GPLv3 First Discussion Draft Rationale

January 16, 2006

Contents

1 Introduction **1**

1.1 Do No Harm 1

1.2 Technological Changes and Legal Threats to Freedom 2

1.3 Design Goals 4

2 Section-by-Section Discussion of Changes **7**

2.1 0. Definitions 8

2.2 1. Source Code 8

2.3 2. Basic Permissions 10

2.4 3. Digital Restrictions Management 11

2.5 4. Verbatim Copying 11

2.6 5. Distributing Modified Source Versions 12

2.7 6. Non-Source Distribution 13

2.8 7. License Compatibility 14

2.9 8. Termination 17

2.10 9. Not a Contract 17

2.11 10. Automatic Licensing of Downstream Users 17

2.12 11. Licensing of Patents 17

2.13 12. Liberty or Death for the Program 18

2.14 [13. Geographical Limitations] 19

2.15 14. Revised Versions of this License 19

2.16 15. Requesting Exceptions 19

2.17 16–17. No Warranty 19

2.18 18. Safety Critical Systems 19

3 Conclusion **21**

Chapter 1

Introduction

In fifteen years of use, version 2 of the GNU General Public License has succeeded beyond our expectations. It has nurtured a spirit of cooperation and trust that has enabled a worldwide community of user/developers to release an extraordinary range of free software. The underlying principle of respect for users' right to cooperate has spread beyond the field of software, to inspire other creative and scientific endeavors.

The success of the GPL is due to its fundamental design principle: the protection of users' freedom to work individually or together to make software do what they wish. To carry the GPL into the future, we have undertaken to adapt the license to uphold this principle through the opportunities and menaces of today's technological and legal environment.

The core legal mechanism of the GNU GPL is that of copyleft, which requires modified versions of GPL'd software to be GPL'd themselves. Copyleft is essential for preventing the enclosure of the free software commons, today as it was in 1991. But today's environment is more complex and diverse; thus, a fully effective copyleft calls for additional legal measures. Devising these measures is complicated by another aspect of our success: the worldwide adoption of free software principles. We hope and expect that contributors to version 3 of the GPL will come from all over the globe, and from every developer, distributor, and user constituency.

1.1 Do No Harm

Through our years of work on this revision of the GNU GPL, we have remained committed to preserving the established freedoms on which free software users depend. We have also done our utmost to avoid unintentional consequences that would harm these freedoms. While we are confident that our draft, if adopted, will have no unforeseen consequences that would be deleterious to freedom, we must be certain that this will be so. Making sure of this is one primary reason for the public comment process.

To illustrate what this principle implies, consider the treatment of software designed for public use on network servers. Given the variety of needs and concerns in this area, in which different parties have disparate and strongly-held positions, we have chosen not to add requirements about public use of modified versions in the GPL itself. Instead we have made a variety of possible license requirements compatible with the GPL, through an enhanced compatibility provision; thus we leave individual developers scope for choosing among requirements to apply for public use of their code. We have intentionally done nothing that might threaten to divide free software developers from free software users.

1.2 Technological Changes and Legal Threats to Freedom

Computer technology has changed since 1991, but these changes are not primarily what has motivated us to revise the GPL. The concern of the GPL is not the particulars of technology but the maintenance of users' freedoms. To be sure, technological developments of the past fifteen years have enabled new freedoms and have resulted in new threats to freedom. No fundamental change in computer technology has occurred that requires a radical change to our license, however.

It is changes in law, not computer technology, that pose the principal challenges to the free software community. Chief among these changes has been the unwise and ill-considered application of patent law to software. Software patents threaten every free software project, just as they threaten proprietary software and custom software. Any program can be destroyed or crippled by a software patent belonging to someone who has no other connection to the program.

We were among the few to recognize the gravity of the software patent problem in 1991. At that time, however, the problem seemed to be confined to one country, the United States. Today the situation is very different. Most countries have followed the direction of the United States, permitting software to be patented to at least some degree. This worldwide shift in patent law has brought about immense harm and injustice. In 1991 GPLv2 was unique in raising a defense against the problem of software patents, in its section 7. It is indicative of the scale of this problem that, by the end of the decade, commentators were criticizing the GPL for doing too little to combat patents.

A program's own license cannot protect it from the threat of software patents. The only real solution to the problem of software patents is to abolish them. However, we can protect against attempts by some participants in a program's development to use patents against other participants. GPLv3 provides an explicit patent license covering any patents held by the program's developers, replacing the implicit license on which GPLv2 relies. GPLv3 also implements a narrow scheme of patent retaliation against those who undertake this precise form of aggression.

