

A VisionMobile independent research paper

GPLv2 vs GPLv3

The two seminal open source licenses,
their roots, consequences and repercussions

Analysing, understanding and interpreting the GPLv2 and GPLv3 licenses

September 2007

Liz Laffan MA(IPE)



Abstract

As of 2007, open source software is being used by almost 200,000 software projects, including rapidly growing use in consumer electronics and mobile phones. The vast majority of open source projects are licensed under the GPLv2, a highly influential license whose interpretation has been intensely debated. This paper analyses the underpinnings of open source culture that are embodied in GPLv2, dissects the terms and examines the basis of debates. The paper further examines the new terms and differences introduced by the new GPLv3 license and assesses the probable impact of GPLv3 on the software market in general and the mobile industry in particular.

Contents

- The formidable growth trajectory of open source
- Behind the scenes: understanding the cultural roots of open source licensing
- GPLv2: dissecting the fear, uncertainty and doubt
- GPLv3 under the hood: changes and repercussions
- GPLv2 vs GPLv3: differences and debates
- To upgrade to GPLv3 or not to ?
- Appendix: Comparative analysis of GPLv2 vs GPLv3

About the author

Liz Laffan | Senior Analyst

Liz has over 15 years experience in the Technology Industry. Her career portfolio comprises European Telecoms Co's, ISPs, Network Operators and start-up Technology companies. Liz has extensive commercial negotiation and vendor management experience coupled with specialist bespoke software licensing (proprietary and open source) knowledge. Liz has a BA Business Studies (Hons) and an MA in International Political Economy from Warwick University, with a specialisation on Trade Related Intellectual Property Rights.

About VisionMobile

VisionMobile Ltd. is a market know-how services firm delivering industry research and strategic advisory in the wireless sector. VisionMobile brings together passionate people with industry and technology know-how and innovative thinking.

www.visionmobile.com

info@visionmobile.com

Registered address: 84 Kirkland Avenue,
Clayhall, Ilford Essex IG5 0TN, UK
+44 (0) 845 003 8742

Disclaimer

VisionMobile believes the statements contained in this publication to be based upon information that we consider reliable, but we do not represent that it is accurate or complete and it should not be relied upon as such. Opinions expressed are current opinions as of the date appearing on this publication only and the information, including the opinions contained herein, are subject to change without notice.

Use of this publication by any third party for whatever purpose should not and does not, absolve such third party from using due diligence in verifying the publication's contents.

VisionMobile disclaims all implied warranties, including, without limitation, warranties of merchantability or fitness for a particular purpose. VisionMobile, its affiliates and representatives, shall have no liability for any direct, incidental, special, or consequential damages or lost profits, if any, suffered by any third party as a result of decisions made, or not made, or actions taken, or not taken, based on this publication.

This report is provided for informational purposes only. It does not constitute legal advice and should not be interpreted as such.

GPLv2 vs GPLv3

The two seminal open source licenses, their roots, consequences and repercussions

| The formidable growth trajectory of open source

In recent years the mobile software industry has witnessed the emergence of new business models, standards bodies, representative organisations, solutions, services, tools, blogs, podcasts, books and webcasts – all pertaining to the use of FOSS (Free and Open Source Software).

Since 2005 alone we have witnessed the emergence of new industry forums with the LiMo Foundation, the GMAE (Gnome Mobile and Embedded) initiative and the LiPS organisation, collectively backed by industry heavyweights such as Nokia, Motorola, Samsung, Panasonic, LG, Vodafone, Orange, NTT DoCoMo, Intel and Red Hat.

These industry groups have come together to encourage and further the use of FOSS as well as tackle fragmentation issues within mobile handset software. This sense of urgency follows a growing momentum with increased use of FOSS in mobile devices. Led by Motorola, Panasonic, NEC, Samsung, LG and, to some extent Nokia, handset manufacturers have launched over 50 Linux-based handset models as of mid 2007, shipping almost 12 million handsets in 2006 according to Informa Telecoms & Media. In parallel, Trolltech and FIC have launched the first mobile handsets that are majority based on FOSS.

Quietly supporting this growth has been the freedom and choice that FOSS-based software development allows to users, be they major software firms or independent developers. The success of FOSS adoption is due in a major part to the liberal and 'copyleft' licensing regimes perpetuated by FOSS, a detail that is often overlooked. The momentum of FOSS is further reinforced by the use of in the Linux kernel in consumer electronics, mobile phones and telecoms network equipment.

