

License compatibility

From Wikipedia, the free encyclopedia

License compatibility is an issue that arises when licenses applied to copyrighted works, particularly licenses of software packages (software source code as also binary representation^[1]). Licenses can contain contradictory requirements, rendering it impossible to combine source code or content from such works in order to create new ones.^{[2][3]}

Contents

- 1 Example
 - 2 Definitions
- 2.1 Kinds of combined works
 - 3 Compatibility of FOSS licenses
- 3.1 GPL compatibility
 - 3.1.1 Copyleft licenses and GPL
 - 3.1.2 GFDL and GPL
 - 3.1.3 CDDL and GPL
 - 3.1.4 CC BY-SA and GPLv3
 - 3.2 Creative Commons license compatibility
 - 4 Re-licensing for compatibility
 - 4.1 Relicensing examples
 - 5 See also
 - 6 References
 - 7 External links

Example

Suppose a software package has a license that says, "*modified versions must mention the developers in any advertising materials,*" and another package's license says "*modified versions cannot contain additional attribution requirements.*" Without direct permission from the copyright holder(s) for *at least* one of the two packages, it would be impossible to legally distribute a combination of the two because these specific license requirements cannot be simultaneously fulfilled. Thus, these two packages would be license-incompatible.

Definitions

License compatibility can be defined differently on with varying strength around the concepts of "collective/combined/aggregated work" and "derivative work". The first "collective work" license compatibility definition allows the usage of various licensed works in a combined context.

"License compatibility: The characteristic of two (or more) licenses according to which the codes distributed under these licenses may be put together in order **to create a bigger distributable software**."

— Philippe Laurent, European Open source Lawyers Event 2008, ^[4]

A stronger definition, also used by the FSF,^[5] includes the capability to change the license. As most prominent example, the copyleft licenses demand that the "derived work" combined from code under various licenses as whole is applied under the copyleft license.

"License compatibility: The characteristic of a license according to which the code distributed under this license may be integrated in a bigger software that will be **distributed under another license**."

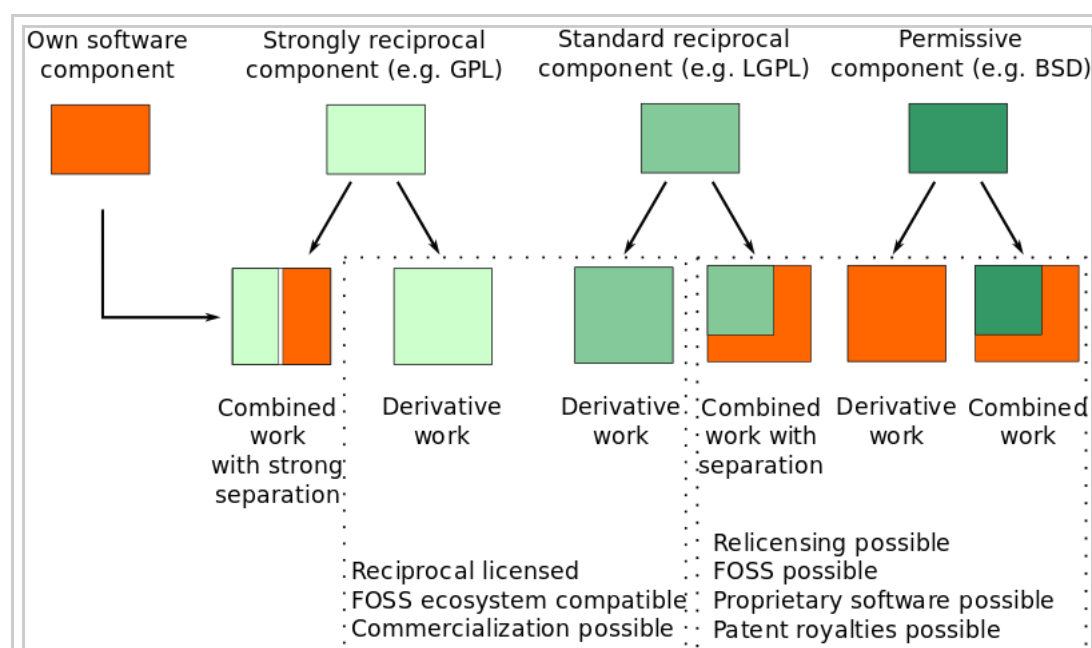
— Philippe Laurent, European Open source Lawyers Event 2008, ^[6]

Kinds of combined works

A "combined work" consists of multiple differently licensed parts (avoiding relicensing). To achieve a combined work including also copyleft licensed components (which have a viral property leading potentially to a "derived work") proper isolation/separation needs to be applied.

With individually licensed source code files multiple non-reciprocal licenses (as permissive licenses or own proprietary code) can be separated, while the combined compiled program could be relicensed (but is not required). Such source code file separation is too weak for copyleft/reciprocal licenses (as the GPL), as they require then the complete work to be relicensed under the reciprocal license as derivative.