Our draft of GPLv3 makes clear that we do not entirely share the current enthusiasm of others in the free software community for including broad forms of patent retaliation in licenses. Theorists of patent retaliation have, in our view, overestimated the deterrent value of denying access to free software. In this area, we have chosen instead to follow our general guidelines of limiting freedom only where demonstrably necessary to protect freedom, and of doing no more in granting permissions than permissions can be expected to accomplish.

Technology that restricts users' traditional rights in copyrighted works, often known as Digital Restrictions Management or Digital Rights Management (DRM), is another threat to free software. As a campaign to limit users' rights, the adoption of DRM is fundamentally at odds with the spirit of the free software movement. Unfree software implementing DRM technology is simply a prison in which users can be put to deprive them of the rights that the law would otherwise allow them. Our aim is, and must be, the abolition of DRM as a social practice. Anything less than complete victory leaves the freedom of software in grave peril.

Free software is software that respects the user's essential freedoms; the adoption of free software is a step forward because it means the spread of freedom. Even as companies imposing DRM prohibit access to digitally restricted data by free users, they often seek to transform free software into tools of user restriction. We must not tolerate this assault on users' freedom merely because the software used for this assault is a version of our own.

Someday, we hope, copyright law's traditional respect for individual user rights will be restored, and user-disabling DRM will no longer be permitted. In the meantime, we have designed GPLv3 to forbid such perversion of free software.

Another challenge facing the free software community is the proliferation of incompatible free software licenses. Of course, we cannot make the GPL compatible with all such licenses. GPLv3 contains provisions that are designed to reduce license incompatibility by making it easier for developers to combine code carrying non-GPL terms with GPL'd code.

We hope to encourage more free software developers to use the GPL when licensing their software, and, more generally, we are determined to convince more developers of the merits of copyleft. The proponents of fully permissive, non-copyleft licenses have, in effect, argued in favor of sacrificing the preservation and extension of user freedom in order to facilitate the short-term commercialization of software. Our position has always been that software built in freedom might easily be rendered non-free if governed by such arrangements. Developments in the years since 1991 have only strengthened this view.

Although the concerns of business have never been our main priority, we do make one observation on this subject. For us, there has never been any inconsistency between protecting users' freedom and enabling the commercial use of software. Whatever doubts may have existed in 1991, we have shown since then that a copyleft license, a license designed for durable protection of user freedom, can form the basis of a larger set of commercially useful software than any non-copyleft free software license has ever produced. Although business concerns are secondary to freedom, it is important that the GNU GPL enable business to succeed while respecting freedom, and we do not intend to interfere with the synergy between them.

1.3 Design Goals

The GPL operates by granting permission beyond what copyright law itself requires. As a copyleft license, the GPL's primary goal is to defend a set of core freedoms for all software users. For this reason, the GPL places some requirements on the licensee, but only to the extent necessary to prevent some users from denying freedom to others. In a few parts of the revised license, we have reacted to alarming developments by adding certain new requirements, as in our section on DRM. These requirements are narrowly drawn and are directed at preventing the mechanism of freedom from being turned against itself.

We recognize that, overall, the changes made in GPLv3 have increased the complexity of the license. We would have liked to oblige those who have asked us for a simpler and shorter GPL, but we had to give priority to making GPLv3 do the job that needs to be done. We appreciate simplicity in licenses, but simplicity must not be allowed to interfere with the goal of protecting users' freedom. We enthusiastically invite those who believe that the GPL can protect freedom just as well in fewer words to join our comment process and show us how this can be done.

Chapter 2

Section-by-Section Discussion of Changes

Table of Corresponding Sections

GPLv2	GPLv3
0	0, 2
1	4
2	5
3	1, 7
4	8
5	9
6	10
7	12
8	[13]
9	14
10	15
11	16
12	17

2.1 0. Definitions

Section 0 includes definitions of two new terms: "covered work" and "propagate." The use of the term "covered work" enables some of the wording in the revised GPL to be simpler and clearer.

The term "propagate" serves two purposes. First, "propagate" provides a simple and convenient means for distinguishing between the kinds of uses of a work that the GPL imposes conditions on and the kinds of uses that the GPL does not (for the most part) impose conditions on.

