. Section 2(c) requires Programs

¹⁰ Since activities other than copying, distribution, and modification are outside the scope of the GPL, could the right to run a Program be restricted by a separate "use" license that operates alongside the GPL? Could that license require a fee? The GPL does not give a clear answer to these questions.

(in most cases) to display a “splash screen” on startup that includes a copyright notice, a disclaimer of warranty, a statement that the Program may be redistributed under the GPL, and finally, instructions on how to view the GPL.

The second condition, Section 2(b), is more substantive. It makes the GPL incompatible with a closed-source environment by requiring any work 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.” This clause has two important consequences. First, the entire modified work must be distributed under the GPL; any new content added to the original work is thus automatically free software. This is the source of the “GPL as virus” concept.

Second, no one may charge a license fee (as distinct from a distribution charge) for access to software that is derived from GPL’d code. The purpose of the GPL, implemented in Section 2(b), is to prevent a modified Program from being “taken proprietary” or made subject to license fees.

The remainder of Section 2 elaborates on this requirement and creates a very limited exception to its breathtaking implications. The exception is contained in the second full paragraph, which states that although the three conditions on distribution apply to the work as a whole, “[i]f identifiable sections of [the] work are not derived from the Program, and can reasonably be considered separate and independent works in themselves,” those sections may be *separately distributed* free of the GPL. However, if they are distributed together with GPL’d code in a “derivative or collective work based on the Program,” the GPL will apply to the whole work.

It’s fair to ask what this means in the real world. Here is an example. Graphics hardware manufacturer VideoCo decides to develop a driver module that lets its products work with Linux.

In operation, that module is loaded into the computer's memory along with the Linux kernel, and communicates with the kernel via a set of defined interfaces. It is not, however, part of the same program file that contains the kernel on the hard disk and, apart from working with the video card, the kernel will work fine without the module.

If the module is developed completely from scratch and does not contain any GPL'd code, it could be distributed by itself as a stand-alone product under whatever license terms VideoCo chooses.¹¹ But if VideoCo decides it would its hardware sales would benefit if its driver module was included as part of the Linux kernel (which is distributed under the GPL), it may only do so by putting the module under the GPL and making its source code freely available. The GPL's intent is to ensure that source code is available for all the pieces that touch GPL'd Programs. The price of withholding source code is exclusion from the mainstream of free software distribution.

In an attempt to clarify, Section 2 goes on to state:

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.

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.

This makes it clear that the mere presence of a GPL'd Program doesn't infect an entire hard disk or CD-ROM.

¹¹ One might argue that the act of loading the module into memory along with the computer creates a derivative work, but that is the act of the user, not the module's author. The question is whether the module author would then be subject to a claim of contributory infringement under the GPL; it muddies the waters further to consider that someday someone might write a non-GPL but Linux-compatible kernel with which the module could be used.

Questions about mixing GPL'd and proprietary code at the copyright "compilation" level are very common. To avoid any possibility of a compilation claim, it would be wise to clearly separate the two species of software on any CD or other distribution medium. If feasible, it would be best to use two separate directory structures, one for GPL'd and the other for proprietary components. The installation process should be designed so that the GPL'd and proprietary components are installed in separate and visible steps.

Section 3 addresses object code versions of the Program and allows their distribution provided that any one of three conditions is met:

1. the object code is accompanied by a source code distribution that complies with Sections 1 and 2;
2. the distribution includes a written offer, valid for at least three years, to give the complete source code to any third party for no more than the cost of physically performing the distribution (*i.e.*, copying and mailing costs); or
3. for non-commercial distribution of object code versions, the distribution includes the information about receiving the source code that was received from the next person up the distribution chain (in other words, users may pass on object code versions to their friends if they retain the notices contained in the distribution they received).

The important result of Section 3 is that there is no such thing as "object code only" distribution in the GPL world. The right to modify software, guaranteed by the GPL, is only meaningful if source code is available, and to achieve this end the GPL makes it impossible to lock up the source code of programs under its protection.

Sections 2 and 3 form the heart of the GPL, and their requirements are the ones you will need to consider most carefully when considering distribute of GPL'd code. Some of the other GPL provisions are also quite surprising, and are worth careful review.

Section 5 makes the GPL operatively a shrinkwrap license while denying that it is a contract of adhesion. It also reemphasizes the License's focus on modification and distribution, but not use:

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 the Program.

Section 6 states that the recipient of each redistribution of the Program automatically receives a license from the original licensor to copy, distribute, or modify the Program, and that the Licensor may not impose any further restrictions on these rights. This is another important protection against the "hijacking" of GPL'd code: subsequent distributors cannot weaken the GPL by adding further restrictions. Section 6 also lets licensors breathe a bit easier by stating that they are not required to enforce third parties' compliance with the GPL.

