WhiteSource

2020

The Complete Guide for
Open Source Licenses

BSD

Eclipse 1.0

Microsoft
Public

BSD 2

LGPLv2.1

BSD 3

GPLv2

GPLv3

Apache-2.0

MIT

Contents

WHAT IS AN OPEN SOURCE LICENSE?

Software is considered as a work of art in the eyes of the law, similar to a painting, a poem, or a movie script. As such, even if you share your art with the rest of the world to see, it does not mean that anyone can download it and use it as they please. Instead, they have a legal obligation to receive approval from the author who has copyrights on the software.

An open source license is a legal and binding contract between the author and the user, declaring that the software can be used in commercial applications in certain conditions. The license is what turns software components into open source components, and allows developers to use that software — as long as they keep to the specific terms and conditions as laid out in the license.

Each open source license states what users are permitted to do with the software components, their obligations when using them, and what they cannot do as per the terms and conditions. This might sound simple, but there are over 200 open source licenses out there, varying in complexity and requirements. It's up to each developer or organization to choose which components are most compatible with their policies and existing software to ensure that they remain compliant.

TYPES OF OPEN SOURCE LICENSES: COPYLEFT VS. PERMISSIVE

Open source licenses can be divided into two main categories: copyleft and permissive. This division is based on the requirements and restrictions the license places on users.

Copyright is a law that restricts the right to use, modify, and share creative works without the permission of the copyright holder. When an author releases a program under a copyleft license, he or she makes a claim on the copyright of the work and issues a statement that other people have the right to use, modify, and share the work as long as the reciprocity of the obligation is maintained. In short, if they are using a component with this kind of open source license, then they too must make their code open for use by others as well.

A permissive open source license is a non-copyleft open source license that guarantees the freedom to use, modify, and redistribute, while also permitting proprietary derivative works. Permissive open source licenses, lovingly referred to as "Anything Goes", place minimal restrictions on how others can use open source components. That means that this type of license allows varying degrees of freedom to use, modify, and redistribute open source code, permitting its use in proprietary derivative works, and requiring nearly nothing in return in regards to obligations moving forward.

It's important to note that there are no good or bad licenses, and that no one license is better than another. Anyone can create an open source license that suits their fancy, which is the reason that there are so many out there.

Copyleft	Weak Copyleft	Permissive
(logos)	CDDL MPL eclipse	MIT APACHE BSD

This could make choosing an open source license complicated business, especially for those of us who are not well versed in the law. In order to help narrow down the decision and make sense of it all, the OSI (Open Source Initiative) put together a list of approved licenses, best suited for commercial use. The list consists of a little over 80 open source licenses that are most commonly used.
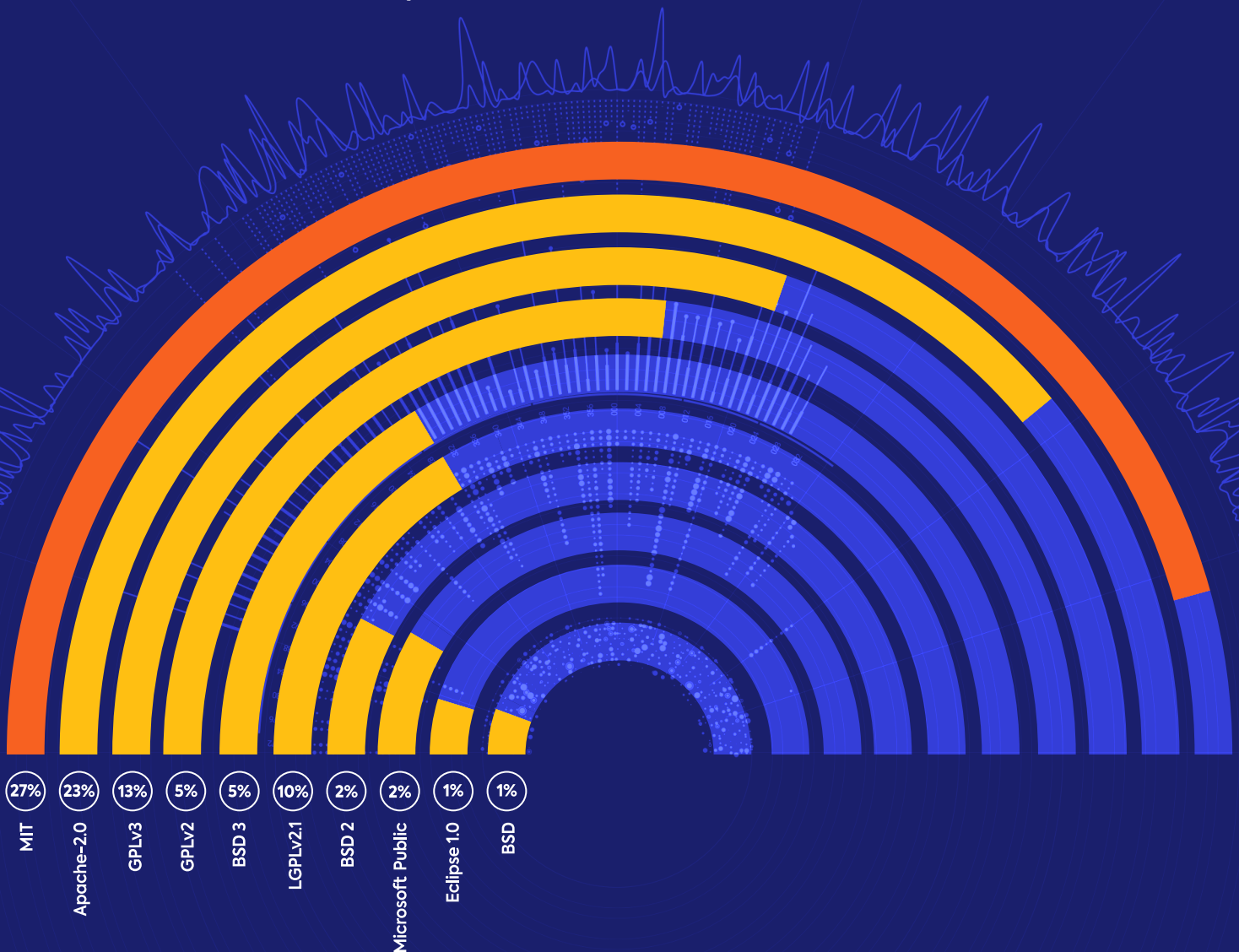
TOP
OPEN SOURCE LICENSES IN 2019