Second, "propagate" furthers our goal of making the license as global as possible in its wording and effect. When a work is licensed under the GPL, the copyright law of some particular country will govern certain legal issues arising under the license. A term like "distribute," or its equivalent in languages other than English, is used in several national copyright statutes. The scope of "distribution" in the copyright context can differ from country to country. We do not wish to force on the GPL the specific meaning of "distribution" that exists under United States copyright law or any other country's copyright law.

We therefore define the term "propagate" by reference to activities that require permission under "applicable copyright law," but we exclude execution and private modification from the definition. Our definition gives examples of activities that may be included within "propagation," but it also makes clear that, under the copyright laws of a given country, "propagation" may include other activities as well.

Section 0 also clarifies that modification of a work includes extending the work, such as by adding text to the work. This was implicit in GPLv2.

2.2 1. Source Code

Section 1 retains GPLv2's definition of "source code" and adds an explicit definition of "object code" as "any non-source version of a work." Object code is not restricted to a narrow technical meaning and is to be understood broadly as including any form of the work other than the preferred form for making modifications to it. Object code therefore includes any kind of transformed version of source code, such as bytecode. The definition of object code also ensures that licensees cannot escape their obligations under the GPL by resorting to shrouded source or obfuscated programming.

The definition of "Complete Corresponding Source Code" given in the second paragraph of section 1 is as broad as necessary to protect users' exercise of their rights under the GPL. We follow the definition with particular examples to remove any doubt that they are to be considered Complete Corresponding Source Code. We wish to make completely clear that a licensee cannot avoid complying with the requirements of the GPL by dynamically linking an add-on component to the original version of a program.

Though the definition of Complete Corresponding Source Code in the second paragraph of section 1 is expansive, it is not sufficient to protect users' freedoms in many circumstances. For example, a GPL'd program, or a modified version of such a program, might need to be signed with a key or authorized with a code in order for it to run on a particular machine and function properly. Similarly, a program that produces digitally-restricted files might require a decryption code in order to read the output.

The third paragraph of section 1 addresses this problem by making clear that Complete Corresponding Source Code includes any such encryption, authorization, and decryption codes. By requiring the inclusion of this information whenever the GPL requires distribution of Complete Corresponding Source Code, we thwart efforts to obstruct the goals of the GPL, and we ensure that users will remain in control over their own machines. We recognize an exception where use of the program normally implies that the user already has the codes. For example, in secure systems a computer owner might possess any keys needed to run a program, while the distributor of the program might not have the keys.

The final paragraph of section 1 revises the exception to the source code distribution requirement in GPLv2 that we have sometimes called the system library exception. This exception has been read to prohibit certain distribution arrangements that we consider reasonable and have not sought to prevent, such as distribution of gcc linked with a non-free C library that is included as part of a larger non-free system. This is not to say that such non-free libraries are legitimate; rather, preventing free software from linking with these libraries would hurt free software more than it would hurt proprietary software.

As revised, the exception has two parts. Part (a) rewords the GPLv2 exception for clarity but also removes the words "unless that component itself accompanies the executable." By itself, (a) would be too permissive, allowing distributors to evade their responsibilities under the GPL. We have therefore added part (b) to specify when a system library that is an adjunct of a major essential operating system component, compiler, or interpreter does not trigger the requirement to distribute source code. The more low-level the functionality provided by the library, the more likely it is to be qualified for this exception.

2.3 2. Basic Permissions

We have included the first sentence of section 2 to further internationalize the GPL. Under the copyright laws of some countries, it may be necessary for a copyright license to include an explicit provision setting forth the duration of the rights being granted. In other countries, including the United States, such a provision is unnecessary but permissible.

The first paragraph of section 2 also acknowledges that licensees under the GPL enjoy rights of copyright fair use, or the equivalent under applicable law. These rights are compatible with, and not in conflict with, the freedoms that the GPL seeks to protect, and the GPL cannot and should not restrict them.

Section 2 distinguishes between activities of a licensee that are permitted without limitation and activities that trigger additional requirements. The second paragraph of section 2 guarantees the basic freedoms of privately modifying and running the program. However, the right to privately modify and run the program is terminated if the licensee brings a patent infringement lawsuit against anyone for activities relating to a work based on the program.

