

OPEN SOURCE LICENSES

Presented by Bob Primak

To the Lexington Computers and technology Users Group (LCTG)

Wednesday, November 19, 2025

First, **What Is a Software License** (legally)?
(Gemini AI in Google Search)

A software license is a legal contract that grants a user permission to use a software product under specific terms and conditions, which can include restrictions on how it can be copied, modified, and shared. This agreement is between the copyright holder (the software author or publisher) and the user (the licensee), and it outlines the user's rights and the author's expectations. It is not a sale of ownership but rather a rental of rights to use intellectual property.

Key components and types of software licenses

Grant of use: The license defines the scope of use, such as the number of devices or users, and what actions are permitted, like copying or modification.

Copyright and intellectual property: The license establishes how the software's intellectual property can be used. It also protects the author by defining unacceptable uses.

Types of licenses:

Proprietary: Often includes restrictions on copying, distribution, and modification.

Open Source: Allows users to view, modify, and distribute the source code, with different variations like:

Copyleft licenses: Require that modifications also be made available under the same license terms.

Permissive licenses: Have very few conditions, often only requiring that copyright information be retained.

Public Domain: The developer forgoes all copyright and gives the software away with essentially no restrictions.

Common license structures:

Per-device: Allows installation on a specific number of devices. A device may be a physical or a virtual instance.

Concurrent use: Permits a specific number of users to access the software simultaneously, often with the help of a license manager.

SaaS/Cloud-based: Involves subscriptions to use software over the internet, where the provider manages updates and access. Most commercial software is moving in this direction.

Open Source License Types: Permissive and Copyleft (Gemini AI in Google Search)

The main types of open-source licenses are **permissive** and **copyleft**.

Permissive licenses have minimal restrictions, allowing code to be used in most projects, including proprietary ones, with simple attribution requirements.

Copyleft licenses are more restrictive, requiring that any derivative works be distributed under the same, or a compatible, copyleft license. A third category is weak or limited copyleft licenses, which provide a middle ground by requiring only certain parts of the code to remain open-source.

Copyleft is a legal term, and a specific type of license that uses copyright law to allow the free distribution and modification of a work, with the condition that any derivative works must also be distributed under the same copyleft license. It is not the opposite of copyright, but rather a method of licensing that works within the existing copyright framework to guarantee ongoing freedom for the work and its future versions.

Based on copyright: Copyleft licenses are only possible because the author first owns the copyright to the original work, giving them the right to decide how it can be distributed.

- **A licensing method:** It is a way to grant certain rights to users while imposing conditions, primarily that any modified versions must be shared under the same terms.
- **Examples:** Well-known examples of copyleft licenses include the GNU General Public License (GPL) and the Creative Commons ShareAlike (CC BY-SA) license.

Open Source Licenses by Category

Page created on September 19, 2006 | Last modified on November 2, 2022

<https://opensource.org/licenses-old/category>

I'm going to take this list very fast, then concentrate on several well-known open source licenses and some of the controversies surrounding them.

(These will be flashed as a rapid series of slides.)

Licenses that are “popular and widely-used or with strong communities”

The below list is based on publicly available statistics obtained at the time of the Report of License Proliferation Committee.

Apache License 2.0 (Apache-2.0)

3-clause BSD license (BSD-3-Clause)

2-clause BSD license (BSD-2-Clause)

GNU General Public License (GPL)

GNU Lesser General Public License (LGPL)

MIT license (MIT)

Mozilla Public License 2.0 (MPL-2.0)

Common Development and Distribution License 1.0 (CDDL-1.0)

Eclipse Public License 2.0 (EPL-2.0)

International licenses

CeCILL License 2.1

European Union Public License (EUPL-1.2)

Licence Libre du Québec – Permissive (LiLiQ-P) version 1.1 (LiLiQ-P-1.1)

Licence Libre du Québec – Réciprocité (LiLiQ-R) version 1.1 (LiLiQ-R-1.1)

Licence Libre du Québec – Réciprocité forte (LiLiQ-R+) version 1.1 (LiLiQ-Rplus-1.1)

Mulan Permissive Software License v2 (MulanPSL – 2.0)

Special purpose licenses

Certain licensors, such as schools and the US government, have specialized concerns, such as specialized rules for government copyrights. Licenses that were identified by the License Proliferation Committee as meeting a special need were placed in this group.

BSD+Patent (BSD-2-Clause-Patent)

CERN Open Hardware Licence Version 2 – Permissive

CERN Open Hardware Licence Version 2 – Weakly Reciprocal
CERN Open Hardware Licence Version 2 – Strongly Reciprocal
Educational Community License, Version 2.0 (ECL-2.0)
IPA Font License (IPA)
Lawrence Berkeley National Labs BSD Variant License (BSD-3-Clause-LBNL)
NASA Open Source Agreement 1.3 (NASA-1.3)
OSET Public License version 2.1 (OSET-PL-2.1)
SIL Open Font License 1.1 (OFL-1.1)
Unicode License Agreement – Data Files and Software
The Unlicense (Unlicense)
Upstream Compatibility License v1.0 (UCL-1.0)

Other/Miscellaneous licenses

These licenses do not fall neatly into any category.

0-clause BSD License (0BSD)
1-clause BSD License (BSD-1-Clause)
Adaptive Public License (APL-1.0)
Artistic license 2.0 (Artistic-2.0)
Free Public License 1.0.0 (0BSD)
JAM License (Jam)
MIT No Attribution License (MIT-0)
Open Software License (OSL-3.0)
Q Public License (QPL-1.0)
Universal Permissive License (UPL)
Zero-Clause BSD (0BSD)
zlib/libpng license (Zlib)

Licenses that are redundant with more popular licenses

Several licenses in this group are excellent licenses and have their own followings, however these licenses were perceived by the License Proliferation Committee as completely or partially redundant with existing licenses.