Section 7 attempts to avoid limitations on the right to distribute (such as requiring royalty payments) that might result from infringement claims or the like. In short, if a court requires conditions on distribution that are inconsistent with the GPL, the right to distribute is simply terminated; the GPL's philosophy is that no distribution is better than restricted distribution. Section 8 furthers this by allowing the original licensor to prohibit distribution of the Program in countries that would impose restrictions based on patents or copyrighted interfaces.

Sections 11 and 12 include fairly standard warranty disclaimers and limitations on liability. These protections extend, to the extent permitted by law, to “all copyright holders, or any other party who may modify and/or redistribute the Program.”

In summary, the GPL’s goal is to ensure that users are free to modify and further distribute GPL’d Programs subject only to the requirement that they pass that freedom on to all subsequent users. The danger is that incautious use of GPL’d code can result in an application becoming free software without the copyright holder having any say in the matter.

VII. THE LESSER GPL

The FSF realized that there was one circumstance under which the GPL might be considered a bit presumptuous even by those fully supporting the concept of free software. That case is where an otherwise new and original work ends up incorporating (or “linking”) code from a pre-existing standard “library” of software routines into the run-time version. The use of libraries providing the code to implement commonly used functions is fundamental to modern computer programming; it avoids repetitive work and encourages portability of code from one environment to another. The compiler programs that are used to turn source code into executable programs typically include several libraries of standard functions for linking purposes, and Unix-based environments like Linux include a standard set of libraries upon which virtually all programs rely.

The catch is that if a GPL’d library is used, the resulting combination of object code and library is a derivative work subject to the GPL and all its requirements for free source code distribution.¹² In other words, a program developed completely independently but compiled using GPL’d standard libraries becomes subject to the GPL by the mere fact of being linked with those libraries. That is a result that could be charitably viewed as over-reaching.

The Lesser GPL¹³ is important because it (albeit grudgingly) recognizes that this is not always a desirable outcome and adjusts distribution requirements for programs linking to LGPL’d libraries accordingly. It acknowledges that use of these libraries does not impact the resulting executable program. This is an important issue because the GNU C compiler (“gcc”)

¹² It’s questionable whether that’s really true. Dynamic linking to a library that is already in the computer’s memory does not create a copy, and given the GPL’s disclaimer of control over use of the work, one could argue that this use of a GPL’d library does not create a derivative work. However, the GPL would govern the copying and distribution of a program that is statically linked against a GPL’d library.

¹³ The Lesser GPL was formerly known as the Library GPL; the name was changed because the FSF thought the previous name provided too much encouragement to use this license for libraries rather than the original GPL.

and its accompanying LGPL'd libraries are very commonly used in both open and closed source development projects. Unfortunately, the LGPL is even more difficult to parse than its cousin, and subtle technical distinctions can have major legal ramifications.

The LGPL is a modified version of the GPL with the addition of language addressing how licensed libraries may be used. Section 0 defines a “library” as “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.” For technical reasons that may become apparent below, the term “executables” is used in the LGPL instead of the very similar concept of “object code” used in the GPL.

The LGPL creates an important distinction between a “work based on the Library,” which contains code derived from the library (for example, a modified version of the library), and a “work that uses the Library,” which must be combined with the Library in order to run (for example, a program that uses functions provided by the Library at compile-time).

This distinction is critical. Under LGPL Section 2, a “work based on the Library,” *i.e.*, a modified version of the Library, is subject to essentially the same source code availability requirements as the GPL, with some additional requirements to maintain compatibility between the original and modified versions. Because modification and distribution of libraries in source code form is a situation that will arise but seldom in commercial settings, I won't talk further in this paper about “works based on the Library.”

Section 5 defines a “work that uses the Library” more specifically as “a program that contains no derivative of any portion of the Library, but is intended to work with the Library by being compiled or linked with it,” and says that “[s]uch a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.” Although this sounds

as though it might give such programs a way out of the LGPL structure, the story is not quite so simple. Section 5 goes on to state that this exemption applies only to certain, usually source code, versions of the program:

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 the terms for distribution of such executables.