This narrowly-targeted patent retaliation provision is the only form of patent retaliation that GPLv3 imposes by its own force. We believe that it strikes a proper balance between preserving the freedom of a user to run and modify a program, and protecting the rights of other users to run, modify, copy, and distribute code free from threats by patent holders. It is particularly intended to discourage a GPL licensee from securing a patent directed to unreleased modifications of GPL'd code and then suing the original developers or others for making their own equivalent modifications.

Several other free software licenses include significantly broader patent retaliation provisions. In our view, too little is known about the consequences of these forms of patent retaliation. As we explain below, section 7 permits distribution of a GPL'd work that includes added parts covered by terms other than those of the GPL. Such terms may include certain kinds of patent retaliation provisions that are broader than those of section 2.

The third paragraph of section 2 represents another effort to compensate for variation in national copyright law. We distinguish between propagation that enables parties other than the licensee to make or receive copies, and other forms of propagation. As noted above, the meaning of "distribution" under copyright law varies from country to country, including with respect to whether making copies available to other parties (such as related public or corporate entities) is "distribution." "Propagation," however, is a term not tied to any statutory language. Propagation that does not enable other parties to make or receive copies — for example, making private copies or privately viewing the program — is permitted unconditionally. Propagation that does enable other parties to make or receive copies is permitted as "distribution," subject to the conditions set forth in sections 4–6.

2.4 3. Digital Restrictions Management

DRM is fundamentally in conflict with the freedoms of users that the GPL is designed to safeguard, but our ability to oppose DRM by means of free software licenses is limited. In section 3 we provide developers with some forms of leverage that they can use against DRM. The first paragraph essentially directs courts to interpret the GPL in light of a policy of discouraging and impeding DRM and other technical restrictions on users' freedoms and illegal invasions of users' privacy. This provides copyright holders and other GPL licensors with means to take action against activities contrary to users' freedom, if governments fail to act.

The second paragraph of section 3 declares that no GPL'd program is part of an effective technological protection measure, regardless of what the program does. Ill-advised legislation in the United States and other countries has prohibited circumvention of such technological measures. If a covered work is distributed as part of a system for generating or accessing certain data, the effect of this paragraph is to prevent someone from claiming that some other GPL'd program that accesses the same data is an illegal circumvention.

2.5 4. Verbatim Copying

Section 4 has been revised from its corresponding section in GPLv2 in light of the new section 7 on license compatibility. A distributor of verbatim copies of the program's source code must obey any existing additional terms that apply to parts of the program. In addition, the distributor is required to keep intact all license notices, including notices of such additional terms.

2.6 5. Distributing Modified Source Versions

Section 5 contains a number of changes relative to the corresponding section in GPLv2. Subsection 5a slightly relaxes the requirements regarding notice of changes to the program. In particular, the modified files themselves need no longer be marked. This reduces administrative burdens for developers of modified versions of GPL'd software.

Under subsection 5a, as in the corresponding provision of GPLv2, the notices must state "the date of any change," which we interpret to mean the date of one or more of the licensee's changes. The best practice would be to include the date of the latest change. However, in order to avoid requiring revision of programs distributed under "GPL version 2 or later," we have retained the existing wording.

Subsection 5b is the central copyleft provision of the license. It now states that the GPL applies to the whole of the work. The license must be unmodified, except as permitted by section 7, which allows GPL'd code to be combined with parts covered by certain other kinds of free software licensing terms. Another change in subsection 5b is the removal of the words "at no charge," which was often misinterpreted by commentators. The last sentence of subsection 5b explicitly recognizes the validity of disjunctive dual-licensing.

Subsection 5c generalizes the requirements in subsection 2c of GPLv2 to apply to various kinds of interactive user interfaces. The new text distinguishes between interfaces that present a list of user commands or options, such as a menu, and those that do not. For the first class of interfaces, the list must include a command to display the copyright notice and other information required by the first sentence of subsection 5c. For the second class of interfaces, which include command-line interfaces, voice-activated interfaces, and so on, the conditions are essentially the same as in GPLv2: the modified program must display the information at startup, unless the modification was to an interactive program that did not display such information at startup. The displayed information must include the central list of non-GPL terms applicable to added parts, as required by section 7.

The paragraph following subsection 5c has been revised for clarity, but the underlying meaning is unchanged. When independent non-derivative sections are distributed for use in a combination that is a covered work, the whole of the combination must be licensed under the GPL, regardless of the form in which such combination occurs, including combination by dynamic linking. The final sentence of the paragraph adapts this requirement to the new compatibility provisions of section 7.