Academic Free License 3,0 (AFL-3.0)

Attribution Assurance License (AAL)

Eiffel Forum License V2.0 (EFL-2.0)

Fair License (Fair)

Historical Permission Notice and Disclaimer (HPND)

Lucent Public License Version 1.02 (LPL-1.02)

OpenLDAP Public License Version 2.8 (OLDAP-2.8)

The PostgreSQL License (PostgreSQL)

University of Illinois/NCSA Open Source License (NCSA)

X.Net License (Xnet)

Zope Public License 2.1

Non-reusable licenses

Licenses in this group are specific to their authors and cannot be reused by others. Many, but not all, of these licenses fall into the category of vanity licenses.

Apple Public Source License (APSL-2.0)

Computer Associates Trusted Open Source License 1.1 (CATOSL-1.1)

eCos License version 2.0

EU DataGrid Software License (EUDatagrid)

Entessa Public License (Entessa)

Frameworkx License (Frameworkx-1.0)

IBM Public License 1.0 (IPL-1.0)

LaTeX Project Public License 1.3c (LPPL-1.3c)

Motosoto License (Motosoto)
Multics License (Multics)
Naumen Public License (Naumen)
Nethack General Public License (NGPL)
Nokia Open Source License (Nokia)
OCLC Research Public License 2.0 (OCLC-2.0)
Python License (Python-2.0)
CNRI Python license (CNRI-Python) (CNRI portion of Python License)
RealNetworks Public Source License V1.0 (RPSL-1.0)
Ricoh Source Code Public License (RSCPL)
Sleepycat License (Sleepycat)
Sun Public License 1.0 (SPL-1.0)
Sybase Open Watcom Public License 1.0 (Watcom-1.0)
Vovida Software License v. 1.0 (VSL-1.0)
W3C License (W3C)
wxWindows Library License (Wxwindows)

Superseded licenses

Licenses in this category have been superseded by newer versions.

Apache Software License 1.1 (Apache-1.1)
Artistic license 1.0 (Artistic-1.0)
Common Public License 1.0 (CPL-1.0)
Eclipse Public License 1.0 (EPL-1.0)
Educational Community License, Version 1.0 (ECL-1.0)
Eiffel Forum License V1.0 (EFL-1.0)
EUPL V1.1 (EUPL-1.1)
Lucent Public License ("Plan9"), version 1.0 (LPL-1.0)

Mozilla Public License 1.0 (MPL-1.0)
Mozilla Public License 1.1 (MPL-1.1)
Open Software License 1.0 (OSL-1.0)
Open Software License 2.1 (OSL-2.1)
PHP License 3.0 (PHP-3.0)
Reciprocal Public License, version 1.1 (RPL-1.1)
Zope Public License 2.0 (ZPL-2.0)

Licenses that have been voluntarily retired

Self-defining category. No one should use these licenses going forward, although we assume that licensors may or may not choose to continue to use them.

CUA Office Public License Version 1.0 (CUA-OPL-1.0)
Intel Open Source License (Intel)
Jabber Open Source License
MITRE Collaborative Virtual Workspace License (CVW)
Sun Industry Standards Source License (SISSL)
Uncategorized Licenses
Boost Software License (BSL-1.0)
Cryptographic Autonomy License v.1.0 (CAL-1.0)
Common Public Attribution License 1.0 (CPAL-1.0)
GNU Affero General Public License version 3 (AGPL-3.0)
ISC License (ISC)
Microsoft Public License (MS-PL)
Microsoft Reciprocal License (MS-RL)
MirOS Licence (MirOS)
Non-Profit Open Software License 3.0 (NPOSL-3.0)
NTP License (NTP)
Open Group Test Suite License (OGTSL)

Reciprocal Public License 1.5 (RPL-1.5)
Simple Public License 2.0 (SimPL-2.0)

(End of rapid-fire slides)

As can be seen, there's a plethora of open source license types, and they span the entire range from nearly copyright, through restricted open source, to truly copyleft, as defined above.

Let's concentrate on some of the more commonly seen license types.

(Gemini AI in Google Search)

Five of the most common open-source license types are (1) the [MIT License](#), (2) the [Apache License 2.0](#), (3) the [GNU General Public License \(GPL\)](#), the (4) [Mozilla Public License \(MPL\)](#), and the (5) [BSD Licenses](#). These licenses vary in how they handle software modification and distribution, with some being more "permissive" (like MIT and BSD) and others being "copyleft" (like GPL). (I'll skip the [GNU Lesser General Public License \(LGPL\)](#) due to time constraints.) [Creative Commons Licensing](#) is also used, but its structure is complex, and I don't have enough time to go into its nuances.

License Category	Examples	Key Characteristics
<u>Permissive Licenses</u>	MIT Apache 2.0 BSD	- Allow broad use, modification, and redistribution. - Can be used in proprietary (closed-source) software. - Typically only require attribution and copyright notices to be preserved.
<u>Copyleft Licenses</u>	GNU GPL Mozilla Public License (MPL)	- Strong Copyleft (GPL): Requires derivative works to be licensed under the same terms. - Weak Copyleft (MPL): Applies the copyleft provisions to the licensed code itself, but allows for linking with non-copyleft code.

Permissive licenses

- **MIT License**: A very simple and permissive license that allows users to do almost anything with the code, as long as they include the original copyright and license notice.
- **BSD Licenses (2-Clause and 3-Clause)**: Permissive licenses that allow unrestricted use, modification, and distribution, requiring only the original copyright and disclaimer to be retained. The 3-Clause version includes an additional clause that restricts advertising the software as derived from the original without permission.
- **Apache License 2.0**: A permissive license that grants rights to use, modify, and distribute the code. It also includes an explicit grant of patent rights from contributors to users.

Copyleft licenses

- **GNU General Public License (GPL)**: A strong "copyleft" license. If you use GPL-licensed code in your project and distribute it, your entire project must also be licensed under the GPL, making all of its source code open.
- **Mozilla Public License (MPL)**: A "weak copyleft" license that is file-based. Modifications to MPL-licensed files must be shared under the MPL, but you can combine these files with files under other licenses, including proprietary ones.

(1) MIT License

Exploring the MIT Open Source License: A Comprehensive Guide

Dive into the history and significance of the MIT Open Source License. Learn how this permissive license empowers open source projects.