A slightly stronger approach is a separation on the linking stage with binary object code (statical linking) with all the components living in the resulting program in the same process and address space. This satisfies "weak copyleft/standard reciprocal" combined works (as LGPL



License compatibility: Compatibility for derived works and combined works of own code and external developed open source licensed code (adapted from Mikko 2005).^[7] "Derived works" are mixing code, "combined works" have components under differing license. From right to left needs a combined work a stronger separation between differently licensed parts to prevent them in becoming a derived work ("separation": combined by statically linking or source file separation with all components living in the same process and address space, "strong separation": components connected only via binary interfaces and which live in separated processes).^[8]

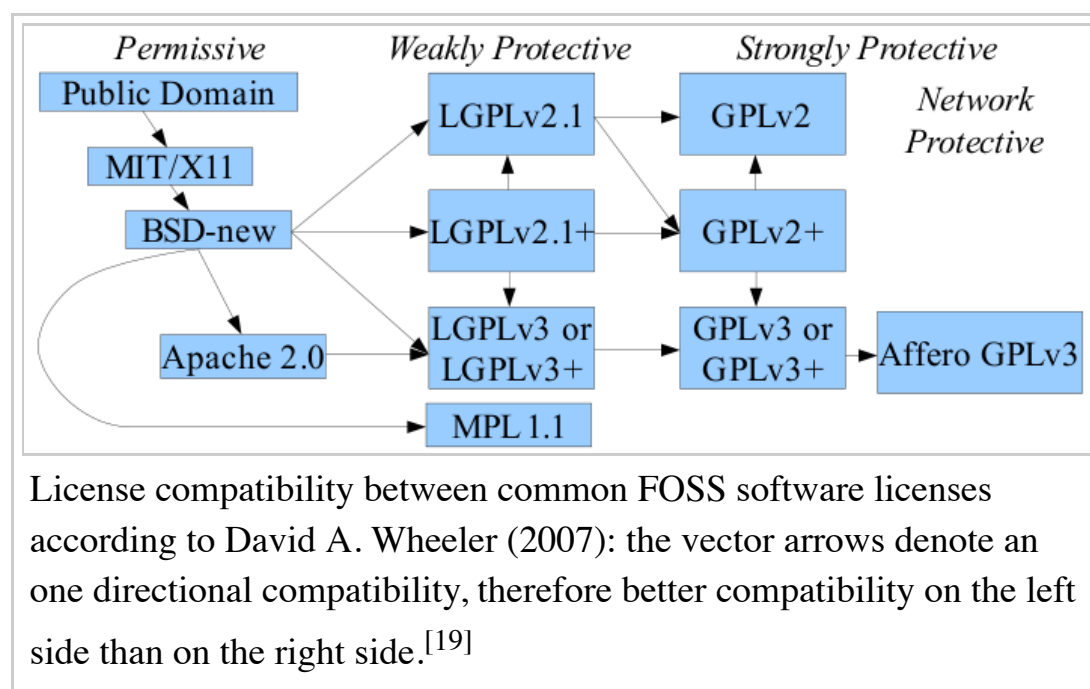
licensed ones), but not "strong copyleft/strong reciprocal" combined works. While commonly accepted that linking (static and even dynamic linking) constitutes a derivative of a strong copyleft'd work,^{[9][10][11][12][13]} there are individual alternative interpretations.^{[14][15]}

For "combined works" with "strong copyleft" modules, a stronger isolation is required. This can be achieved for instance by separating the programs by an own process and only communication via binary ABIs or other indirect means.^[10] Examples are Android's kernel space-to-user space separation via Bionic, or Linux distros which have proprietary binary blobs included despite having a strong copyleft kernel.^{[8][16]}

While for some domains agreement exist if an isolation is suitable, there are domains in dispute and up to now untested in court. For instance, in 2015 the SFC sued VMware in an ongoing dispute whether loadable kernel modules (LKMs) are derivative works of the GPL'ed Linux kernel or not.^{[17][18]}

Compatibility of FOSS licenses

Even accepted and common Open-source licenses are not necessarily compatible between each others.^[20] Which can make it legally impossible to mix (or link) even open-source code if the components are available under different licenses. For example, software that combined code released under version 1.1 of the Mozilla Public License (MPL) with code under the GNU General Public License (GPL) could not be distributed without violating one of the licenses' terms by default.^[21] This is despite both licenses being approved by the Open Source Initiative^[22] and Free Software Foundation.^[23] License compatibility between a copyleft license and another license is often only a one-way compatibility.^[24] This characteristic makes the copyleft GPL (and most other copyleft licenses) incompatible with proprietary commercial licenses as also many non-proprietary licenses.^{[25][26]} This "one-way compatibility" characteristic is for instanced criticized by the Apache Foundation, who provides the more permissive Apache license which doesn't have this characteristic.^[27] Non-copyleft licenses as the FOSS permissive licenses have a less complicated license interaction and have most often a in general better license compatibility.^{[28][29]}



GPL compatibility

To minimize license proliferation and license incompatibilities in the FOSS ecosystem, some organizations (for instance the FSF) and individuals (for instance David A. Wheeler), argue that compatibility to the widely used GPL is an important feature of software licenses.^[30] Many of the most common free software licenses, especially the permissive licenses, such as the original MIT/X license, BSD licenses (in the three-clause and two-clause forms, though not the original four-clause form), MPL 2.0, and LGPL, are "GPL-compatible". That is, their code can be combined with a program under the GPL without conflict and the new combination would have the GPL applied to the whole (not the other license).