From a licensing perspective, the vast majority (typically 60-70%) of all open source projects are licensed under the GNU Public License version 2 (GPLv2). The license has been surrounded by much debate since it was created in 1991, due to its 'copyleft' properties, at the same time a key reason for its perpetuation. As of June 2007, the saga of open source licensing is entering a second wave with the introduction of GPLv3, a license which stands to have an immense impact on open source going forward. Indeed since GPLv3's publication to September 2007 nearly 600 FOSS projects have moved from GPLv2 to GPLv3 license.

This paper is set out in three parts: firstly we review the FOSS organisations and their role in the creation of FOSS licensing; secondly we discuss why GPLv2 has historically provoked such debate and how this has been regarded in the industry; and thirdly we summarise GPLv2 and GPLv3 and conclude with some observations on the future of GPLv2 and GPLv3 for FOSS in mobile technology.

Behind the scenes: understanding the cultural roots of open source licensing

The organisations behind FOSS have played a decisive role in the evolution of open source licensing. Probably most influential is the Free Software Foundation (FSF - www.fsf.org) which was founded by Richard Stallman in 1985. Stallman advocates four 'freedoms' pertaining to the use of software. These four freedoms shape the GPLv2 license, which Stallman authored:

1. *The freedom to run the program, for any purpose (freedom 0).*
2. *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.*
3. *The freedom to redistribute copies so you can help your neighbor (freedom 2).*
4. *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.*

In summary the FSF aims to maintain the freedom to run, copy, distribute, study, change and improve the software; this philosophy underpins both GPLv2 and GPLv3 licenses and should guide their interpretation.

The second most influential organisation is the Open Source Initiative (www.osi.org). The OSI was founded by Eric Raymond (author of the seminal book 'The Cathedral and the Bazaar') and Bruce Perens in 1998. The OSI is responsible for approving open source licenses. The OSI appears to have a more pragmatic philosophy compared to the FSF regarding users' freedoms as can be witnessed in the OSI definition. The OSI definition advocates free redistribution, access and distribution of source

code and ability to create derivative works, all of which are similar to the four freedoms of the FSF. However where the OSI and the FSF diverge is that the OSI also require that an OSI approved open source license should not restrict commercially important freedoms, such as the ability to distribute open source and non open source software together, and to not discriminate against any persons, field of endeavour or technology products. This last point is a particular area of differentiation with the FSF, as GPLv3 contains wording which appears to preclude use of GPLv3 covered code in specific technology areas.

Another influential initiative is the www.gpl-violations.org project, set-up in 2001 by Harald Welte and self-tasked to uphold the license conditions of the GPLv2. As of 2007, the initiative reports a number of successful legal cases brought against GPLv2 violators. These cases have primarily focussed on infringement of the GPLv2 license (whether intended or not) by software or hardware firms who distribute GPLv2-licensed code in their products without making the source code available or without including the GPLv2 license terms with their products.

As epitomised by gpl-violations.org, GPLv2 has been upheld and defended by the community – arguably to a greater extent than most proprietary software is either monitored or audited.

GPLv2: dissecting the fear, uncertainty and doubt

Published in 1991, GPLv2 is the license under which the majority of open source software and most notably the Linux kernel are distributed. The impact of GPLv2 is further reinforced with the ever increasing number of consumer electronics and mobile phones that embed the Linux kernel.

At the simplest level GPLv2 provides the user (whether commercial firms or individual developers) with permission to run the software internally, without ever making distributions of object code or source code, with minimal obligations. If the user does distribute code (for example as part of a software product or application) then they are obliged to distribute the source code and the source code of any 'derivative works' that have been created. The intent, as always, is to ensure the preservation of users' freedom to 'run, copy, distribute, study, change and improve the software', now and in the future.

GPLv2 imposes further obligations such as publication of notices of changes that have been made; provision of access to the source code for a minimum of 3 years; and automatic acceptance of the GPLv2 terms. Most importantly the license is immediately terminated if the user fails to abide by its terms.

Whilst these obligations appear relatively straightforward, interpreting the GPLv2 is a complex matter due to three main reasons: the ambiguity of many terms contained in the license, differing international copyright law and more recently due to patent developments in the software industry. These factors have led to a situation whereby there is considerable uncertainty relating to the exact obligations that must be satisfied under GPLv2 covered code.