<https://tlo.mit.edu/understand-ip/exploring-mit-open-source-license-comprehensive-guide>

Historical Background

The MIT Open Source License, often referred to as the "MIT License," has a rich history dating back to the early days of software development and the Massachusetts Institute of Technology. This permissive open-source license was born out of a need for collaborative, freely shareable software, and it has since become one of the most popular licenses in the open source community.

The MIT License traces its roots to the MIT's Project MAC, an ambitious initiative in the 1960s that aimed to develop a compatible time-sharing operating system. During this period, software was often created as part of academic and research projects, and there was a growing interest in sharing these programs across the academic and scientific community.

Over the years, the MIT License evolved and gained recognition for its simplicity and flexibility. Its permissive nature allows developers to use, modify, and distribute software under this license with minimal restrictions, making it a top choice for many open source projects.

Key Features

The MIT License is known for its brevity and clarity. It grants permission to use, modify, and distribute the software, with the condition that the original copyright notice and the license text are retained in the redistributed software. This ensures proper attribution to the original authors while offering maximum freedom for developers.

Today, the MIT Open Source License is widely adopted in the open source community, contributing to the growth of countless projects, libraries, and applications. Its simplicity and non-restrictive nature make it an appealing choice for developers, as it encourages collaboration and the free flow of knowledge.

The MIT Open Source License is more than just a legal document; it's a testament to the collaborative spirit of the open source software community. Understanding its history and significance can help developers make informed decisions about licensing their projects and contribute to the continued growth of open source software.

Some popular open source projects that use the MIT License include:

Python

Ruby on Rails

TensorFlow

React

jQuery

Node.js

Controversies Surrounding the MIT License

(Gemini AI)

The main controversy surrounding the MIT License is its **lack of explicit patent protection**, which leaves users vulnerable to patent lawsuits from the original copyright holder. Other controversies include how its permissiveness can be seen as problematic when large companies use open-source code for proprietary products without contributing back, and confusion over licensing when a project is forked. Some projects have addressed this by adding separate patent licenses or switching to more comprehensive licenses like the [Apache License 2.0](#).

Patent issues

- **No explicit patent grant:** The MIT license doesn't mention patents, and while some interpret an implicit patent license, it is not a guarantee. This leaves users open to "submarine patent" threats, where the license holder could later sue over patent infringement.
- **A need for explicit grants:** Some projects, like Facebook's React, have previously addressed this by adding a separate patent license to the MIT license. More modern licenses, like Apache 2.0, have explicit patent provisions to avoid this ambiguity.

"Abuse" by large corporations

- **One-way use:** While the license is designed to be permissive, some argue that it can be "abused" when large companies use open-source code in proprietary products without contributing back to the community.
- **Examples:** Some forks of projects like Chromium are turned into proprietary products (like [Chrome](#) and [Edge](#)) while others, like Microsoft's VS Code, restrict access to official extensions for forks, making them less useful.

Forcing forks and other issues

- **Forking issues:** The permissiveness of the MIT license has been criticized in projects like Bitcoin, where its wide adoption has led to a proliferation of forks, causing confusion and debate among developers about project ownership and governance.
- **Trivial misuse:** In some instances, developers have been upset when their work was used without attribution, but the license only requires the original license and copyright notice to be included, not attribution for all modifications.

Potential solutions and alternatives

- **Explicit patent licenses:** For projects that are concerned about patent issues, adding a separate patent grant to the MIT license is a potential solution.
- **Switching to a more comprehensive license:** Projects can opt for a more comprehensive license like the Apache License 2.0, which addresses patents, copyright, and trademark separately.
- **GPL-style licenses:** Some developers switch to copyleft licenses like the GNU General Public License (GPL) to ensure that modifications are also shared under the same license terms, which can help prevent proprietary forks from being created.

(2) Apache License, Version 2.0

<https://www.apache.org/licenses/LICENSE-2.0>

The 2.0 version of the Apache License, approved by the ASF in 2004, helps us achieve our goal of providing reliable and long-lived software products through collaborative, open-source software development.

All packages produced by the ASF are implicitly licensed under the Apache License, Version 2.0, unless otherwise explicitly stated.

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, **"control"** means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or **"Your"**) shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, **"submitted"** means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as **"Not a Contribution."**

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

1. You must give any other recipients of the Work or Derivative Works a copy of this License; and
2. You must cause any modified files to carry prominent notices stating that You changed the files; and
3. You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
4. If the Work includes a "**NOTICE**" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

Controversies Surrounding the Apache License

(Gemini AI)

Controversies surrounding the Apache License 2.0 include its incompatibility with older GPL versions, its demanding documentation and change log requirements, and a patent termination clause that can end rights if a user sues for infringement. The license's complexity also raises concerns, as it requires careful legal review and can be perceived as more restrictive than simpler permissive licenses like MIT.

Key controversies

- **GPL Incompatibility:** The patent termination clause in Apache 2.0 prevents projects from being directly combined with software under older GPL versions (like GPL v2) without relicensing the entire project under GPL v3.
- **Documentation Burden:** Unlike simpler licenses, Apache 2.0 requires including the full license text and extensive change logs, which can be a significant administrative burden and lead to "developer burnout".
- **Patent Clause:** The "in terrorem" clause terminates patent licenses if a user sues a contributor for patent infringement related to the licensed software. While intended to deter lawsuits, this can create legal uncertainty for users who might need to defend themselves legally.
- **Complexity and Legal Uncertainty:** The license is longer and more complex than other permissive licenses, requiring more legal scrutiny from companies. The patent license is granted under contract law, which can lead to variations in interpretation across different jurisdictions.
- **Trademark Rights:** The license explicitly states it does not grant rights to the licensor's trademarks, which is often misunderstood.
- **"Sneaky" License Changes:** Some developers have reported projects changing from Apache 2.0 to a proprietary or "source-available" license without providing a compatible open-source license for the existing code, which undermines the original terms.

Benefits of the license (counterarguments)

- **Patent Protection:** The explicit patent grant makes it very attractive for projects that might be subject to patents, providing clear rights to users.