Copyleft licenses and GPL

When it comes to copyleft software licenses, they are not inherently GPL-compatible, even the GPLv2 is, by itself, not compatible with GPLv3.^{[31][32][33][25]} If you tried to combine code released under both these licenses, you would violate section 6 of GPLv2, resulting in the incompatibility. However, if code is released under GPL “version 2 or later,” that is compatible with GPLv3 because GPLv3 is one of the options it permits.^[34] However, most software released under GPLv2 allows you to use the terms of later versions of the GPL as well but some have exception clauses that allow combining them with software that is under different licenses or license versions.^[35]

GFDL and GPL

Also, the FSF recommended GNU Free Documentation License^[36] is incompatible with the GPL, text licensed under the GFDL cannot be incorporated into GPL software. Therefore, the Debian project decided in a 2006 resolution to use for documentation the GPL.^[37] The FLOSS Manuals foundation followed Debian in 2007.^[38] In 2009 the Wikimedia Foundation switched also from the GFDL to a CC-BY-SA license as main license for their projects.^{[39][40]}

CDDL and GPL

Another important and controversial debated case of GPL compatibility is the CDDL licensed ZFS file system with the GPLv2 licensed Linux kernel.^[41] Despite that both are free software under a copyleft license, ZFS is not distributed with most linux distros like Debian^{[42][43]} (but with FreeBSD and even MacOS) as the CDDL is considered incompatible to the GPL'ed linux kernel by the FSF and most other FOSS parties.^{[44][45]} There exist also alternative positions as the legal interpretation, if and when this combination constitutes a "combined work" or "derivated work" of the GPL'ed kernel, is ambiguous and controversial.^[46] In 2015 the CDDL to GPL compatibility question reemerged when the linux distribution Ubuntu announced to include OpenZFS by default.^[47] In 2016 Ubuntu announced that a legal review resulted in the conclusion that it is legal safe to use ZFS as binary kernel module in linux.^[48] Others followed Ubuntu's conclusion, for instance lawyer James E.J. Bottomley argued there can't be "*a convincing theory of harm*" developed making it impossible to bring the case to court.^[49] Eben Moglen, co-author of the GPLv3 and founder of the SFLC, argues that while the letters of the GPL might be violated the spirit of both licenses is unharmed, which would be the relevant aspect in court.^[50] On the other hand, Bradley M. Kuhn and Karen M. Sandler from the Software Freedom Conservancy argued that Ubuntu would violate both licenses as a binary ZFS module would be a derivative work of the linux kernel and announced their will to achieve clarity in this question, even by court.^{[51][52]}

CC BY-SA and GPLv3

On October 8, 2015 Creative Commons concluded that the CC BY-SA 4.0 is one-way compatible with the GPLv3.^[53]

Creative Commons license compatibility

The Creative Commons Licenses are widely used for content, but not all combinations of the seven recommended and supported licenses are compatible among each other. Also, this is often an one directional compatibility, requiring the complete work to be licensed under the more restrictive license of both.

Re-licensing for compatibility

Sometimes projects gets stuck in a license incompatibility situation and the only feasible way to solve it is the re-licensing of the incompatible parts. Relicensing is achieved by contacting all involved developers and parties and getting their agreement for the changed license. While in the free and open-source domain achieving 100% coverage is often impossible, due to the many contributors involved, the Mozilla relicensing project assumes achieving 95% is enough for the relicensing of the complete code base.^[54] Others in the FOSS domain, as Eric S. Raymond, came to different conclusions regarding the requirements for relicensing of a whole code base.^[55]

Relicensing examples





































An early example of a project who did successful relicensing for license compatibility reason is the Mozilla project and their Firefox browser. The source code of Netscape's Communicator 4.0 browser was originally released in 1998 under the Netscape Public License/Mozilla Public License^[56] but was criticised by the FSF and OSI for being incompatible.^{[57][58]} Around 2001 Time Warner, exercising its rights under the Netscape Public License, and at the request of the Mozilla Foundation, relicensed^[59] all code in Mozilla that was under the Netscape Public License (including code by other contributors) to an MPL 1.1/GPL 2.0/LGPL 2.1 tri-license, thus achieving GPL-compatibility.^[60]

The Vorbis library was originally licensed as LGPL, but in 2001 the license was changed to the BSD license with endorsement of Richard Stallman to encourage adoption.^{[61][62]}

The VLC project has also a complicated license history due to license compatibility: in 2007 it decided for license compatibility reasons to not upgrade to the just released GPLv3.^[63] After the VLC was removed from Apple App Store in begin of 2011, in October 2011 the VLC project re-licensed the VLC library part from the GPLv2 to the LGPLv2 to achieve better compatibility.^{[64][65]} In July 2013 the VLC application could then resubmitted to the iOS App Store relicensed under the Mozilla Public License.^[66]

7-Zip's LZMA SDK, originally dual-licensed under both the GNU LGPL and Common Public License,^[67] with an additional special exception for linked binaries, was placed by Igor Pavlov in the public domain on December 2, 2008.^[68]

Another example is the GNU TLS project, which adopted in 2011 the LGPLv3 license but relicensed in 2013 their code back to LGPLv2.1 due to serious license compatibility problems.^{[69][70][71]}

	 PUBLIC DOMAIN	 PUBLIC DOMAIN	 CC BY	 CC BY-SA	 CC BY-NC-SA	 CC BY-NC-ND
 PUBLIC DOMAIN						
 CC BY						
 CC BY-SA						
 CC BY-NC-SA						
 CC BY-NC-ND						

A license compatibility chart for combining or mixing two CC licensed works.