As this year comes to a close, it is a good time to take a look at the trends of open source license usage in 2019 and compare them to previous years.

Our research team has collected information from the WhiteSource database, which includes over 4 million open source packages and 130 million open source files, covering over 200 programming languages, to learn which were the most popular open source licenses in 2019, compared to previous years. Results show that use of permissive open source licenses continues to rise, while usage of copyleft licenses, and the GPL-family in particular, continues to decrease.

2019 Open Source Licenses: Distribution

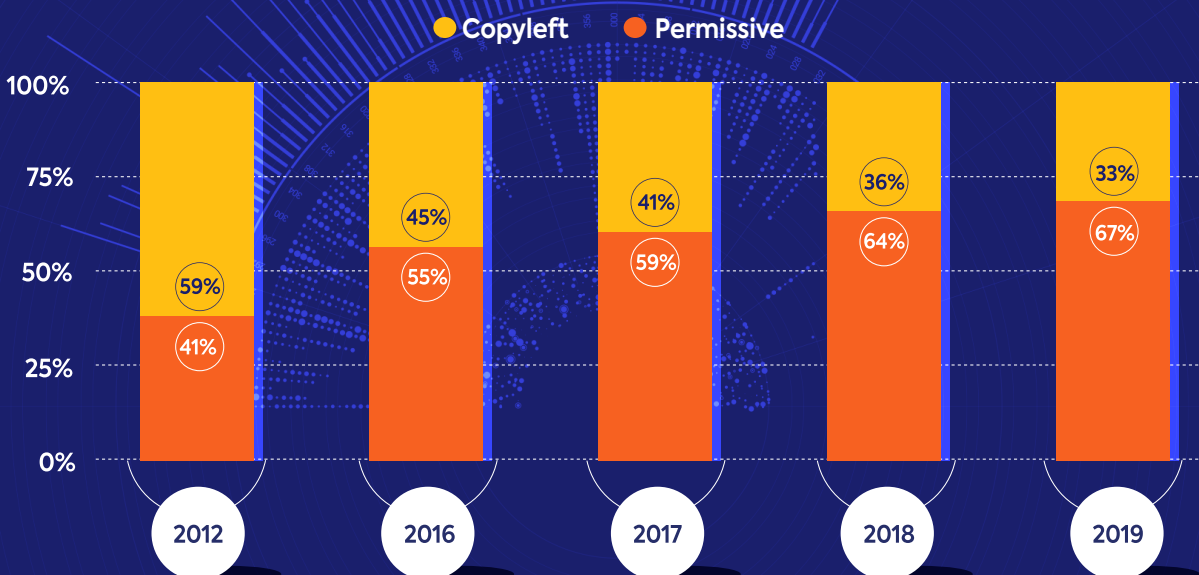| 27%	23%	13%	5%	5%	10%	2%	2%	1%	1%
MIT	Apache-2.0	GPLv3	GPLv2	BSD 3	LGPLv2.1	BSD 2	Microsoft Public	Eclipse 1.0	BSD

PERMISSIVE OPEN SOURCE LICENSES CONTINUE TO TREND

Use of permissive open source licenses is still on the rise, continuing the trend we saw in 2018. MIT and Apache 2.0 licenses once again take first and second place in our list of top 10 most popular open source licenses of the year, both rising by one percent as compared to last year.