- **Clear Attribution:** The requirement to state changes and include notices provides greater specificity about contributor obligations, which can be helpful in larger projects.
- **Attracts Organizations:** The robust legal framework, including patent clauses, can be appealing to larger organizations that are more concerned about patent issues.

(3) GNU General Public License

<https://www.gnu.org/licenses/gpl-3.0.en.html>

Summary of GPL 3:

(Gemini AI)

The GNU General Public License (GPL) is a free software license that grants users the freedom to run, copy, distribute, study, and modify the software. Its key feature is copyleft, which ensures that any modified versions of the software are also released under the same GPL license, preserving its free nature and preventing it from becoming proprietary. When you use GPL-licensed software as a tool, the output of your work is not considered a derivative, but if you modify the GPL code itself, the entire resulting work must be licensed under the GPL.

Key aspects of the GPL

•Four essential freedoms:

- The freedom to use the software for any purpose.
- The freedom to study how the software works and change it to suit your needs.
- The freedom to redistribute copies so you can help others.
- The freedom to distribute copies of your modified versions to others.

•**Copyleft:** This is the core principle of the GPL. It uses copyright law to ensure that software remains free. If you distribute a program based on GPL-licensed code, you must also release its source code under the GPL.

•**Source code availability:** The license requires that the source code must be made available to recipients of the software.

•**No proprietary barriers:** The GPL prevents the software from being turned into proprietary software, ensuring that the freedoms it grants are maintained for all future users.

•**No "viral" effect on derivative works:**

- Using a GPL-licensed program as a tool (like a text editor) does not mean the output of that tool (like a book) must be GPL.
- The "viral" or "copyleft" effect applies to the software itself. Any modifications to the GPL code mean the entire new program is subject to the GPL license.

•**Versions:** The two most widely used versions are GPL v2 (1991) and GPL v3 (2007). GPL v3 offers updated provisions, including clearer procedures for addressing non-compliance and protections against "tivoization" (preventing users from running modified software on a device).

This last item refers to a controversy involving the TiVo video recording devices.

The GPL TiVo controversy, known as "tivoization," stemmed from TiVo's use of the GPLv2-licensed Linux operating system in its DVRs. While TiVo complied with GPLv2 by releasing its modified source code, it circumvented the spirit of the license by using hardware-level digital restrictions (DRM) to prevent users from running modified or third-party software on the devices. This sparked a major debate within the free and open-source software community, leading the Free Software Foundation (FSF) to create the new, more restrictive GPLv3 license to specifically prevent this practice.

The issue:

- **TiVo's action:** TiVo used GPL-licensed Linux but implemented a digital signature check to ensure that only its own authorized software could run on its hardware.
- **The problem:** The Free Software Foundation (FSF) argued that this practice, known as "tivoization," violated the spirit of the GPL by denying users the right to install modified versions of the software on the hardware they owned.
- **The debate:** Although TiVo released its source code as required, the hardware locked users out of running modified versions, which the FSF saw as circumventing the user's freedom to modify and run the software.

The response:

- **GPLv3:** The controversy was a primary reason for the development of the GNU General Public License version 3 (GPLv3).
- **Anti-tivoization clause:** GPLv3 included a specific anti-tivoization clause to prevent this type of hardware restriction, which aims to ensure users can run modified versions of the software they have the source code for.
- **Mixed reactions:** The addition of this clause was controversial. While many free software supporters welcomed it, some, like Linus Torvalds and the Linux community, disagreed with its restrictive nature.

Truth Social and Mastodon

The AGPL 3.0 license is a strong copyleft license that extends the requirements of the GPL to software used over a network. Its main feature is that if you modify the software and use it to provide a service over a network, you must make the complete source code of the modified version available to users. This ensures that the freedom of the software remains available to anyone who interacts with it, even if they only access it remotely.

Key features

- [Strong copyleft:](#)

Like the GPL, it requires that any derivative works are distributed under the same AGPL 3.0 license, but it adds a crucial condition for network use.

- **Network copyleft:**

The license ensures that if you modify the software and offer a service over a network, you must provide the source code of your modified version to anyone who interacts with it.

- **Availability of source code:**

To comply, you must provide the full source code of the modified program and any other software that forms part of the service.

- **Preservation of rights:**

It requires that all copyright and license notices are preserved and includes an express grant of patent rights from contributors.

- **Dual licensing:**

Some software projects, like [MinIO](#), are available under both the AGPL 3.0 and a separate commercial license. This allows developers to use the software commercially if they purchase the commercial license instead of adhering to the open-source requirements.

Who it's for

- **Developers:**

The AGPL 3.0 license is a good choice for developers who want to ensure that their open-source projects and any modifications remain free and open, even for network-based services.

- **Users:**

It protects user freedom by ensuring that everyone can access the source code for any modified version they use over a network.

- **Businesses:**

Businesses may need to use a commercial license if they want to avoid the AGPL 3.0 requirements of making their modified code public, such as when building a proprietary service on top of AGPL software.

The Truth Social Controversy

The Truth Social open-source licensing controversy arose when the platform used code from the open-source project Mastodon without complying with the open-source license's terms. Mastodon is released under the AGPLv3 license, which requires that any network-connected modifications must have their source code made publicly available to all users. Critics like the Software Freedom Conservancy accused Truth Social of violating this license by not providing its source code and misrepresenting the platform as proprietary.

Key details of the controversy

- **Initial use of code:** A beta version of Truth Social was found to be using Mastodon's open-source code, including elements like the software's frontend and underlying HTML.
- **License violation:** By not making its source code publicly available, Truth Social was violating the AGPLv3 license, which is a "copyleft" license requiring all users who operate the software over a network to make their modifications available.
- **Truth Social's response:** The platform initially claimed its source code was "proprietary" and removed references to Mastodon. It eventually published its source code in a ZIP file on its website after public pressure and formal requests from the Software Freedom Conservancy and Mastodon.
- **Legal action:** Mastodon sent a formal notice to Truth Social, giving the company 30 days to comply or face potential legal action, including the possibility of having its right to use the code revoked.
- **Current status:** While Truth Social eventually complied by publishing its source code, the initial controversy highlighted the tension between open-source

principles and commercial interests. The source code is now available in the website's legal section and on GitHub.