The GNU Free Documentation License in version 1.2 is not compatible with the widely used Creative Commons Attribution-ShareAlike license, which was a problem for instance for the Wikipedia.^[72] Therefore, at the request of the Wikimedia Foundation the FSF added with version 1.3 of the GFDL a time-limited section allowing specific types of websites using the GFDL to additionally offer their work under the CC BY-SA license.^[73] Following in June 2009, the Wikimedia Foundation migrated their projects (Wikipedia, etc.) by dual licensing to the Creative Commons Attribution-ShareAlike as main license, additional to the previously used GFDL.^[39] An improved license compatibility with the greater free content ecosystem was given as reason for the license change.^{[40][74]}

Another interesting case was the relicensing of GPLv2 licensed linux kernel header files to the BSD license by Google for their Android library Bionic. To get rid of the GPL, Google claimed that the header files were cleaned from any copyright-able work, reducing them to non-copyrightable "facts".^{[75][76]} This interpretation was challenged for instance by Raymond Nimmer, a law professor at the University of Houston Law Center.^[77]

In 2014 the FreeCAD project changed their license from GPL to LGPLv2 due to GPLv3/GPLv2 incompatibilities.^{[78][79]}

Also the Dolphin project changed their license from "GPLv2 only" to "GPLv2 or any later" for better compatibility in May 2015.^[80]

In June 2015 mpv started the relicensing process of the project's source code for improved license compatibility under LGPLv2 by getting consent from the majority (95%+) of the contributing developers.^[81]

In July 2015 Seafile switched for improved license compatibility, especially with Git, from the GPLv3 to the GPLv2.^{[82][83]}

In 2016 MAME achieved a relicensing of the code base to BSD/GPL^[84] after struggling for years with an own written custom license, with non-commercial license terms.^{[85][86][87][88]}

See also

- License proliferation
- Comparison of free and open-source software licenses (also license compatibility)
- Backward compatibility
- Forward compatibility

References

1. Hancock, Terry (2008-08-29). "What if copyright didn't apply to binary executables?". Free Software Magazine. Retrieved 2016-01-25.
2. O'Riordan, Ciaran (2006-11-10). "How GPLv3 tackles license proliferation". linuxdevices.com. Archived from the original on 2007-12-18.
3. Neary, Dave (February 15, 2012). "Gray areas in software licensing". lwn.net. Retrieved 2016-02-27.
4. Laurent, Philippe (2008-09-24). "The GPLv3 and compatibility issues" (pdf). *European Open source Lawyers Event 2008*. University of Namur – Belgium. p. 3. Retrieved 2015-05-30. *"The characteristic of two (or more) licenses according to which the codes distributed under these licenses may be put together in order to create a bigger distributable software"*

istributable software.

5. "Stallman explains license compatibility while discussing GPLv3". fsfe.org. Retrieved 2015-12-03.
6. Laurent, Philippe (2008-09-24). "The GPLv3 and compatibility issues" (pdf). *European Open source Lawyers Event 2008*. University of Namur – Belgium. p. 4. Retrieved 2015-05-30. *"The characteristic of a licence according to which the code distributed under this license may be integrated in a bigger software that will be distributed under another license"*
7. Välimäki, Mikko (2005). "The Rise of Open Source Licensing: A Challenge to the Use of Intellectual Property in the Software industry". Turre Publishing. p. 119.
8. Larry Troan (2005). "Open Source from a Proprietary Perspective" (pdf). *RedHat Summit 2006 Nashville*. redhat.com. Retrieved 2015-12-29.
9. *Lewis Galoob Toys, Inc. v. Nintendo of America, Inc.*, 964 F.2d 965 (<http://law.justia.com/cases/federal/appellate-courts/F2/964/965/341457/>), ¶10 (9th Cir. 21 May 1992).
10. MereAggregation (<http://www.gnu.org/licenses/old-licenses/gpl-2.0-faq.html#MereAggregation>) *"What constitutes combining two parts into one program? This is a legal question, which ultimately judges will decide. We believe that a proper criterion depends both on the mechanism of communication [...] and the semantics of the communication [...]. If the modules are included in the same executable file, they are definitely combined in one program. If modules are designed to run linked together in a shared address space, that almost surely means combining them into one program. [...]"* on gnu.org
11. License Compatibility Matrix (<https://www.gnu.org/licenses/gpl-faq.html#AllCompatibility>) *"Use a library[...]
linking"* on gnu.org (same compatibility of source code use and library use)
12. LGPL (<https://www.gnu.org/licenses/gpl-faq.html#LGPLv3ContributorVersion>) on GNU.org
13. origins-lgpl (<https://www.freebsd.org/doc/en/articles/bsdl-gpl/origins-lgpl.html>) *"Remember that the GPL requires anything that statically links to any code under the GPL also be placed under the GPL."* on freebsd.org
14. Linus Torvalds, GPL only modules (<https://lkml.org/lkml/2006/12/17/79>), *linux-kernel* mailing list (17 December 2006).
15. Lawrence Rosen, Derivative Works (<http://www.linuxjournal.com/article/6366>), *Linux Journal* (1 January 2003).
16. Välimäki, Mikko (2005). "The Rise of Open Source Licensing: A Challenge to the Use of Intellectual Property in the Software industry". Turre Publishing. Retrieved 2015-12-30.
17. New Lawsuit Targets “Shims” Between Linux and Proprietary Code (<http://torekeland.com/blog/new-lawsuit-targets-shims-between-linux-and-proprietary-code>) Tor Ekeland (2015)
18. Conservancy Announces Funding for GPL Compliance Lawsuit (<http://sfconservancy.org/news/2015/mar/05/vmware-lawsuit/>) on sfconservancy.org (March 5, 2015)
19. The Free-Libre / Open Source Software (FLOSS) License Slide (<http://www.dwheeler.com/essays/floss-license-slide.html>) by David A. Wheeler on September 27, 2007
20. Thomas F. Gordon Open Source license compatibility (<http://www.osscc.net/pdf/QualipsoA1D113.pdf>) Fraunhofer FOKUS Berlin in December 2009.
21. "MPL 1.1 FAQ - Historical Use Only". Mozilla Foundation. 1 February 2012. Retrieved 26 February 2012.
22. "Mozilla Public License 1.1: Licensing". Open Source Initiative. Retrieved 26 February 2012.
23. "GPL-Incompatible Free Software Licenses". Free Software Foundation. 22 February 2012. Retrieved 26 February 2012.
24. Laurent, Philippe (2008-09-24). "The GPLv3 and compatibility issues" (pdf). *European Open source Lawyers Event 2008*. University of Namur – Belgium. p. 7. Retrieved 2015-05-30. *"Copyleft is the main source of compatibility problems"*
25. Nikolai Bezroukov (2000). "Comparative merits of GPL, BSD and Artistic licences (Critique of Viral Nature of GPL v.2 - or In Defense of Dual Licensing Idea)". Archived from the original on 2001-12-22. *"Viral property stimulates proliferation of licenses and contributes to the "GPL-enforced nightmare" -- a situation when many other licenses are logically incompatible with the GPL and make life unnecessary difficult for developers working in the Linux environment (KDE is a good example here, Python is a less known example)."*
26. Fogel, Karl (2005-10-01). "Producing Open Source Software - How to Run a Successful Free Software Project". O'Reilly Media. ISBN 978-0-596-10502-0. Retrieved 2015-11-29. *"The GPL and license compatibility - Because the primary goal of the GPL's authors is the promotion of free software, they deliberately crafted the license to make it impossible to mix GPLed code into proprietary programs. [...]Any derivative work—that is, any work containing a nontrivial amount of GPLed code—must itself be distributed under the GPL. No additional restrictions may be placed on the redistribution of either the original work or a derivative work."*
27. Apache foundation (2015-05-30). "GPL compatibility". Retrieved 2015-05-30. *"Apache 2 software can therefore be included in GPLv3 projects, because the GPLv3 license accepts our software into GPLv3 works. However, GPLv3 software cannot be included in Apache projects. The licenses are incompatible in one direction only, and it is a result of ASF's licensing philosophy and the GPLv3 authors' interpretation of copyright law."*