Firstly we should remember that the GPLv2 was written in 1991 at a time when modern software licenses, contracts and software development practices were only in a nascent stage; it is easy to be critical in retrospect, particularly so given the wealth of understanding which we have today about software licensing.

The challenge of interpreting GPLv2 is exacerbated by the lack of formally 'defined terms' in GPLv2, particularly the lack of definition of the terms 'distribution' and 'derivative work'. One should also consider the fact that GPLv2

was written with US Copyright Law in mind, which cannot be used to interpret GPLv2 outside of the US. To appreciate the degree of ambiguity, consider the following extract from Section 0 of the GPLv2.

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".)

One might argue that this Section is confusing, possibly repetitive and probably unclear – and this has been a major issue with regard to use of GPLv2. The interpretation and application of these terms will determine what a user believes they can do with GPLv2 covered code.

A second core issue with interpreting GPLv2 is that of derivative works. A derivative work under GPLv2 – as worded in the previous extract – is actually a very broad description which has been interpreted differently by the FOSS community as well as by commercial users of GPLv2.

For example, it is generally agreed that a derivative work is produced if one creates new code based on existing GPL covered code (termed a modification). However is a derivative work created when one 'links' GPL covered code with non-GPL covered code? ('linking' being a standard practice of software development). The generally accepted best practice is that a derivative work is indeed created when statically linking GPL software to non-GPL software. However there is no clear agreement within the

FOSS community regarding dynamically linking of GPL software to non-GPL software.

Additionally as there has been no legal precedent supporting either understanding, it is unclear which interpretation should prevail. It is this uncertainty that leads most lawyers to err on the side of conservatism when determining what a derivative work is. However, one is permitted to aggregate (combine) non-GPL software works or Programs with the GPL works/Programs on a CD or similar without mandating the non-GPL software to be under the terms of the GPLv2.

Thirdly there is the question of patents. In 1991 patents were not perceived to be a software issue as patent portfolios did not exist within the software industry to the extent that they are prevalent today. The Diagram below illustrates this quite succinctly:

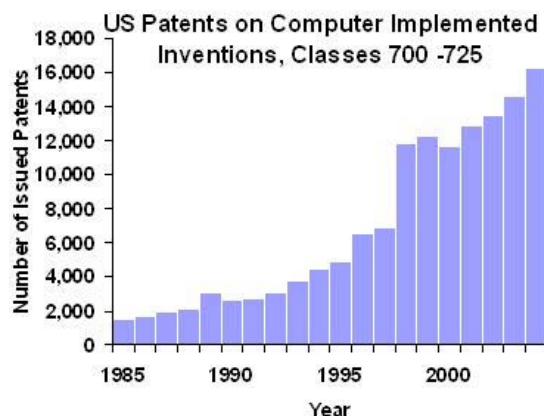


Figure licensed under GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation;
http://en.wikipedia.org/wiki/Image:Software_patents2.JPG#file

Cumulatively patents have been doubling practically every year since 1990. Patents are now probably the most contentious issue in software-related intellectual property rights. This paper does not intend to debate the merits or otherwise of patents here but suffice to say that with increased patent filings and awards both in Europe and the US, the risk has increased

substantially that patents are being infringed and probably unknowingly so.

However we should also be aware that software written from scratch is as likely to infringe patents as FOSS covered software – due mainly to the increasing proliferation of patents in all software technologies. Consequently the risk of patent infringement is largely comparable whether one chooses to write one's own software or use software covered by the GPLv2; one will most likely have to self-indemnify against a potential patent infringement claim in both cases. In addition there is still the issue of certainty of provenance (that is knowledge and confidence in whom has written the code) with regard to any third party code, probably more so with FOSS which does not generally come with warranties and indemnities. These factors must be weighed up against the likelihood of legal action for any such infringement and considered on a case-by-case basis.

The F.U.D. (fear, uncertainty and doubt) that surrounds patents in FOSS has been further heightened by two announcements, both instigated by Microsoft. Firstly in November 2006 Microsoft and Novell¹ entered into a cross-licensing patent agreement where Microsoft gave Novell assurances that it would not sue the company or its customers if they were to be found infringing Microsoft patents in the Novell Linux distribution. Secondly in May 2007 Microsoft² restated (having alluded to the same in 2004) that FOSS violates 235 Microsoft patents. Unfortunately, the Redmond giant did not state which patents in particular were being infringed and nor have they initiated any actions against a user or distributor of Linux.