The final paragraph of section 5 revises the "mere aggregation" exception of GPLv2. We expect that the revised wording will eliminate any uncertainty regarding the meaning of this provision. In particular, this provision makes clear that certain abuses involving the use of compilation copyrights are not allowed.

2.7 6. Non-Source Distribution

Section 6 of GPLv3, which clarifies and revises GPLv2 section 3, requires distributors of GPL'd object code to provide access to the corresponding source code, in one of four specified ways. As noted above, "object code" in GPLv3 is defined broadly to mean any non-source version of a work.

Subsections 6a and 6b now apply specifically to distribution of object code in a physical product. Physical products include embedded systems, as well as physical software distribution media such as CDs. As in GPLv2, the distribution of object code may either be accompanied by the machine-readable source code, or it may be accompanied by a written offer to provide the machine-readable source code to any third party. GPLv3 clarifies that the medium for software interchange on which the machine-readable source code is provided must be a durable physical medium. Subsection 6b does not prevent a distributor from offering to provide source code to a third party by some other means, such as transmission over a network, so long as the option of obtaining source code on a physical medium is presented.

Subsection 6b revises the requirements for the written offer to provide source code. As before, the offer must remain valid for at least three years. In addition, even after three years, a distributor of a product containing GPL'd object code must offer to provide source code for as long as the distributor also continues to offer spare parts or customer support for the product model. We believe that this is a reasonable and appropriate requirement; a distributor should be prepared to provide source code if he or she is prepared to provide support for other aspects of a physical product.

Subsection 6b also increases the maximum permitted price for providing a copy of the source code. GPLv2 stated that the price could be no more than the cost of physically performing source distribution; GPLv3 allows the price to be up to ten times the distributor's cost. It may not be practical to expect some organizations to provide such copies at cost. Moreover, permitting such organizations to charge ten times the cost is not particularly harmful, since some recipient of the code can be expected to make the code freely available on a public network server. We also recognize that there is nothing wrong with profiting from providing copies of source code, provided that the price of a copy is not so unreasonably high as to make it effectively unavailable.

Subsection 6c gives narrower permission than the corresponding subsection in GPLv2. The option of including a copy of an offer received in accordance with subsection 6b is available only for private distribution of object code; moreover, such private distribution is restricted to “occasional non-commercial distribution.” This subsection makes clear that a distributor cannot comply with the GPL merely by making object code available on a publicly-accessible network server accompanied by a copy of the written offer to provide source code received from an upstream distributor.

New subsection 6d, which revises the final paragraph of GPLv2 section 3, addresses distribution of object code by offering access to copy the code from a designated place, such as by enabling electronic access to a network server. Subsection 6d clarifies that the distributor must offer equivalent access to copy the source code “in the same way through the same place.” This wording permits a distributor to offer a third party access to both object code and source code on a single network portal or web page, even though the access may include links to different physical servers. For example, a downstream distributor may provide a link to an upstream distributor’s server and arrange with the operator of that server to keep the source code available for copying for as long as the downstream distributor enables access to the object code. This codifies what has been our interpretation of GPLv2.

The paragraph following subsection 6d expressly prevents a distributor of object code from purporting to satisfy his or her obligations under the GPL by providing source code in some private, locked, digitally-restricted, or other non-free form.

The final paragraph of section 6 takes account of the fact that the Complete Corresponding Source Code may include added parts that carry non-GPL terms, as permitted by section 7.

2.8 7. License Compatibility

In GPLv3 we take a new approach to the issue of combining GPL’d code with code governed by the terms of other free software licenses. Our view, though it was not explicitly stated in GPLv2 itself, was that GPLv2 allowed such combinations only if the non-GPL licensing terms permitted distribution under the GPL and imposed no restrictions on the code that were not also imposed by the GPL. In practice, we supplemented this policy with a structure of exceptions for certain kinds of combinations.

Section 7 of GPLv3 implements a more explicit policy on license compatibility. It formalizes the circumstances under which a licensee may release a covered work that includes an added part carrying non-GPL terms. We distinguish between terms that provide additional permissions, and terms that place additional requirements on the code, relative to the permissions and requirements established by applying the GPL to the code.

Section 7 first explicitly allows added parts covered by terms with additional permissions to be combined with GPL'd code. This codifies our existing practice of regarding such licensing terms as compatible with the GPL. A downstream user of a combined GPL'd work who modifies such an added part may remove the additional permissions, in which case the broader permissions no longer apply to the modified version, and only the terms of the GPL apply to it.