28. Hanwell, Marcus D. (28 Jan 2014). "Should I use a permissive license? Copyleft? Or something in the middle?". opensource.com. Retrieved 2015-05-30. *"Permissive licensing simplifies things One reason the business world, and more and more developers [...], favor permissive licenses is in the simplicity of reuse. The license usually only pertains to the source code that is licensed and makes no attempt to infer any conditions upon any other component, and because of this there is no need to define what constitutes a derived work. I have also never seen a license compatibility chart for permissive licenses; it seems that they are all compatible."*
29. "Licence Compatibility and Interoperability". *Open-Source Software - Develop, share, and reuse open source software for public administrations*. joinup.ec.europa.eu. Retrieved 2015-05-30. *"The licences for distributing free or open source software (FOSS) are divided in two families: permissive and copyleft. Permissive licences (BSD, MIT, X11, Apache, Zope) are generally compatible and interoperable with most other licences, tolerating to merge, combine or improve the covered code and to re-distribute it under many licences (including non-free or "proprietary")."*
30. GPL compatibility (<http://www.dwheeler.com/essays/gpl-compatible.html>) on dwheeler.com
31. Chisnall, David (2009-08-31). "The Failure of the GPL". informit.com. Retrieved 2016-01-24. *"The GPL places additional restrictions on the code, and therefore is incompatible. You can combine APSL, MPL, CDDL, Apache, and BSD-licensed code in the same project easily, but you can only combine one of these with GPLv2 code. Even the Free Software Foundation can't manage to get it right. Version 3 of the LGPL, for example, is incompatible with version 2 of the GPL. This has caused a problem recently for a few GNU library projects that wanted to move to LGPLv3 but were used by other projects that were GPLv2-only."*
32. "Frequently Asked Questions about the GNU Licenses – Is GPLv3 compatible with GPLv2?". gnu.org. Retrieved 2014-06-03. *"No. Some of the requirements in GPLv3, such as the requirement to provide Installation Information, do not exist in GPLv2."*
33. Asay, Clark D. "Michigan Telecommunications and Technology Law Review Volume 14 - Issue 22008 The General Public License Version 3.0: Making or Breaking the Foss Movement". law.umich.edu.
34. Landley, Rob. "CELF 2013 Toybox talk". landley.net. Retrieved 2013-08-21. *"GPLv3 broke "the" GPL into incompatible forks that can't share code."*
35. "GPL-Compatible Free Software Licenses". 2014-11-20. Retrieved 2014-12-29.
36. GNU project: Frequently Asked Questions about the GNU Licenses: Why don't you use the GPL for manuals? (<https://www.gnu.org/licenses/gpl-faq.html#WhyNotGPLForManuals>) Retrieved 20 June 2009.
37. Debian Project: Resolution: Why the GNU Free Documentation License is not suitable for Debian (http://www.debian.org/vote/2006/vote_001.en.html#amendmenttexta). Voted February–March 2006. Retrieved 20 June 2009.
38. FLOSS Manuals foundation: License Change (<http://en.flossmanuals.net/bin/view/Blog/LicenseChange>) 6 June 2007. Retrieved 20 June 2009.
39. Wikimedia license update approval (https://wikimediafoundation.org/wiki/Resolution:Licensing_update_approval)
40. Wikipedia + CC BY-SA = Free Culture Win! (<http://creativecommons.org/weblog/entry/15411>) on creativecommons.org by Mike Linksvayer, June 22nd, 2009
41. 2.2 What is the licensing concern? (<http://zfsonlinux.org/faq.html#WhatAboutTheLicensingIssue>) on zfsonlinux.com
42. Aron Xu (2014-08-28). "[zfs-discuss] Summary of ZFS on Linux for Debian (was: zfs-linux_0.6.2-1_amd64.changes REJECTED)". debian.org. Retrieved 2016-01-14. *"Upstream ZoL project [3] holds the view that in this case the combination of the two in the same binary would create a derived work, so this is not acceptable for redistribution. We accept the interpretation that this last case is not acceptable for redistribution. Therefore our package does not (and never will) ship or facilitate building a custom kernel where the ZoL ZFS driver is built-in in a monolithic binary, instead of built as an independent dynamic LKM."*
43. Pkg-zfsonlinux-devel -zfs-linux_0.6.2-1_amd64.changes REJECTED (<http://comments.gmane.org/gmane.linux.file-systems.zfs.user/18380>) by Paul Richards Tagliamonte *"Our consensus was that this package appears to violate the spirit of the GPL at minimum, and may cause legal problems. Judges often interpret documents as they're intended to read, hacks to comply with the letter but not the intent are not looked upon fondly. This may be a hard thing for technical folks to accept, but in legal cases one usually isn't dealing with technical people. As such, this package has been rejected"* (26 Aug 2014)
44. Yao: The State of ZFS on Linux (<https://lwn.net/Articles/611657/>) September 11, 2014 by jake on lwn.net
45. CDDL (<http://www.gnu.org/licenses/license-list.en.html#CDDL>) on gnu.org
46. Jaeger, Till (2005-03-01). "Die GPL kommentiert und erklärt - Institut für Rechtsfragen der Freien und Open Source Software, 1. Auflage" (PDF). *Ziffer 2 GPL* (in German). O'Reilly Media. p. 70. ISBN 3-89721-389-3. Retrieved 2016-01-12. *"In der Praxis ist stark umstritten, ob ein Kernelmodul als "derivative work" betrachtet werden muss. Die Auseinandersetzungen um Binär-Treiber für Linux werden mit Heftigkeit geführt. Man wird wohl nicht für sämtliche Kernelmodule eine einheitliche Antwort finden können: Wann ein Kernelmodul von Linux »abgeleitet« ist, hängt stark von der technischen Umsetzung ab und richtet sich nach den oben dargelegten Kriterien. [...] Es*