Nonetheless these actions have served to create tension surrounding Microsoft's claims of patent infringement by FOSS.

The FOSS community have reacted to these actions by co-opting the patent system and setting up the Patent Commons

(<http://www.patentcommons.org>). This initiative, managed by the Linux Foundation, coordinates and manages a patent commons reference library, documenting information about patent-related pledges in support of Linux and FOSS that are provided by large software companies. Moreover, software giants such as IBM and Nokia have committed not to assert patents against the Linux kernel and other FOSS projects. In addition, the FSF have strengthened the patent clause of GPLv3, a topic we return to later in this paper.

As a final thought on this section it is useful to understand that historically there have been explicit reasons why certain organisations do not endorse with the FSF and OSI principles. These are organisations whose business model is primarily predicated on the ability to provide source code and then allow the code to be productised. Such firms are reluctant to use GPLv2 covered code for fear of potential cannibalisation of revenue streams could it be argued that their product is a derivative of GPLv2 covered code (which would require publication of source code on distribution of such product). Whilst this apprehension may appear somewhat exaggerated, it remains a strong concern within commercial, proprietary software firms.

Notwithstanding the above we note that a number of proprietary software firms are opening up access to their source code to some degree. Indeed in July 2007 Microsoft not only submitted two of their more permissive Licenses to the OSI for approval but they also launched an open source website³. In addition Microsoft has been experimenting with product-specific source code access under a so-called 'shared source' license scheme since 2001. In November 2006 the Redmond giant announced full Windows Embedded CE kernel source code access (albeit primarily for 'reference purposes') on reasonably liberal licensing terms. Some observers argue that this is a move to co-opt the FOSS

movement whilst others suggest that this move represents for Microsoft a tacit acknowledgement of the success of the FOSS movement and its impossible-to-ignore impact on the software industry.

GPLv3 under the hood: changes and repercussions

GPLv3 was published in July 2007, some 16 years following the creation of GPLv2. The purpose of this new license is to address some of the areas identified for improvement and clarification in GPLv2 – such as patent indemnity, internationalisation and remedies for inadvertent license infringement (rather than the previous immediate termination effect). The new GPLv3 license is nearly double the length of the GPLv2; such has been the fortitude to write a license which is more precise, clearer in language and ideally more consistently interpreted.

GPLv3 is written by Richard Stallman of the FSF and Eben Moglen of the Software Freedom Law Centre. To arrive at the GPLv3, the authors used a very broad, consensus-driven process, seeking feedback from four separate committees and broad comment over 18 months of public consultation:

- Committee A comprised mostly Free Software supporters and Projects such as Debian, Google, Samba, SleepyCat, Red Hat and others.
- Committee B included the erstwhile giants of the IT and software world such as IBM, HP, Sun Microsystems, Apple, Nokia, Intel and so on.
- Committee C constituted various academics, lawyers and activists in the public domain with an interest in the GPL. Last but not least
- Committee D comprised interested onlookers, programmers and licensing enthusiasts.

GPLv3 has been written in the same spirit and essence of GPLv2. The intent, as always, has been to ensure the preservation of users' freedom to 'run, copy, distribute, study, change and improve the software'.

I GPLv2 vs GPLv3: differences and debates

GPLv3 differs to GPLv2 in several important ways. Firstly it provides more clarity on patent licenses and attempts to clarify what is meant by both a distribution and derivative works. Secondly it revokes the immediate termination of license clause in favour of licensee opportunities to 'fix' any violations within a given time-period. In addition there are explicit 'Additional Terms' which permits users to choose from a fixed set of alternative terms which can modify the standard GPLv3 terms. These are all welcome, positive moves which should benefit all users of the GPLv3 license.

Nonetheless there are three contentious aspects of GPLv3 that have provoked much discussion in the FOSS community and could deter adoption of GPLv3 by more circumspect users and organisations.