In its treatment of terms that impose additional requirements, section 7 extends the range of licensing terms with which the GPL is compatible. An added part carrying additional requirements may be combined with GPL'd code, but only if those requirements belong to an set enumerated in section 7. We must, of course, place some limit on the kinds of additional requirements that we will accept, to ensure that enhanced license compatibility does not defeat the broader freedoms advanced by the GPL. Unlike terms that grant additional permissions, terms that impose additional requirements cannot be removed by a downstream user of the combined GPL'd work, because no such user would have the right to do so.

Under subsections 7a and 7b, the requirements may include preservation of copyright notices, information about the origins of the code or alterations of the code, and different warranty disclaimers. Under subsection 7c, the requirements may include limitations on the use of names of contributors and on the use of trademarks for publicity purposes. In general, we permit these requirements in added terms because many free software licenses include them and we consider them to be unobjectionable. Because we support trademark fair use, the limitations on the use of trademarks may seek to enforce only what is required by trademark law, and may not prohibit what would constitute fair use.

Under subsection 7d, the added part may require the program to contain functioning facilities that allow users to obtain copies of the program's Complete Corresponding Source Code. This is intended to enable compatibility with licensing terms that, for example, require modified versions of a program that interacts with users through a network to preserve an opportunity for users to request network transmission of the source code.

Subsection 7e permits the additional requirements to include a software patent retaliation provision that may be broader than the narrow patent retaliation clause in section 2 of GPLv3. However, the retaliation is specifically limited to retaliation against one of two types of software patent infringement lawsuits: (1) lawsuits that constitute aggression because they are not brought in retaliation against software patent aggression, and (2) lawsuits that target the GPL'd code or the added part.

We have placed these limits on acceptable patent retaliation provisions for a number of reasons. We reject overbroad patent retaliation provisions that give undue advantage to the program's author. We also do not support overbroad patent retaliation provisions that would impose burdensome patent searching requirements on users. Furthermore, a patent retaliation provision ought not to punish those who have brought a patent infringement claim in defense against an act of patent aggression.

Section 7 notes that the GPL does not provide for enforcement of terms imposing additional requirements. Rather, section 7 simply establishes that a combination of code covered by such terms with GPL'd code does not violate the GPL. One reason why we do not seek to enforce such terms is that they are not written by us. In addition, we seek to encourage experimentation with different licensing terms, and therefore we do not wish to commit ourselves to any particular form of the additional requirements we permit under section 7.

Section 7 requires a downstream user of a covered work to preserve the non-GPL terms covering the added parts just as they must preserve the GPL, as long as any substantial portion of those parts is present in the user's version.

As we permit combinations of code covered by different licensing terms, we also aim to ensure that determining what terms apply to a particular section of code does not become a burdensome task for the developer. Section 7 accordingly requires that all non-GPL terms included in a combined GPL'd work be listed in a central place in the work.

However, as a special exception, this list is not required for works that are licensed under a previous version of the GPL in addition to GPLv3. We include this exception so that a distributor of a work licensed under GPLv2 "or any later version" that includes code carrying non-GPL licensing terms will not be in violation of GPLv3 merely by failing to provide the list.

2.9 8. Termination

GPLv2 provided for automatic termination of the rights of a person who copied, modified, sublicensed, or distributed a work in violation of the license. Automatic termination can be too harsh for those who have committed an inadvertent violation, particularly in cases involving distribution of large collections of software having numerous copyright holders. A violator who resumes compliance with GPLv2 would need to obtain forgiveness from all copyright holders, but even to contact them all might be impossible.

Section 8 of GPLv3 replaces automatic termination with a non-automatic termination process. Any copyright holder for the licensed work may opt to terminate the rights of a violator of the license, provided that the copyright holder has first given notice of the violation within 60 days of its most recent occurrence. A violator who has been given notice may make efforts to enter into compliance and may request that the copyright holder agree not exercise the right of termination; the copyright holder may choose to grant or refuse this request.

If a licensee who is in violation of GPLv3 acts to correct the violation and enter into compliance, and the licensee receives no notice of the past violation within 60 days, then the licensee need not worry about termination of rights under the license.

2.10 9. Not a Contract

Section 9 revises the corresponding section in GPLv2 in various ways to make the provision clearer.

2.11 10. Automatic Licensing of Downstream Users

Section 10 includes changes that provide consistency with new section 7.