- hängt stark von der technischen Umsetzung ab und richtet sich nach den oben dargelegten Kriterien. [...] Es existieren allerdings auch Kernelmodule, die älter sind als Linux, etwa das Dateisystem AFS. Dort liegt es auf der Hand, dass sie als funktional eigenständig anzusehen sind, da sie gar nicht »für Linux« geschrieben sein können."
47. Ubuntu Is Planning To Make The ZFS File-System A "Standard" Offering (https://www.phoronix.com/scan.php?page=news_item&px=Ubuntu-ZFS-Standard-Plans) on phoronix by Michael Larabel (6 October 2015)
 48. ZFS Licensing and Linux (<https://insights.ubuntu.com/2016/02/18/zfs-licensing-and-linux/>) on ubuntu.com by Dustin Kirkland (18 February 2016)
 49. Are GPLv2 and CDDL incompatible? (<http://blog.hansenpartnership.com/are-gplv2-and-cddl-incompatible/>) on hansenpartnership.com by James E.J. Bottomley *"What the above analysis shows is that even though we presumed combination of GPLv2 and CDDL works to be a technical violation, there's no way actually to prosecute such a violation because we can't develop a convincing theory of harm resulting. Because this makes it impossible to take the case to court, effectively it must be concluded that the combination of GPLv2 and CDDL, provided you're following a GPLv2 compliance regime for all the code, is allowable."* (23 February 2016)
 50. The Linux Kernel, CDDL and Related Issues (<https://softwarefreedom.org/resources/2016/linux-kernel-cddl.html>) on softwarefreedom.org by Eben Moglen & Mishi Choudhary (26 February 2016)
 51. GPL Violations Related to Combining ZFS and Linux (<https://sfconservancy.org/blog/2016/feb/25/zfs-and-linux/>) on sfconservancy.org by Bradley M. Kuhn and Karen M. Sandler *"Ultimately, various Courts in the world will have to rule on the more general question of Linux combinations. Conservancy is committed to working towards achieving clarity on these questions in the long term. That work began in earnest last year with the VMware lawsuit, and our work in this area will continue indefinitely, as resources permit. We must do so, because, too often, companies are complacent about compliance. While we and other community-driven organizations have historically avoided lawsuits at any cost in the past, the absence of litigation on these questions caused many companies to treat the GPL as a weaker copyleft than it actually is."* (February 25, 2016)
 52. GPL Violations Related to Combining ZFS and Linux (<https://sfconservancy.org/blog/2016/feb/25/zfs-and-linux/>) on sfconservancy.org by Bradley M. Kuhn and Karen M. Sandler *"Conservancy (as a **Linux copyright holder ourselves**), along with the members of our coalition in the GPL Compliance Project for Linux Developers, all agree that Canonical and others infringe Linux copyrights when they distribute zfs.ko."*
 53. Compatible Licenses (<https://creativecommons.org/compatiblelicenses/>) on creativecommons.org *"GPLv3: The GNU General Public License version 3 was declared a "BY-SA-Compatible License" for version 4.0 on 8 October 2015. Note that compatibility with the GPLv3 is one-way only, which means you may license your contributions to adaptations of BY-SA 4.0 materials under GPLv3, but you may not license your contributions to adaptations of GPLv3 projects under BY-SA 4.0."*
 54. O'Riordan, Ciaran (2006-10-06). "(About GPLv3) Can the Linux Kernel Relicense?". fsfe.org. Retrieved 2015-05-28. *"Someone who works with many lawyers on free software copyright issues later told me that it is not necessary to get permission from 100% of the copyright holders. It would suffice if there was permission from the copyright holders of 95% of the source code and no objections from the holders of the other 5%. This, I'm told, is how Mozilla was able to relicense to the GPL in 2003 despite years of community contributions."*
 55. Licensing HOWTO (<http://www.catb.org/~esr/Licensing-HOWTO.html>) by Eric Steven Raymond&Catherine Olanich Raymond *"Changing an existing license [...]You can change the license on a piece of code under any of the following conditions: If you are the sole copyright holder[...]If you are the sole registered copyright holder[...] If you obtain the consent of all other copyright holders[...]If no other copyright holder could be harmed by the change"* (accessed on 2015-11-21)
 56. Netscape Public License FAQ (<http://website-archive.mozilla.org/www.mozilla.org/mpl/MPL/NPL/1.0/FAQ.html>) on mozilla.org
 57. "Licenses by Name - Open Source Initiative". Open Source Initiative. Retrieved 2014-08-27.
 58. On the Netscape Public License (<https://www.gnu.org/philosophy/netscape-npl.html>) by Richard Stallman on GNU.org
 59. "Mozilla Relicensing FAQ Version 1.1". mozilla.org. Archived from the original on 2010-05-13. *"Some time ago mozilla.org announced its intent to seek relicensing of Mozilla code under a new licensing scheme that would address perceived incompatibilities of the Mozilla Public License (MPL) with the GNU General Public License (GPL) and GNU Lesser General Public License (LGPL)."*
 60. Relicensing Complete (http://blog.gerv.net/2006/03/relicensing_complete/) on gerv.net by Gervase Markham (March 31, 2006)
 61. February 2001 (<http://lists.xiph.org/pipermail/vorbis/2001-February/001758.html>) on xiph.org *"With the Beta 4 release, the Ogg Vorbis libraries have moved to the BSD license. The change from LGPL to BSD was made to enable the use of Ogg Vorbis in all forms of software and hardware. Jack Moffitt says, "We are changing the license in response to feedback from many parties. It has become clear to us that adoption of Ogg Vorbis will be accelerated even further by the use of a less restrictive license that is friendlier toward proprietary software and hardware"*