Firstly patents. On the whole there is agreement that introducing an explicit patent provision is a strong benefit to users of GPLv3 covered software, but there is much concern regarding the actual wording that is used to provide this patent coverage. The wording has been criticised as too broad given that it covers 'future patents' that might be filed or invented with regard to a technology area similar to that which a user contributes or licenses under GPLv3.

Additionally there is a new term 'knowingly rely' which implies that if a user distributes code knowingly relying on a patent license and that source code which is patented is not available freely to all then the user must either make that

source code available, deprive itself from the benefits of that patent license or extend that patent license downstream. In trying to interpret these new concepts there are concerns about what specifically constitutes 'knowingly relying' and also the implied direct responsibility that the user has for ensuring that the source code patent is made available to others. These are just two of the concerns that have been identified to date. The complicated nature of these concepts will see many lawyers attempting to establish interpretation(s) in 2007.

Secondly, a new section has been added to the license under the heading "Protecting Users' Legal Rights from Anti-Circumvention Law". In a nutshell, this section is intended to prevent GPLv3-covered code from being included in technology or products that would be used to enforce the Digital Millennium Copyright Act (DMCA). The DMCA criminalizes production and dissemination of technology, devices, or services which are used to get around processes that control access to copyrighted works (commonly known as DRM). So what this constitutes is a constraint in the product type to which the code can be used – which many argue should not be the concern of FOSS licensing.

Thirdly and perhaps more expected is the GPLv3 section which attempts to deal with Tivoisation⁴. Tivoisation is a term named after the Tivo product, a digital video recorder and consumer device which allows users to capture television programming to an internal hard disk storage for viewing later – with which one can record many TV channels at once and watch them later (so-called time-shifting). Tivo contains a small Linux OS which under GPLv2 requires the hardware manufacturer to make the source code available to users – which Tivo does. Users can modify the GPLv2-covered source code and then compile it, but the software code won't run because Tivo contains a special mechanism which shuts down if it notices changes to the code. Therefore whilst

Tivo is fulfilling its' obligations as required under GPLv2, it is actually inhibiting the four freedoms as set-out by the FSF, that is "to preserve the users' freedom to run, copy, distribute, study, change and improve the software".

To prevent Tivoisation, GPLv3 introduces new terms and obligations. The new terms are 'User Product' and 'Installation Information' as quoted below:-

"A 'User Product' is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling."

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made."

These new terms are naturally intended to ensure that entities using GPLv3 licensed software for any user product also provide any and all additional information necessary to ensure installation and running of the software. This concept is of significance in that there is in most cases intellectual property and software 'know-how' contained in installation methods that may actually provide unique value to the entity using GPLv3 licensed software. Moreover, it appears that if a user or any other 3rd party has the ability to update GPLv3 covered software stored in the ROM then they trigger the obligation to provide source code.

It is unclear whether deploying mobile software management technologies such as FOTA

(firmware-over-the-air) updates forces the provision of access to the source code? (a use case that was unlikely to have been considered when the license wording was being agreed). Additionally there are potential product liability issues which may arise by giving users the ability to modify the code, not least the network operator concerns could such access lead to the possibility of the radio network being adversely affected.

Separately it can be argued that this effort to defeat Tivoisation may ultimately be overcome by technology in the form of new virtualisation software, as reported by Bruno Zappos⁵. Indeed this predicament in GPLv3 may initiate many unforeseen consequences.

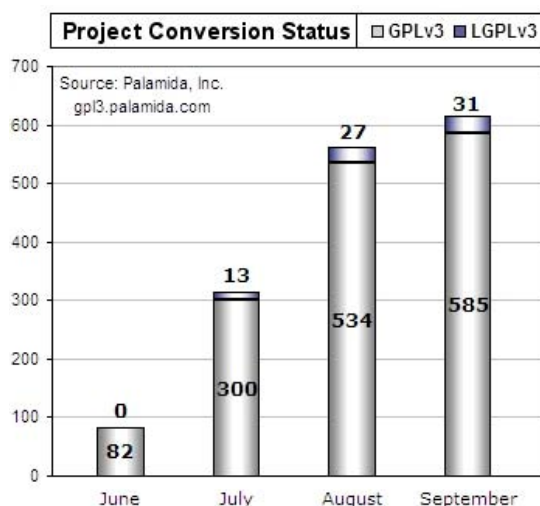
In summary these new concepts of 'control' in GPLv3 are probably the most controversial aspects of the license. Attempts to extend the terms of the license by mandating the uses of the code may be perceived by many to be beyond the remit of a license and ultimately may not support the aims of the broader FOSS movement.

| To upgrade to GPLv3 or not to?