At bottom, this somewhat circular language means that source or object code versions that merely reference functions contained in the Library do not fall under the Lesser GPL, while executable versions that directly incorporate Library code to implement such functions, do.¹⁴

So, what does Section 6 allow us to do with such executables? Not surprisingly, this is a lengthy and complex provision. In short, executables that include code from the Library may be distributed under terms of the licensor’s choosing, “provided that the terms permit modification of the work for the customer’s own use and reverse engineering for debugging such modifications.” This is a significant proviso, and it is likely to conflict with standard software license terms for end users; as a result, the Lesser GPL does not totally solve the problem of using GPL tools in closed-source projects.

Section 6 imposes further requirements largely aimed at implementing the licensee’s ability to modify the executable. In addition to establishing copyright and other notice provisions tracking those in the GPL, it requires that the licensor make available the source code and tools necessary to permit the user to modify the Library (but not necessarily the independently-developed work) and then re-link to create a new executable program. Five ways to do this are offered, with two primary results:

¹⁴ The same issue noted in Footnote 12 applies with respect to dynamically linked programs – it’s questionable whether copyright law would recognize such a situation as creating a derivative work.

1. If the program is “statically linked,” the licensor must make available the full Library in source code form and also make the work using the Library available separately in either object or source code form.¹⁵ Static linking means that the Library is permanently linked into the executable program during the compilation process. This is common in the DOS world, but less so in Unix-like environments. In this case, the licensor must make available all the tools (other than those that normally come with the operating system environment) necessary to reproduce the executable from the combination of the object code version and the Library. A significant commercial concern is that if those tools are themselves proprietary, the licensor theoretically has to acquire a license for any user that requests one, and cannot pass through any license fees it pays to do so.

2. If the program is dynamically linked to a shared library that already exists on the computer (in other words, linking occurs when the program is run, not when it is compiled), no further action is necessary. Most Linux and FreeBSD systems primarily use shared libraries, so this option makes the LGPL work well in those environments. This option also applies to “DLL” libraries in the Windows world.

In summary, the Lesser GPL allows libraries adopting it to be fairly easily used in shared-library environments such as FreeBSD, Linux, and possibly Windows. It does not work as well (from the closed-source developer’s point of view) in “static library” environments, because it requires distributing intermediate files and development tools that are not normally given to end users. If it’s practical to live under these rules, the LGPL can be a useful tool.

¹⁵ Although not commonly done, most compilers allow the “intermediate” object code files that exist after compilation, but before linking, to be saved. These files can be distributed to comply with Section 6 because they allow relinking with modified Libraries to create a new executable.

VIII. THE BSD LICENSE

Significant portions of modern Unix implementations (both free and commercial) derive from work done at the University of California – Berkeley. The original Unix work there was called the Berkeley Software Distribution or “BSD” for short. The software was made freely available by the University under what became known as the BSD License. The BSD License’s terms are short and straightforward (particularly in comparison with the GPL) and this license is used for many free software projects, including the FreeBSD Unix clone.

The BSD License simply states that “redistribution and use in source and binary forms, with or without modification, are permitted.” It imposes a few non-painful conditions: retention of copyright notices and license terms on redistribution; an agreement not to use the copyright holder’s name without consent; a disclaimer of all warranties; and a complete exclusion of all liability on the part of the copyright holders and contributors. Unlike the GPL, the BSD License does not attempt to enforce freedom by “infecting” derivatives or requiring source code availability.

An earlier version of the BSD license included a clause requiring that credit be given to the copyright holder; the FSF opined that this “advertising” clause was inconsistent with the GPL, and that the BSD license was therefore not compatible. The advertising clause was removed several years ago, and current versions of the BSD license are deemed to be GPL-compatible.

IX. THE MIT LICENSE

Another significant component of modern Unix distributions is the X Windows System, which provides users with a graphical interface. X was originally developed at MIT, and is licensed under terms very similar to the revised BSD License (the one that does not include the advertising clause), but with a few more words defining the permitted uses, and a few less devoted to the liability limitation clause. The practical effect is that the MIT License and the revised BSD License are virtually indistinguishable.

X. THE ARTISTIC LICENSE

The Artistic License, under which Perl (an important Unix tool) and some other programs are licensed, attempts to be a simpler version of the GPL. It permits distribution of modified source code versions so long as the modifications are placed in the public domain or otherwise made freely available, or the executable programs are renamed so they do not conflict with the standard versions (which must be included in the modified distribution).

Object code versions may be distributed provided that the modified version are given non-standard names, the original object or source code versions are included in the package, and the differences between the original and modified versions are clearly documented.

The Artistic License permits the licensor to charge for distribution or support, but not for the software itself. Importantly (because Perl is a widely-used program development tool), scripts and library files provided as input to, or output from, the program do not fall under the License. In other words, a program developed using a tool licensed under the Artistic License does not fall under the License's terms, and may be licensed in whatever manner the developer wishes.