- systems. We want everyone to be able to use Ogg Vorbis."
62. RMS on license change (<http://lwn.net/2001/0301/a/rms-ov-license.php3>) on lwn.net
 63. Denis-Courmont, Rémi. "VLC media player to remain under GNU GPL version 2". videolan.org. Retrieved 2015-11-21. *"In 2001, VLC was released under the OSI-approved GNU General Public version 2, with the commonly-offered option to use "any later version" thereof (though there was not any such later version at the time). Following the release by the Free Software Foundation (FSF) of the new version 3 of its GNU General Public License (GPL) on the 29th of June 2007, contributors to the VLC media player, and other software projects hosted at videolan.org, debated the possibility of updating the licensing terms for future version of the VLC media player and other hosted projects, to version 3 of the GPL. [...] There is strong concern that these new additional requirements might not match the industrial and economic reality of our time, especially in the market of consumer electronics. It is our belief that changing our licensing terms to GPL version 3 would currently not be in the best interest of our community as a whole. Consequently, we plan to keep distributing future versions of VLC media player under the terms of the GPL version 2. [...]we will continue to distribute the VLC media player source code under GPL "version 2 or any later version" until further notice."*
 64. "Changing the VLC engine license to LGPL". Retrieved 23 October 2011.
 65. Vaughan-Nichols, Steven. "No GPL Apps for Apple's App Store". zdnet.com. Retrieved 23 October 2011.
 66. VLC under Mozilla public relaunched. (<http://arstechnica.com/apple/2013/07/vlc-media-player-returns-to-the-ios-app-store-after-30-month-hiatus/>) on Ars Technica (Accessed 10/10/2013)
 67. "Browse /LZMA SDK/4.23". Sourceforge. Retrieved 2014-02-12.
 68. Igor Pavlov (2013). "LZMA SDK (Software Development Kit)". Retrieved 2013-06-16.
 69. Mavrogiannopoulos, Nikos (2013-03-26). "The perils of LGPLv3". gnutls.org. Retrieved 2015-11-18. *"LGPLv3 is the latest version of the GNU Lesser General Public License. It follows the successful LGPLv2.1 license, and was released by Free Software Foundation as a counterpart to its GNU General Public License version 3. The goal of the GNU Lesser General Public Licenses is to provide software that can be used by both proprietary and free software. This goal has been successfully handled so far by LGPLv2.1, and there is a multitude of libraries using that license. Now we have LGPLv3 as the latest, and the question is how successful is LGPLv3 on this goal? In my opinion, very little. If we assume that its primary goal is to be used by free software, then it blatantly fails that."*
 70. Version 2.99.4 (<http://upstream.rosalinux.ru/changelogs/gnutls/3.2.1/changelog.html>) (released 2011-07-23)[...] ** libgnutls: license upgraded to LGPLv3
 71. 2013-03-14 Nikos Mavrogiannopoulos (nmav@gnutls.org) (<http://www.gnutls.org/abi-tracker/changelog/gnutls/3.1.10/log.html>) * COPYING.LESSER, README: gnutls 3.1.10 is LGPLv2.1
 72. why-the-wikimedia-projects-should-not-use-gfdl-as-a-stand-alone-license-for-images (<http://notablog.notafish.com/post/2005/04/21/26-why-the-wikimedia-projects-should-not-use-gfdl-as-a-stand-alone-license-for-images>)
 73. "FDL 1.3 FAQ". Gnu.org. Retrieved 2011-11-07.
 74. Licensing update rolled out in all Wikimedia wikis (<http://blog.wikimedia.org/2009/06/30/licensing-update-rolled-out-in-all-wikimedia-wikis/>) on wikimedia.org by Erik Moeller on June 30th, 2009 *"Perhaps the most significant reason to choose CC-BY-SA as our primary content license was to be compatible with many of the other admirable endeavors out there to share and develop free knowledge"*
 75. Google android and the linux headers (http://www.theregister.co.uk/2011/03/29/google_android_and_the_linux_headers/) on theregister.com (2011)
 76. Android: Sued by Microsoft, not by Linux (<http://www.itworld.com/article/2744480/open-source-tools/android--sued-by-microsoft--not-by-linux.html>) "Microsoft launches new Android suit, Linus Torvalds' take on Linux kernel headers and Android" on ITworld (March 21, 2011)
 77. Infringement and disclosure risk in development on copyleft platforms (<http://www.ipinfoblog.com/archives/licensing-law-issues-infringement-and-disclosure-risk-in-development-on-copyleft-platforms.html/>) on ipinfoblog.com by Raymond Nimmer (2011)
 78. Prokoudine, Alexandre (2012-12-27). "LibreDWG drama: the end or the new beginning?". libregraphicsworld.org. Retrieved 2013-08-23. *"[...]the unfortunate situation with support for DWG files in free CAD software via LibreDWG. We feel, by now it ought to be closed. We have the final answer from FSF. [...] "We are not going to change the license."*
 79. "license". freecadweb.org. 2014. Retrieved 2015-03-25. *"Licences used in FreeCAD - FreeCAD uses two different licenses, one for the application itself, and one for the documentation: Lesser General Public Licence, version 2 or superior (LGPL2+) [...] Open Publication Licence"*
 80. Relicensing Dolphin: The long road to GPLv2+ (<https://dolphin-emu.org/blog/2015/05/25/relicensing-dolphin/>) Written by JMC47, MaJoR on May 25, 2015
 81. Possible LGPL relicensing #2033 (<https://github.com/mpv-player/mpv/issues/2033>) on github.com *"GPL-incompatible dependencies such as OpenSSL are a big issue for library users, even if the library user is shewith the*