GPLv2, by virtue of its privileged position as the license used by the Linux kernel, has been bestowed with great advantages. It has proven successful as the foundation for a firm majority of FOSS projects. Despite the lively legal discussions and debates surrounding GPLv2, it has proved to be by far and away the most accepted and used FOSS License.

But GPLv3 also has the potential to be a pervasive FOSS License. We may argue that the license seeks to extend its' remit beyond preserving users' freedoms but perhaps this is acknowledgement that one cannot truly separate rights and responsibilities from beliefs and philosophies.

Already, a significant number of GPLv2 licensed projects have already moved over to GPLv3 license, according to Palamida, a US based company which provides tools for managing software IPR.



Extracted from Palamida GPLv3 and LGPLv3 website on 13th September 2007
(<http://gpl3.palamida.com:8080/index.jsp>)

As of September 2007, nearly 600 projects of the 5000+ active projects listed on Sourceforge (the prevalent open source project repository) and licensed under GPLv2 or later have moved over to GPLv3. This is a transfer rate of the order of 10% - not bad if one considers that the License is, at the time of publication of this paper, just over two months old.

Furthermore if a FOSS Project were to be started post GPLv3 publication, and all other things being equal, it would probably be more advantageous to choose GPLv3 over GPLv2 for the additional patent protection it provides. We would argue that this practical advantage outweighs other more intangible concerns as understood in these early days of GPLv3 interpretations and understanding, notwithstanding the fact that the full implications of the anti-tivoisation section are as yet

unknown). Indeed there has been further support for GPLv3 with the OSI licensing board officially approving GPLv3 in September 2007⁶. Given that the OSI has now approved the license we can expect this to encourage further use and take-up of the license.

To conclude, referenced below is a quote from Linus Torvalds, the creator of the Linux kernel who was interviewed post GPLv3 publication in August 2007 by EFY Times.

*"I think it is much improved over the early drafts, and I don't think it's a horrible licence. I just don't think it's the same kind of 'great' licence that the GPLv2 is. So in the absence of the GPLv2, I could see myself using the GPLv3. But since I have a better choice, why should I? That said, I try to always be pragmatic, and the fact that I think the GPLv3 is not as good a licence as the GPLv2 is not a 'black and white' question. It's a balancing act. And if there are other advantages to the GPLv3, maybe those other advantages would be big enough to tilt the balance in favour of the GPLv3."*⁷

GPLv2 and GPLv3 Comparative Tables

The following tables are intended to help simplify and illustrate the GPLv2 and GPLv3 licenses. We have illustrated the obligations, responsibilities and differences across the licenses. The reader should note that this report and its contents are provided for informational purposes only. The tables do not constitute legal advice and do not absolve the reader from undertaking their own license due diligence and taking formal legal advice on the use of either GPL license.

Appendix

Comparative analysis of GPL2 vs GPLv3

GPLv2 vs GPLv3: Comparative analysis of license contents

This table is provided for informational purposes only. It does not constitute legal advice and should not be interpreted as such.

	GPLv2	GPLv3
License Grant – what can I do?	run the unmodified Program and §1 copy and distribute verbatim copies of the Program's source code (unmodified)	§2 make, run and propagate covered works that you do not convey (unmodified)
But if I change (modify) the code then:-	§1 modify your copy or copies of the Program or any portion PROVIDED A. Notices B. same terms as this License C. Print or display copyright notice	§4 convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code PROVIDED A. Notices B. same terms as this License (excepting Clause 7) C. Print or display copyright notice
However to be clear:-	§2 Above is not applicable to separate or independent works. Aggregation of 'other works' with the Program does not bring the other work under the scope of this License	§5 Above is not applicable to separate or independent works Aggregation (and compilation) of 'other works' with the Program does not bring the other work under the scope of this License
If I distribute (convey) Object code of the Program (or modifications) then I must :-	A. Accompany the object code with source code B. Accompany it with a written offer, valid for at least three years, to provide source code (at no more cost than the cost of physically performing source distribution) [source code means associated interface definition files, plus the scripts used to control compilation and installation of the executable]	A. Convey the object code in a physical product accompanied by the Corresponding Source B. Convey the object code with a written offer, valid for at least three years (or as long as you provide customer support for that Product), to provide Corresponding Source on a physical medium, for a price no more than your reasonable cost of physically performing this conveying of source, or to access to provide the Corresponding Source from a network server at no charge C. Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge

	GPLv2	GPLv3
Do I get a Patent License?	Preamble contains wording regarding Patents i.e. "we have made it clear that any patent must be licensed for everyone's free use or not licensed at all" – Implied Patent License (but contested by many)	§11 Explicit Patent License provided, see new GPLv3 Terms for full details.
What happens if I fail to fulfil my obligations around distribution of source code or try to do anything else not covered by this License?	§4 You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise will automatically terminate your rights under this License (doesn't impact downstream recipients)	§8 You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise will automatically terminate your rights under this License (including patent). BUT If you are advised that you have violated the License and you fix this violation within 30 days (or 60 days) your License rights are reinstated.
But I haven't actually accepted these Terms!	§5 You are not required to accept this License but nothing else grants you permission to modify or distribute the Program - by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License.	§9 You are not required to accept this License in order to receive or run a copy of the Program. Nothing other than this License grants you permission to propagate or modify any covered work. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.
When I redistribute (convey) the Program, can I change the terms of the License?	§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.	§10 Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You may not a. Impose any further restrictions on the exercise of the rights granted or affirmed under this License (subject to the Additional Terms below) b. Impose a license fee, royalty, or other charge for exercise of rights under this License c. initiate litigation (including a cross-claim or counterclaim) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it. BUT You can add those 'Additional Terms' as outlined in the next section 'New in GPLv3'
What if I am forced to accept conditions other than those agreed to in this License?	§7 If, as a consequence of a court judgment or allegation of patent infringement, conditions are imposed on you 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 you may not distribute the Program at all.	§12 If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all (no surrender of others' freedom.)

	GPLv2	GPLv3
What if other national Laws on copyright or patents	§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 may add an explicit geographical distribution limitation excluding those countries.	
Does the GPL License get upgraded? And if so what happens to my License?	§9 The FSF may publish revised versions of the GPL. 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.	§14The FSF may publish revised versions of the GPL. Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation
Do I get a Warranty or Indemnity?	§11 & 12 No, GPL terms normally come without Warranty or Indemnity (this does not preclude others from providing the same in return for a fee)	§15 & 16 As per GPLv2 and but exceptions are provided as per the Additional Terms (see new GPL Terms)

New terms introduced in GPLv3

This table is provided for informational purposes only. It does not constitute legal advice and should not be interpreted as such.

Terms in GPLv3 and not in GPLv2	
New Definitions	This License ⁸ , Copyright ⁹ , The Program ¹⁰ , Licensee & Licensees ¹¹ , Modify ¹² , Covered Work ¹³ , Propagate ¹⁴ , Convey ¹⁵ , Appropriate Legal Notices ¹⁶ , source code & object code ¹⁷ , Standard Interfaces ¹⁸ , Systems Libraries & Major Components ¹⁹ , Corresponding Source ²⁰
Section 3 Protecting Users' Legal Rights from Anti-Circumvention Law	No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures. Anti-Circumvention Laws are those laws that have been enacted in the US and Europe (via WIPO and the DMCA) which prohibit the circumvention (i.e. avoid, get around, hack perhaps?) of technological barriers that prevent the copying of intellectual property. To this extent GPLv3 attempts to prohibit the use of GPLv3 covered software in products and/or technology that would then be used support prevention of copying (or reengineering) of software.
Section 6 – User Product and Installation Information (Anti-Tivoisation)	Briefly the intent here is to prevent a Licensee from creating a 'User Product' and prohibiting the User from successfully running the modified source code by providing source code but tweaking the User Product such that it no longer runs if the source code has been changed. Ultimately the intent here is to preclude Tivoisation (see my blog posting on this specific issue with regard to GPLv3 and what I think could be some unintended consequences of this matter). Source code conveyed under this section must be accompanied by the Installation Information [not required if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM)]. Source code and Installation Information must be in a format that is publicly documented and must require no special password or key for unpacking, reading or copying. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network. New Defined Terms:- User Product²¹ Installation Information²²
Section 7 Additional Terms	Again this is a new and distinct topic for the GPL License. The purpose of 'Additional permissions' is to provide Licensees with the ability to make 'exceptions' to certain parts of the GPLv3 License, but you can obviously only do so to the work that you have 'created' or alternatively by obtaining the permission of the original copyright holders. These exceptions fall into 6 categories, namely:- - Amending or disclaiming the warranty/liability differently to that stated in the GPLv3 License - Requiring preservation of specific legal notices and/or author attributes - Preventing misrepresentation of the origin of the material or requiring modified versions to be marked in a particular way - Limiting the use for publicity purposes of names of Licensors/authors of material - Declining to grant rights under trademark law - Requiring indemnification of Licensors/authors Any and all other terms are not allowed and are titled 'further restrictions' which you as a Licensee on receipt of the GPLv3 License can remove. If you do choose to use any of these 'additional terms' then you must put a notice in the source code files indicating the same.