Permissive open source licenses place minimal restrictions on how others can use open source components. This type of license allows varying degrees of freedom to use, modify, and redistribute open source code, permitting its use in proprietary derivative works, and requiring nearly nothing in return.

According to this year's data, 67% of open source components have permissive licenses. That's a 3% rise from last year's 64%. Only 33% of the top 10 most popular open source licenses are copyleft, compared to 36% last year and 59% in 2012. The numbers show that developers and organizations continue to prefer permissive licenses.

Permissive vs. Copyleft Open Source Licenses Over Time

● Copyleft   ● Permissive

| Year	Copyleft	Permissive
2012	59%	41%
2016	45%	55%
2017	41%	59%
2018	36%	64%
2019	33%	67%

This can be explained by the continuous rise in open source usage. Open source has become mainstream, and the open source community is embraced and supported by the commercial software community. With companies like Microsoft and Google standing behind some major open source projects, the "Us" vs. "Them" mentality that ruled in the early days of open source is long gone. In the interest of this widespread cooperation, and encouraging open source usage, permissive licenses are winning.

Users, in turn, are choosing the components with the licenses that have fewer strings attached. Open source components with permissive licenses seem to offer them all a simple solution, by minimizing the challenges of open source licensing compliance for legal departments.

STILL AT THE TOP: THE MIT OPEN SOURCE LICENSE

The MIT license remains at the top of the popular open source licenses list, at 27%. This shouldn't come as a surprise, as it's been trending on GitHub since 2015. Ben Balter, attorney, open source developer, and Senior Product Manager at GitHub, said then that developers choose the MIT license because "It's short and to the point. It tells downstream users what they can't do, it includes a copyright (authorship) notice, and it disclaims implied warranties (buyer beware). It's clearly a license optimized for developers. You don't need a law degree to understand it, and implementation is simple."

Top Open Source Licenses 2019

| | MIT | Apache-2.0 | GPLv3 | GPLv2 | BSD 3 | LGPLv2.1 | BSD 2 | Microsoft Public | Eclipse 1.0 | BSD |
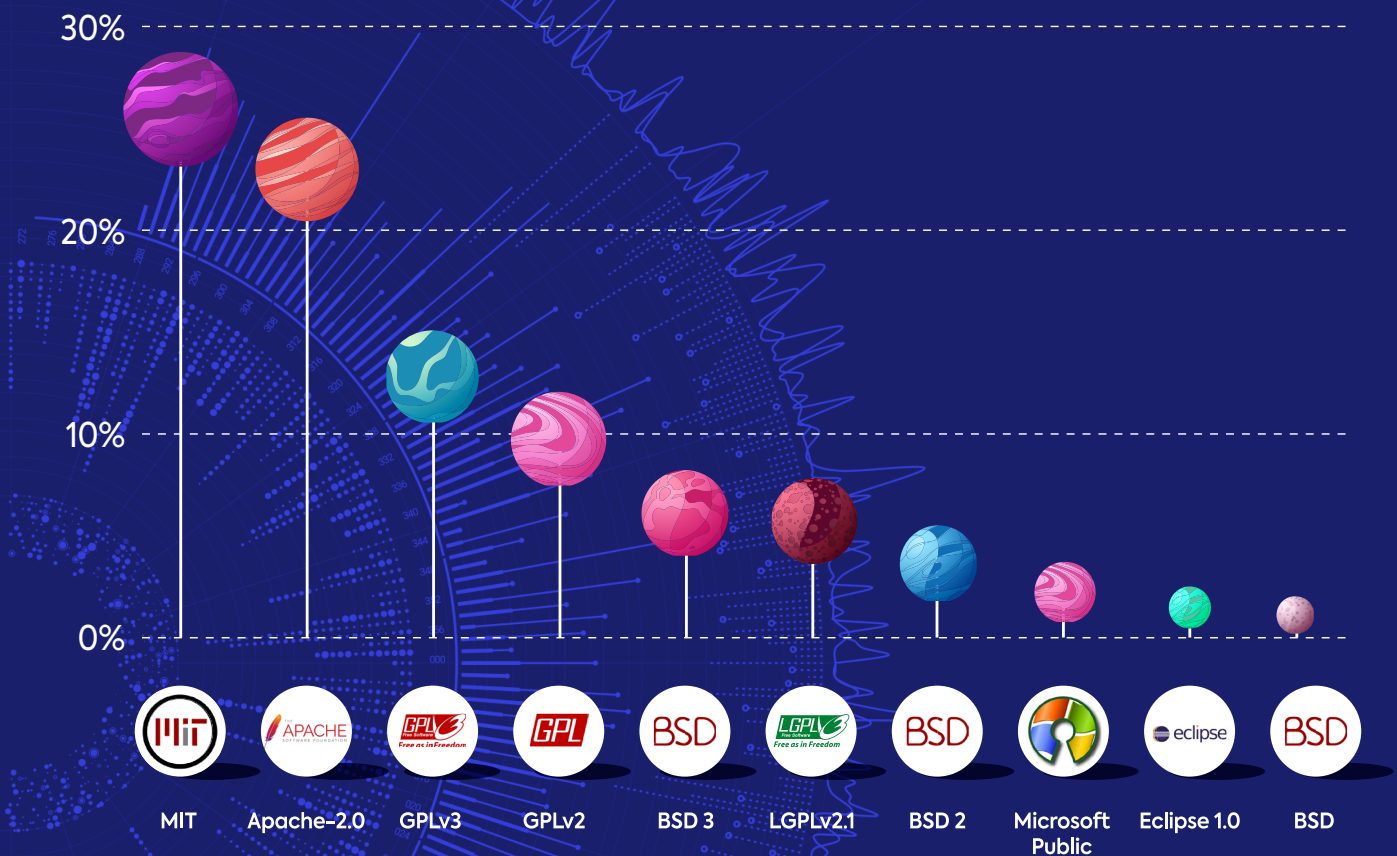
According to GitHub's choosealicense.com, The MIT license "lets people do anything they want with your code as long as they provide attribution back to you and don't hold you liable." Two years ago Facebook very publicly replaced the contentious React license with an MIT license.