(4) Mozilla Public License

<https://www.mozilla.org/en-US/MPL/>

About the License

Mozilla is the custodian of the Mozilla Public License ("MPL"), an open source/free software license.

The current version of the license is MPL 2.0 ([html](#) | [plain text](#)). If you want to use or distribute code licensed under the MPL 2.0 and have questions about it, you may want to read the [FAQ](#).

MPL 2.0 Revision Process

The release of MPL 2.0 was the result of a two year process that revised MPL 1.1. A [Revision FAQ](#) documents this process, and explains the most significant changes made.

Historical Documents

Various historical documents relating to the Mozilla and Netscape Public Licenses are available, including deprecated versions of the license such as MPL 1.1.

Mozilla Licensing Information

The Mozilla Project is only one of many users of the MPL, but because many people come to this page looking for information about Mozilla's open source licensing policies and practices, we've provided the information below as a reference.

Correctly Licensing New Source Code

Any new code checked into Mozilla's source repositories needs to comply with Mozilla's source code licensing policy. Please use the appropriate header text at the top of each file.

Licenses For Existing Source Code

Most Mozilla software projects use the MPL, but some have different terms. Detailed information on the licensing of existing code can be found by inspecting its license headers, or by visiting the license information page in the relevant Mozilla software.

For information on how other things are licensed, including Mozilla's trademarks and websites, see our general licensing information page.

Questions?

If, after reading all the above carefully (particularly the FAQ) you have a further question about the MPL or the licensing terms of Mozilla project code, please send it to licensing@mozilla.org.

Mozilla Public License

Version 2.0

<https://www.mozilla.org/en-US/MPL/2.0/>

1. Definitions

1.1. "Contributor"

means each individual or legal entity that creates, contributes to the creation of, or owns Covered Software.

1.2. “Contributor Version”

means the combination of the Contributions of others (if any) used by a Contributor and that particular Contributor’s Contribution.

1.3. “Contribution”

means Covered Software of a particular Contributor.

1.4. “Covered Software”

means Source Code Form to which the initial Contributor has attached the notice in Exhibit A, the Executable Form of such Source Code Form, and Modifications of such Source Code Form, in each case including portions thereof.

1.5. “Incompatible With Secondary Licenses”

means

that the initial Contributor has attached the notice described in Exhibit B to the Covered Software; or

that the Covered Software was made available under the terms of version 1.1 or earlier of the License, but not also under the terms of a Secondary License.

1.6. “Executable Form”

means any form of the work other than Source Code Form.

1.7. “Larger Work”

means a work that combines Covered Software with other material, in a separate file or files, that is not Covered Software.

1.8. “License”

means this document.

1.9. “Licensable”

means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently, any and all of the rights conveyed by this License.

1.10. “Modifications”

means any of the following:

any file in Source Code Form that results from an addition to, deletion from, or modification of the contents of Covered Software; or

any new file in Source Code Form that contains any Covered Software.

1.11. “Patent Claims” of a Contributor

means any patent claim(s), including without limitation, method, process, and apparatus claims, in any patent Licensable by such Contributor that would be infringed, but for the grant of the License, by the making, using, selling, offering for sale, having made, import, or transfer of either its Contributions or its Contributor Version.

1.12. “Secondary License”

means either the GNU General Public License, Version 2.0, the GNU Lesser General Public License, Version 2.1, the GNU Affero General Public License, Version 3.0, or any later versions of those licenses.

1.13. “Source Code Form”

means the form of the work preferred for making modifications.

1.14. “You” (or “Your”)

means an individual or a legal entity exercising rights under this License. For legal entities, “You” includes any entity that controls, is controlled by, or is under common control with You. For purposes of this definition, “control” means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants and Conditions

2.1. Grants

Each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

under intellectual property rights (other than patent or trademark) Licensable by such Contributor to use, reproduce, make available, modify, display, perform, distribute, and otherwise exploit its Contributions, either on an unmodified basis, with Modifications, or as part of a Larger Work; and

under Patent Claims of such Contributor to make, use, sell, offer for sale, have made, import, and otherwise transfer either its Contributions or its Contributor Version.

2.2. Effective Date

The licenses granted in Section 2.1 with respect to any Contribution become effective for each Contribution on the date the Contributor first distributes such Contribution.

2.3. Limitations on Grant Scope

The licenses granted in this Section 2 are the only rights granted under this License. No additional rights or licenses will be implied from the distribution or licensing of Covered Software under this License. Notwithstanding Section 2.1(b) above, no patent license is granted by a Contributor:

for any code that a Contributor has removed from Covered Software; or

for infringements caused by: (i) Your and any other third party's modifications of Covered Software, or (ii) the combination of its Contributions with other software (except as part of its Contributor Version); or

under Patent Claims infringed by Covered Software in the absence of its Contributions.

This License does not grant any rights in the trademarks, service marks, or logos of any Contributor (except as may be necessary to comply with the notice requirements in Section 3.4).

2.4. Subsequent Licenses

No Contributor makes additional grants as a result of Your choice to distribute the Covered Software under a subsequent version of this License (see Section 10.2) or under the terms of a Secondary License (if permitted under the terms of Section 3.3).

2.5. Representation

Each Contributor represents that the Contributor believes its Contributions are its original creation(s) or it has sufficient rights to grant the rights to its Contributions conveyed by this License.

2.6. Fair Use

This License is not intended to limit any rights You have under applicable copyright doctrines of fair use, fair dealing, or other equivalents.

2.7. Conditions

Sections 3.1, 3.2, 3.3, and 3.4 are conditions of the licenses granted in Section 2.1.

3. Responsibilities

3.1. Distribution of Source Form

All distribution of Covered Software in Source Code Form, including any Modifications that You create or to which You contribute, must be under the terms of this License. You must inform recipients that the Source Code Form of the Covered Software is governed by the terms of this License, and how they can obtain a copy of this License. You may not attempt to alter or restrict the recipients' rights in the Source Code Form.