Terms in GPLv3 and not in GPLv2 (continued)

Section 11 Patents	Explicit patent license grant with new defined terms. Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version. If you convey a covered work, knowingly relying on a patent license (you have knowledge that your covered work would infringe one or more patents) then you must ensure that the Corresponding Source of the work is available for anyone to copy, free of charge and under the terms of this License or you must either stop using that particular work or extend the patent license to downstream recipients. A patent license is "discriminatory" if it does not include all of the rights that are specifically granted under this License. You may not convey a covered work if you in an arrangement with a third party under which you make payment to that third party based on the extent of your activity of conveying the work, and if that third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007. New Defined Terms:- Contributor²³, contributor's Essential Patent Claims²⁴, Patent License²⁵ and to Grant²⁶.
Section 13. Use with the GNU Affero GPL	You can link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

End notes

The following are provided for informational purposes only and should not be interpreted as legal advice.

¹ Guth, Robert A and Clark, Don; 3rd Nov 2006, "Microsoft, Novell Reach Accord on Linux ", Wall Street Journal http://online.wsj.com/public/article/SB116249026689311557-helTbrheLKgbaJ5iO5z40ZFCiOs_20061109.html?mod=blogs

² Parloff, Robert; 14th May 2007, "Microsoft takes on the free world", Fortune Magazine http://money.cnn.com/magazines/fortune/fortune_archive/2007/05/28/100033867/index.htm?postversion=2007051409

³ Galli, Peter; 10th August 2007, "Microsoft submits two Licenses for OSI Approval", E-Week.com, <http://www.eweek.com/article2/0,1759,2169433,00.asp>

⁴ for an analysis of GPLv3 and Tivoisation the reader is referred to Laffan, Liz; 26th July 2007, "GPLv3: Is is anti-Tivoisation flawed?", www.visionmobile.com <http://visionmobile.com/blog/2007/07/gplv3-is-anti-tivoisation-flawed/>

⁵ Zoppos, Bruno; 27th August 2007, "How hypervisors can defeat GPLv3's "anti-tivoization" ", Linuxdevices.com, <http://linuxdevices.com/articles/AT3899346728.html>

⁶ Aslett, Matthew; 10th September 2007, "GPLv3 is officially open source", Computer Business Review online, http://www.cbronline.com/article_news.asp?guid=A4BE06FA-EF8F-4D1E-8C62-B3DEF66F893F

⁷ Bhartiya Swapnil, August 2007, "LinusTorvalds: "Linux Is Much Bigger Than Me!" ", EfyTimes.com, <http://www.efytimes.com/archive/147/news.htm>

⁸ "This License" refers to version 3 of the GNU General Public License.

⁹ "Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

¹⁰ "The Program" refers to any copyrightable work licensed under this License.

¹¹ Each licensee is addressed as "you", "Licensees" and "recipients" may be individuals or organizations.

¹² To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

¹³ A "covered work" means either the unmodified Program or a work based on the Program.

¹⁴ To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution with or without modification), making (available to the public, and in some countries other activities as well.

¹⁵ To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying

¹⁶ An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface

presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

¹⁷ The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

¹⁸ A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

¹⁹ The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

²⁰ The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work. The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source. The Corresponding Source for a work in source code form is that same work.

²¹ A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

²² "Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

²³ A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

²⁴ A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

²⁵ A "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement).

²⁶ To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.



Knowledge. Passion. Innovation.