2.12 11. Licensing of Patents

GPLv3 adds a new section on licensing of patents. GPLv2 relies on an implied patent license. The doctrine of implied license is one that is recognized under United States patent law but may not be recognized in other jurisdictions. We have therefore decided to make the patent license grant explicit in GPLv3. Under section 11, a redistributor of a GPL'd work automatically grants a nonexclusive, royalty-free and worldwide license for any patent claims held by the redistributor, if those claims would be infringed by the work or a reasonably contemplated use of the work.

The patent license is granted both to recipients of the redistributed work and to any other users who have received any version of the work. Section 11 therefore ensures that downstream users of GPL'd code and works derived from GPL'd code are protected from the threat of patent infringement allegations made by upstream distributors, regardless of which country's laws are held to apply to any particular aspect of the distribution or licensing of the GPL'd code.

A redistributor of GPL'd code may benefit from a patent license that has been granted by a third party, where the third party otherwise could bring a patent infringement lawsuit against the redistributor based on the distribution or other use of the code. In such a case, downstream users of the redistributed code generally remain vulnerable to the applicable patent claims of the third party. This threatens to defeat the purposes of the GPL, for the third party could prevent any downstream users from exercising the freedoms that the license seeks to guarantee.

The second paragraph of section 11 addresses this problem by requiring the redistributor to act to shield downstream users from these patent claims. The requirement applies only to those redistributors who distribute knowingly relying on a patent license. Many companies enter into blanket patent cross-licensing agreements. With respect to some such agreements, it would not be reasonable to expect a company to know that a particular patent license covered by the agreement, but not specifically mentioned in it, protects the company's distribution of GPL'd code.

2.13 12. Liberty or Death for the Program

The wording in the first sentence of section 12 has been revised slightly to clarify that an agreement, such as a litigation settlement agreement or a patent license agreement, is one of the ways in which conditions may be "imposed" on a GPL licensee that may contradict the conditions of the GPL, but which do not excuse the licensee from compliance with those conditions. This change codifies what has been our interpretation of GPLv2.

We have removed the limited severability clause of GPLv2 section 7 as a matter of tactical judgment, believing that this is the best way to ensure that all provisions of the GPL will be upheld in court. We have also removed the final sentence of GPLv2 section 7, which we consider to be unnecessary.

2.14 [13. Geographical Limitations]

To our knowledge, no one has invoked this section to add an explicit geographical distribution limitation since GPLv2 was released in 1991. We have concluded that this provision is not needed and is not expected to be needed in the future, and that it therefore should be removed. However, we invite members of the community who believe that this provision should be retained to provide us with their views.

2.15 14. Revised Versions of this License

No substantive change has been made in section 14. The wording of the section has been revised slightly to make it clearer.

2.16 15. Requesting Exceptions

No change has been made in section 15.

2.17 16–17. No Warranty

No substantive changes have been made in sections 16 and 17.

2.18 18. Safety Critical Systems

In safety-critical settings, software that does not perform correctly may expose developers to potentially high liability for damages. Some non-free licenses address this problem by prohibiting software from being used in safety-critical applications, a solution that is not appropriate for a free software license. New section 18 instead protects developers from liability by making clear that, by default, GPL'd software has not been tested for use in safety-critical systems. Users remain free to run and modify the software for any purpose.

Chapter 3

Conclusion

Our goal is for the GPLv3 public discussion process to be as transparent and accessible as possible. We look forward to receiving many informative and helpful comments on the draft of GPLv3 from all relevant viewpoints across the free software community. Mindful of the likely volume of comments, we have built a web-based interface for entering comments designed to anchor each submitted comment firmly to the discussion draft. We ask those who comment to keep in mind that we are engaged in the specialized work of drafting a license, rather than a more general exploration of philosophy and purpose.

As we explained in the GPL3 Process Definition document, available at `http://gplv3.fsf.org/process-definition`, we are forming discussion committees whose task will be to identify issues based on the public comments we receive. Final decisions on all issues will be made, accompanied by full reasons given by the Free Software Foundation.

In 1991 we released GPLv2 to apply it to our own software, and in the hope that other developers might wish to apply it to theirs. Today we are delighted that our software is being used by so many people around the world. We are just as pleased to see so many developers applying the GPL to their own programs. The process of revising the GPL affords an opportunity to everyone who uses, or wishes to use, the GPL to help us make it better. We invite all members of the community to join us in this effort.