3.2. Distribution of Executable Form

If You distribute Covered Software in Executable Form then:

such Covered Software must also be made available in Source Code Form, as described in Section 3.1, and You must inform recipients of the Executable Form

how they can obtain a copy of such Source Code Form by reasonable means in a timely manner, at a charge no more than the cost of distribution to the recipient; and

You may distribute such Executable Form under the terms of this License, or sublicense it under different terms, provided that the license for the Executable Form does not attempt to limit or alter the recipients' rights in the Source Code Form under this License.

3.3. Distribution of a Larger Work

You may create and distribute a Larger Work under terms of Your choice, provided that You also comply with the requirements of this License for the Covered Software. If the Larger Work is a combination of Covered Software with a work governed by one or more Secondary Licenses, and the Covered Software is not Incompatible With Secondary Licenses, this License permits You to additionally distribute such Covered Software under the terms of such Secondary License(s), so that the recipient of the Larger Work may, at their option, further distribute the Covered Software under the terms of either this License or such Secondary License(s).

3.4. Notices

You may not remove or alter the substance of any license notices (including copyright notices, patent notices, disclaimers of warranty, or limitations of liability) contained within the Source Code Form of the Covered Software, except that You may alter any license notices to the extent required to remedy known factual inaccuracies.

3.5. Application of Additional Terms

You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, You may do so only on Your own behalf, and not on behalf of any Contributor. You

must make it absolutely clear that any such warranty, support, indemnity, or liability obligation is offered by You alone, and You hereby agree to indemnify every Contributor for any liability incurred by such Contributor as a result of warranty, support, indemnity or liability terms You offer. You may include additional disclaimers of warranty and limitations of liability specific to any jurisdiction.

4. Inability to Comply Due to Statute or Regulation

If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Software due to statute, judicial order, or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be placed in a text file included with all distributions of the Covered Software under this License. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

5. Termination

5.1. The rights granted under this License will terminate automatically if You fail to comply with any of its terms. However, if You become compliant, then the rights granted under this License from a particular Contributor are reinstated (a) provisionally, unless and until such Contributor explicitly and finally terminates Your grants, and (b) on an ongoing basis, if such Contributor fails to notify You of the non-compliance by some reasonable means prior to 60 days after You have come back into compliance. Moreover, Your grants from a particular Contributor are reinstated on an ongoing basis if such Contributor notifies You of the non-compliance by some reasonable means, this is the first time You have received notice of non-compliance with this License from such Contributor, and You become compliant prior to 30 days after Your receipt of the notice.

5.2. If You initiate litigation against any entity by asserting a patent infringement claim (excluding declaratory judgment actions, counter-claims, and cross-

claims) alleging that a Contributor Version directly or indirectly infringes any patent, then the rights granted to You by any and all Contributors for the Covered Software under Section 2.1 of this License shall terminate.

5.3. In the event of termination under Sections 5.1 or 5.2 above, all end user license agreements (excluding distributors and resellers) which have been validly granted by You or Your distributors under this License prior to termination shall survive termination.

6. Disclaimer of Warranty

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. The entire risk as to the quality and performance of the Covered Software is with You. Should any Covered Software prove defective in any respect, You (not any Contributor) assume the cost of any necessary servicing, repair, or correction. This disclaimer of warranty constitutes an essential part of this License. No use of any Covered Software is authorized under this License except under this disclaimer.

7. Limitation of Liability

Under no circumstances and under no legal theory, whether tort (including negligence), contract, or otherwise, shall any Contributor, or anyone who distributes Covered Software as permitted above, be liable to You for any direct, indirect, special, incidental, or consequential damages of any character including, without limitation, damages for lost profits, loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses, even if such party shall have been informed of the possibility of such damages. This limitation of liability shall not apply to liability for death or personal injury resulting from such party’s negligence to the extent applicable law prohibits such limitation. Some jurisdictions do not allow the

exclusion or limitation of incidental or consequential damages, so this exclusion and limitation may not apply to You.

8. Litigation

Any litigation relating to this License may be brought only in the courts of a jurisdiction where the defendant maintains its principal place of business and such litigation shall be governed by laws of that jurisdiction, without reference to its conflict-of-law provisions. Nothing in this Section shall prevent a party's ability to bring cross-claims or counter-claims.

9. Miscellaneous

This License represents the complete agreement concerning the subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not be used to construe this License against a Contributor.

10. Versions of the License

10.1. New Versions

Mozilla Foundation is the license steward. Except as provided in Section 10.3, no one other than the license steward has the right to modify or publish new versions of this License. Each version will be given a distinguishing version number.

10.2. Effect of New Versions

You may distribute the Covered Software under the terms of the version of the License under which You originally received the Covered Software, or under the terms of any subsequent version published by the license steward.

10.3. Modified Versions

If you create software not governed by this License, and you want to create a new license for such software, you may create and use a modified version of this License if you rename the license and remove any references to the name of the license steward (except to note that such modified license differs from this License).

10.4. Distributing Source Code Form that is Incompatible With Secondary Licenses

If You choose to distribute Source Code Form that is Incompatible With Secondary Licenses under the terms of this version of the License, the notice described in Exhibit B of this License must be attached.

Exhibit A - Source Code Form License Notice

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <https://mozilla.org/MPL/2.0/>.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

Exhibit B - “Incompatible With Secondary Licenses” Notice

This Source Code Form is “Incompatible With Secondary Licenses”, as defined by the Mozilla Public License, v. 2.0.

Controversies surrounding the Mozilla Public License:

The main controversy surrounding the Mozilla Public License involves **user confusion and backlash over its updated terms of service for the Firefox browser, particularly concerning data use and licensing**.

Users expressed concern over vague language that seemed to grant Mozilla a broad license to their data, which they feared could lead to misuse.

Mozilla clarified the license was necessary for basic browser functions and did not grant them ownership or rights to misuse personal data.