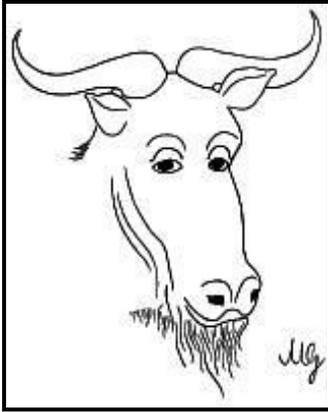
The original version of the Artistic License contains ambiguities which prevented the FSF from deeming it GPL-compatible. A revised version resolves these issues. Also, note that Perl language interpreter is actually licensed under the user's choice of either the Artistic License, or the GPL.

XI. CONCLUSION

Free software and the Open Source movement can no longer be ignored by the commercial software world. No one other than a few zealots expect that all software will adopt the Open Source model, but it's clear that free software will be finding its way into many development organizations. If you work for, or represent, an organization that uses a traditional closed-source development model, it's vitally important that you understand the significance of the GPL and other free software licenses, and use that knowledge to avoid development models that could result in the unintentional creation of new free software.

APPENDICES

GNU General Public License



[[Czech](#) \ [English](#) \ [Japanese](#)]

- [What to do if you see a possible GPL violation](#)
 - [Translation of the GPL](#)
 - [GPL Frequently Asked Questions](#)
 - The GNU General Public License (GPL) [in plain text format](#)
 - The GNU General Public License (GPL) [in Texinfo format](#)
 - The GNU General Public License (GPL) [as an appendix in Doc. Book format](#)
-

Table of Contents

- [GNU GENERAL PUBLIC LICENSE](#)
 - [Preamble](#)
 - [TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION](#)
 - [How to Apply These Terms to Your New Programs](#)
-

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

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

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

Preamble

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

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

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

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

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

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

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

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

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

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

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

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

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

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

- **a)** You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
- **b)** You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
- **c)** If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty)

and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

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

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

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

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

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

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

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

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

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

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

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

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

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

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

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

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

NO WARRANTY

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

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

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

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

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

one line to give the program's name and an idea of what it does.
Copyright (C) yyyy name of author

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

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

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA.

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

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

Gnomovision version 69, Copyright (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.

Return to [GNU's home page](#).

FSF & GNU inquiries & questions to gnu@gnu.org. Other [ways to contact](#) the FSF.

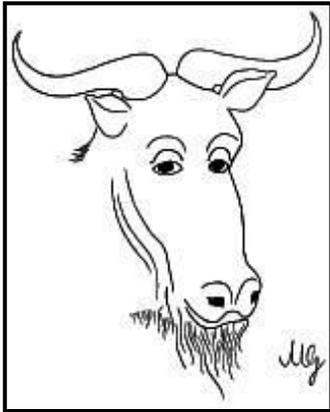
Comments on these web pages to webmaster@www.gnu.org, send other questions to gnu@gnu.org.

Copyright notice above.

Free Software Foundation, Inc., 59 Temple Place – Suite 330, Boston, MA 02111, USA

Updated: Last Modified: Sun Jul 15 13:13:30 CEST 2001

GNU Lesser General Public License



[English \ Japanese]

- [Why you shouldn't use the Lesser GPL for your next library](#)
- [What to do if you see a possible LGPL violation](#)
- [Translations of the LGPL](#)
- The GNU Lesser General Public License as a [text file](#)
- The GNU Lesser General Public License as a [Texinfo file](#)

This GNU Lesser General Public License counts as the successor of the GNU Library General Public License. For an explanation of why this change was necessary, read [the Why you shouldn't use the Lesser GPL for your next library](#) article.

Table of Contents

- GNU LESSER GENERAL PUBLIC LICENSE
 - [Preamble](#)
 - [TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION](#)
 - [How to Apply These Terms to Your New Libraries](#)

GNU LESSER GENERAL PUBLIC LICENSE

Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts
as the successor of the GNU Library Public License, version 2, hence
the version number 2.1.]

Preamble

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

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

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

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

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

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

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

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

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

- **a)** The modified work must itself be a software library.
- **b)** You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.
- **c)** You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.
- **d)** If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

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

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

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

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

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

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

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

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

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

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

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

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

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

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

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

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

- **a)** Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)
- **b)** Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.
- **c)** Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.
- **d)** If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.
- **e)** Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

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

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

- **a)** Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.
- **b)** Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

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

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

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to

satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

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

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

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

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

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

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

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

NO WARRANTY

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

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

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 an 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 Lesser General Public License as published by the Free Software Foundation; either version 2.1 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 Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

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

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!