incompatible dependencies such as OpenSSL are a big issue for library users, even if the library user is ok with the GPL."

82. switchin-from-gplv3-to-gplv2 (<https://www.seafile-server.org/en/switchin-from-gplv3-to-gplv2/>)
83. license.txt (<https://github.com/haiwen/seafile/blob/master/LICENSE.txt>) on github.com
84. MAME is now Free and Open Source Software (<http://mamedev.org/?p=422>) on mamedev.org (March 4, 2016)
85. the-already-dead-theory
(<http://web.archive.org/web/20131031202643/http://mamedev.emulab.it/haze/2013/10/25/the-already-dead-theory>)
on mamedev.emulab.it
86. So why did this annoy me so much? (<http://www.mameworld.info/ubbthreads/showflat.php?Number=316273>) on
mameworld.info (10/22/13)
87. "10 months later, MAME finishes its transition to open source". *Gamasutra*. Retrieved 5 March 2016.
88. "MAME is going open source to be a 'learning tool for developers' ". *Gamasutra* (UBM plc). Retrieved 27 May 2015.

External links

- FLOSS Compatibility Diagram (<http://www.dwheeler.com/essays/floss-license-slide.html>) by David A. Wheeler (2007, some licenses superseded)

Retrieved from "https://en.wikipedia.org/w/index.php?title=License_compatibility&oldid=710258566"

Categories: Copyright licenses

-
- This page was last modified on 15 March 2016, at 22:22.
 - Text is available under the Creative Commons Attribution-ShareAlike License; additional terms may apply. By using this site, you agree to the Terms of Use and Privacy Policy. Wikipedia® is a registered trademark of the Wikimedia Foundation, Inc., a non-profit organization.