Firefox terms of use and licensing

- **Broad license grant:** The new terms included a clause stating users grant Mozilla a "non-exclusive, royalty-free, worldwide license to use that information to help you navigate, experience, and interact with online content as you indicate with your use of Firefox".
- **User backlash:** This language led to significant user concern that Mozilla was claiming broad rights to their data for purposes beyond basic functionality, even for AI or monetization.
- **Mozilla's clarification:** Mozilla responded that the license is necessary for the browser to function, similar to how a website needs a license to process your login information or how a file upload needs a license for the server to process it.
- **No ownership or misuse:** The company clarified that the license does not give them ownership of user data or the right to use it in ways not outlined in the privacy policy.
- **Removal of "not selling data" statement:** Mozilla also removed explicit statements about not selling user data, which was done to comply with new California privacy laws that redefined "selling data," though they maintain the change was to improve transparency rather than indicate a shift in their practices, according to Mozilla's statements.

Other MPL controversies

- License abuse:** A separate controversy involved a situation where a project developer used the Mozilla Public License to file complaints and enforce copyright, leading to the shutdown of another open-source project.
- Debate on ethical use:** This incident sparked a debate within the open-source community about whether it is ethical to use the license in this manner to shut down a project, even if the claims were legally valid.
- "Exploitation" risk:** Another issue noted by [some developers](#) is that the file-level copyleft nature of MPL 1.1 might allow proprietary modules to be bundled without fully exposing underlying changes, a gap that could lead to exploitation by large companies.

There was a notable controversy in 2021 where the Pale Moon project team was accused of misusing the terms of the Mozilla Public License (MPL) to shut down a fork of their project named Mypal. The Mypal project aimed to continue supporting older Windows operating systems (XP and Vista), which Pale Moon had ceased supporting.

The dispute centered on the interpretation of the MPL's requirements for providing source code:

- The Allegation:** The Pale Moon team alleged that the Mypal developer (Feodor2) violated the MPL by not providing the source code in "the form of the work preferred for making modifications".
- The Counterargument:** Critics argued that the source code was effectively available (e.g., as diffs or in a slightly different repository format) and that the enforcement actions were overly aggressive and went against the collaborative spirit of FOSS (Free and Open Source Software).
- The Outcome:** The public nature of the conflict and the strict enforcement of the license terms over a technicality led to significant debate in the open-source community, with many perceiving the action as a way to eliminate a competing project.

While the MPL allows for clear legal enforcement of its terms, the incident highlighted the tension between strict license compliance and the community-oriented ideals often associated with open-source development.

Further Details:

The Pale Moon vs. MyPal controversy stems from a licensing and development dispute, primarily involving the use of Pale Moon's code and the subsequent fork of the MyPal browser. MyPal, initially a fork of Pale Moon for Windows XP, was forced to rebase on Firefox code due to licensing issues with Pale Moon's lead developer, leading to conflict and a divergence in the projects' development paths.

Origins of the conflict

- **Initial fork:** MyPal was initially a fork of Pale Moon, designed to continue support for older systems like Windows XP after Pale Moon moved to support more modern systems.
- **Licensing issues:** A dispute arose between the two projects regarding the licensing of the code.
- **Code rebase:** The conflict led MyPal to rebase its codebase on Firefox Quantum, causing a split from the Pale Moon project.

Subsequent developments

- **MyPal's path:** After the split, MyPal's code was based on Firefox's ESR68-78 codebase, rather than Pale Moon's own development.
- **Pale Moon's path:** Pale Moon continued to develop independently, focusing on its own features and requiring more modern systems, while MyPal continued to support legacy systems like Windows XP.
- **Branding and trademarks:** Pale Moon's lead developer has asserted ownership of the project's name, logo, and trademarks.

Key takeaways

- The controversy was primarily a licensing and code-licensing dispute, not a difference in technical direction.
- The two browsers ultimately diverged, with MyPal focusing on supporting older operating systems and Pale Moon focusing on modern systems.

(5) The BSD 3-Clause License

Open Source Software Licenses 101: The BSD 3-Clause License

<https://fossa.com/blog/open-source-software-licenses-101-bsd-3-clause-license/>

BSD 3-Clause License: The Basics

BSD licenses are permissive, meaning that they don't require very much from users of the licensed open source software. In fact, the requirements of the BSD 3-Clause License are very similar to those of the widely used [MIT License](#). And, like the MIT License, the BSD 3-Clause is short (albeit with a few important nuances).

Requirements

The BSD 3-Clause License has two key requirements of those who want to use the licensed code. They're actually the same as those of the MIT license. If you plan to copy, modify, or distribute any code licensed under BSD, you must include:

1. The full text of the license
2. The original copyright notice

It's easy to confuse these two, though. For a license like BSD or MIT, the copyright notice is "baked in" to the license. So, if you copy the entire license, you are actually copying both.

Each of these is a template license. In BSD, you will see this line at the beginning of the terms: "Copyright (c) \$YEAR \$OWNER, All rights reserved." The author of the code fills out this information before putting the license in the code repository. This is different from licenses like GPL or Apache 2.0, where the copyright notices are not in the base license document.

Using the Licensed Code

Users of BSD-licensed code may:

- **[Use the code commercially](#)**. Companies can include BSD'd code in proprietary software that they then sell to the public.
- **Modify the code**. Developers are permitted to update or rework the original code.
- **Distribute reworked versions or copies of the code**. An individual or company can make their updated version(s) of the code available to others, either commercially or privately.
- **Place warranty**. Users of the open source code are allowed to place a warranty on the licensed software.

The terms of BSD say that contributors to BSD 3 Clause-licensed code cannot be held liable for any damages resulting from modifications or updates to the original work. In addition, any person or company who makes use of code licensed under BSD 3 is prohibited from using

the name of the project or its contributors to promote their derivative work without written permission. In fact, this is the subject of the titular third clause of the BSD 3-Clause License.

(There are variations on the BSD license.)

The BSD 3-Clause License vs. Other Permissive Licenses

Below, we'll compare several other popular permissive licenses — the [MIT License](#) and the [Apache License 2.0](#) — to the BSD 3-Clause.