Return to [GNU's home page](#).

FSF & GNU inquiries & questions to gnu@gnu.org. Other [ways to contact](#) the FSF.

Comments on these web pages to webmasters@www.gnu.org, send other questions to gnu@gnu.org.
Copyright notice above.

Free Software Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111, USA

Updated: 27 Nov 2000 paulv

The BSD License

The following is a BSD license template. To generate your own license, change the values of OWNER, ORGANIZATION and YEAR from their original values as given here, and substitute your own.

Note: The advertising clause in the license appearing on BSD Unix files was officially [rescinded](#) by the Director of the Office of Technology Licensing of the University of California on July 22 1999. He states that clause 3 is "hereby deleted in its entirety."

Note the new BSD license is thus equivalent to the [MIT License](#), except for the no-endorsement final clause.

<OWNER> = Regents of the University of California
 <ORGANIZATION> = University of California, Berkeley
 <YEAR> = 1998

In the original BSD license, the first occurrence of the phrase "COPYRIGHT HOLDERS AND CONTRIBUTORS" in the disclaimer read "REGENTS AND CONTRIBUTORS".

Here is the license template:

Copyright (c) <YEAR>, <OWNER>
 All rights reserved.

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

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of the <ORGANIZATION> nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE 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.

The MIT License

Copyright (c) <year> <copyright holders>

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

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

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

The Artistic License

Preamble

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

Definitions:

- "Package" refers to the collection of files distributed by the Copyright Holder, and derivatives of that collection of files created through textual modification.
- "Standard Version" refers to such a Package if it has not been modified, or has been modified in accordance with the wishes of the Copyright Holder.
- "Copyright Holder" is whoever is named in the copyright or copyrights for the package.
- "You" is you, if you're thinking about copying or distributing this Package.
- "Reasonable copying fee" is whatever you can justify on the basis of media cost, duplication charges, time of people involved, and so on. (You will not be required to justify it to the Copyright Holder, but only to the computing community at large as a market that must bear the fee.)
- "Freely Available" means that no fee is charged for the item itself, though there may be fees involved in handling the item. It also means that recipients of the item may redistribute it under the same conditions they received it.

1. You may make and give away verbatim copies of the source form of the Standard Version of this Package without restriction, provided that you duplicate all of the original copyright notices and associated disclaimers.
2. You may apply bug fixes, portability fixes and other modifications derived from the Public Domain or from the Copyright Holder. A Package modified in such a way shall still be considered the Standard Version.
3. You may otherwise modify your copy of this Package in any way, provided that you insert a prominent notice in each changed file stating how and when you changed that file, and provided that you do at least ONE of the following:
 - a) place your modifications in the Public Domain or otherwise make them Freely Available, such as by posting said modifications to Usenet or an equivalent medium, or placing the modifications on a major archive site such as ftp.uu.net, or by allowing the Copyright Holder to include your modifications in the Standard Version of the Package.
 - b) use the modified Package only within your corporation or organization.
 - c) rename any non-standard executables so the names do not conflict with standard executables, which must also be provided, and provide a separate manual page for each non-standard executable that clearly documents how it differs from the Standard Version.
 - d) make other distribution arrangements with the Copyright Holder.
4. You may distribute the programs of this Package in object code or executable form, provided that you do at least ONE of the following:
 - a) distribute a Standard Version of the executables and library files, together with instructions (in the manual page or equivalent) on where to get the Standard Version.

- b) accompany the distribution with the machine-readable source of the Package with your modifications.
 - c) accompany any non-standard executables with their corresponding Standard Version executables, giving the non-standard executables non-standard names, and clearly documenting the differences in manual pages (or equivalent), together with instructions on where to get the Standard Version.
 - d) make other distribution arrangements with the Copyright Holder.
5. You may charge a reasonable copying fee for any distribution of this Package. You may charge any fee you choose for support of this Package. You may not charge a fee for this Package itself. However, you may distribute this Package in aggregate with other (possibly commercial) programs as part of a larger (possibly commercial) software distribution provided that you do not advertise this Package as a product of your own.
 6. The scripts and library files supplied as input to or produced as output from the programs of this Package do not automatically fall under the copyright of this Package, but belong to whomever generated them, and may be sold commercially, and may be aggregated with this Package.
 7. C or perl subroutines supplied by you and linked into this Package shall not be considered part of this Package.
 8. The name of the Copyright Holder may not be used to endorse or promote products derived from this software without specific prior written permission.
 9. THIS PACKAGE IS PROVIDED "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

The End