BSD 3-Clause License vs. the MIT License

These two licenses are very similar, with the key exception of the BSD 3's non-endorsement clause, which prohibits promotion of any derived work using the name of the license or its authors. In addition, the language of the [MIT License](#) is simpler and shorter.

BSD 3-Clause License vs. Apache License 2.0

The Apache License 2.0 differs from the BSD 3-Clause License in several key respects. The first is the Apache License's explicit grant of patent rights and defensive termination provision, which is not included in any of the BSD license variants. This aspect of the license provides legal protection and peace of mind to companies that make use of Apache-licensed code. In contrast, the BSD license is thought to grant some patent rights by implication, and the scope of that license is, at best, unclear. Secondly, the Apache License requires all users to list out significant changes and modifications to the original code. The BSD 3-Clause License has no such provision. Finally, the BSD license is compatible with every major [copyleft license](#), including GPL v2, while Apache 2.0 is arguably incompatible with GPL v2.

Authors tend to choose permissive licenses like the BSD 3-Clause because they're easy to implement, don't have many requirements, and offer flexibility. The BSD license is compatible with every major copyleft license, including GPL version 2. If an author wants their OSS code to reach the widest possible audience, a permissive license is the best option. However, an author may also want to ensure that the name of the project or its contributors aren't used without permission. In that case, the BSD 3-Clause License in particular has the advantage over MIT or Apache. (Although it's worth noting that trademark law is a stronger protector than the BSD trademark clause.)

Controversies surrounding the BSD license and its variants

Controversies surrounding the BSD license and its variants primarily stem from its permissive nature, which allows for code to be used in proprietary software without requiring the modified source code to be shared, and the historical "advertising clause" that required attribution in all advertisements. The historical advertising clause, removed in the 3-clause BSD, was problematic because it made the license incompatible with the GNU GPL and created an unwieldy attribution burden when combined with other software. A separate concern is the lack of an explicit patent grant, which can expose users to legal risks, although this is a separate issue from the advertising and non-endorsement clauses.

Historical "advertising clause"

The problem: The original 4-clause BSD license included an "advertising clause" that required every advertisement for software using the code to include a specific acknowledgment to the University of California, Berkeley.

The consequence: Developers began to replace the original text with their own organization's name, leading to escalating attribution requirements that made the license impractical for large projects where multiple components were combined.

The solution: This clause was removed in 1999, creating the 3-clause (or "New BSD") license. A further simplification led to the 2-clause (or "FreeBSD") license, which removes both the advertising and non-endorsement clauses.

Permissive nature and "contribution" concerns

The debate: The license's permissiveness allows for code to be incorporated into proprietary, closed-source software without any obligation to share the modified source code back with the original project.

Concerns: Some in the open-source community argue that this can lead to "leeching," where corporations use open-source code for commercial gain without contributing back any improvements.

Counterargument: Others defend the license as its intended use, as the goal was to encourage broad adoption, and the creators were content with the code being used widely, even in commercial products, say Reddit users.

Lack of explicit patent grant

The issue: Unlike some other licenses (like the Apache 2.0), the BSD license does not contain an explicit grant of patent rights from the contributor to the user.

The risk: This omission could potentially expose users of BSD-licensed code to patent infringement claims from individual contributors, which can be a deterrent for businesses in patent-sensitive industries.

The solution: Developers may use licenses that provide clearer patent protection or avoid the risk by not using BSD-licensed code in certain contexts.

What's the issue with the BSD license? : r/linuxquestions - Reddit

Aug 10, 2020 — If the above is confusing, let me put it into practical terms: * With the GPL any modified version that is distributed...

Permissiveness and "BSD exploitation"

- The permissive nature of the license means developers can use BSD-licensed code in proprietary software without sharing their own code.
- This has led to criticism that companies can "exploit" free code without giving back to the open-source community, potentially undermining the financial sustainability of open-source projects.

Lack of patent grant

- Unlike licenses such as the [Apache License 2.0](#), the BSD license does not include an explicit patent grant.
- This omission can expose users to legal risks, as they could face patent infringement claims from individual contributors who hold patents on parts of the code, even if they are using it in a commercial product.

The "advertising clause"

- The original BSD license included an "advertising clause" that required all advertising materials for a derivative work to include a specific acknowledgment of the original source.
- This clause became impractical as projects grew to include many components from different sources, creating lengthy and cumbersome attribution requirements.
- As a result, this clause has been removed in newer versions, such as the 2-clause BSD and 3-clause BSD, which are now more commonly used.

Conclusions:

Open source licenses are designed to allow the free exchange of software, code and developer resources, without the opaqueness and restrictions of closed-source, proprietary licensing types.

Copyleft is an important term for understanding how open source code and software projects differ from proprietary, copyrighted code and software projects.

Issues have arisen when differing licensing types have resulted in differing interpretations of how much sharing and forking is permitted, and how to provide attribution where it's required. Some abuses have included taking open source code into projects which are then made proprietary. Some issues also involve developers withdrawing their code from open source projects despite the requirement to keep the code available. (The Linux kernel developers had such a controversy.)

The idea behind open source development is not to let everyone have cost-free access to code or products which take developer time and effort to produce. Instead, the main idea is to allow code to be seen and evaluated by everyone, and used in a variety of projects. There are ways other than charging money for code to create a revenue stream, and leaving a fork of a project open can provide a good test bed for a paid, proprietary product or service. Even if the code and the software are free, support options can still be paid, providing a further revenue stream.

From an end user perspective, open source software provides a low cost or no cost way to try out software, and to have a basic library of software for everyday tasks.

Personally, while I am a heavy user of open source software, I do pay for utilities like backup programs, the Windows operating system, and other specialized software for which I find open source alternatives inadequate.

Now is the time for discussion, comments, opinions and questions.

I invite those with better legal and technical knowledge than myself to fill in the details of how open source licensing works.

-- Bob Primak --

-- for the Lexington Computers and Technology Users Group (LCTG) --

-- Wed., November 19, 2025 --

-- Revised Friday, November 14, 2025. 08:13:36 AM EST