Other popular open source projects licensed under MIT are Angular.js, rails, and .NET Core.

THE APACHE 2.0 LICENSE CONTINUES TO DOMINATE

Two years ago, when we rounded up the numbers for 2017, the permissive Apache 2.0 license shook things up by making a leap to 2nd place on our top 10 open source licenses list, replacing the copyleft GPL 3.0 license. This year, the rise in Apache 2.0's popularity continues, as it gains another one percent and comes in strong at second place with 23%.

According to GitHub's choosealicense.com, The Apache 2.0 license's main conditions require preservation of copyright and license notices, providing an express grant of patent rights, and allowing licensed works, modifications, and larger works to be distributed under different terms and without source code. Apache 2.0 is the license for quite a few popular open source projects, including Kubernetes, Swift, and PDF.js — to name a few.

Apache-2.0 Popularity Over Time

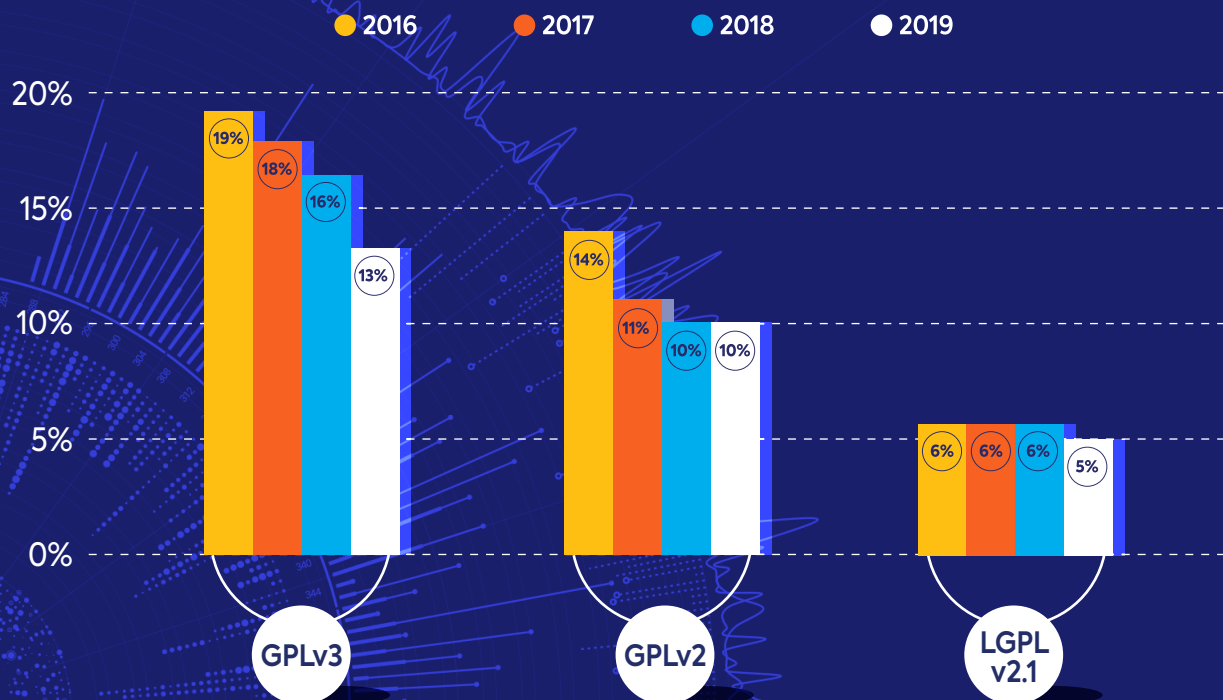|			
2016	2017	2018	2019

Scale: 0% to 25%
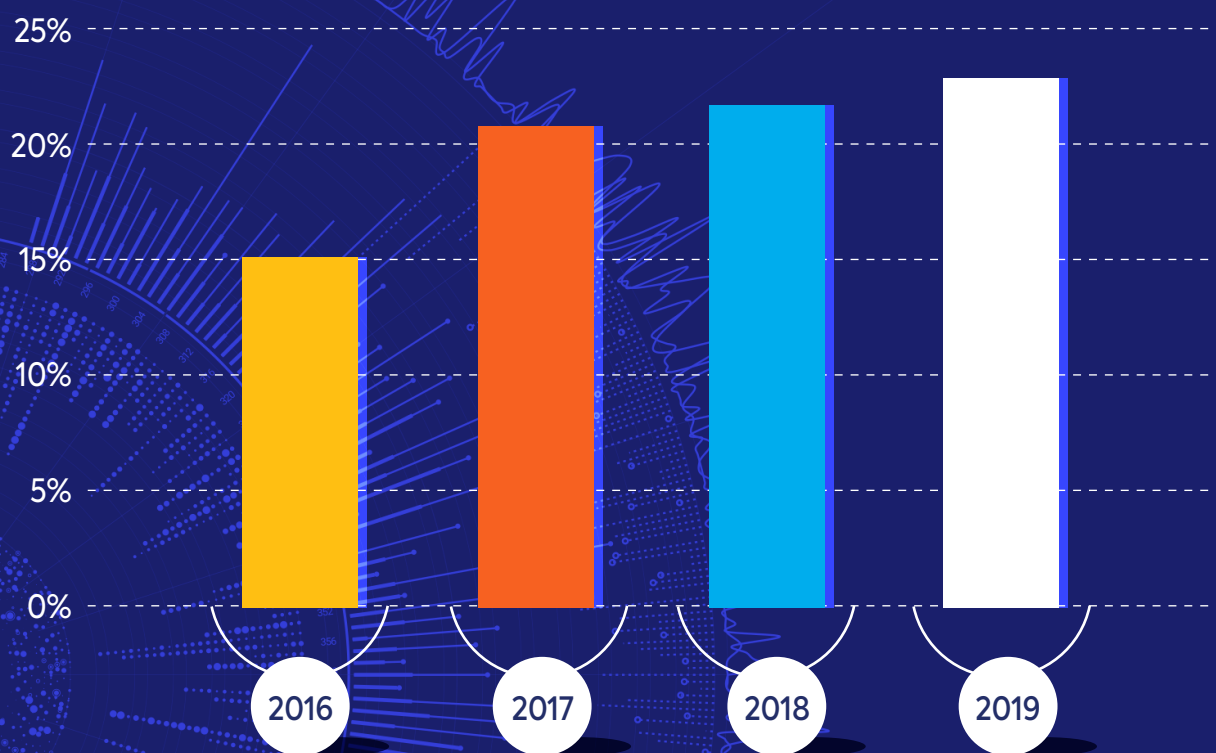
THE SLOW DECLINE OF COPYLEFT LICENSES

GPLv3 and GPLv2 both took another hit this year. GPLv3 still came in at number three, but lost another three percent, coming in with 13%, compared to 16% in 2018. GPLv2 also kept its fourth place, remaining with 10% from 2018.

This year GPL v3.0, GPL v2.0, and LGPLv2.1, which all came in at top 10, got a combined 28% out of all top 10 licenses, which marks another significant decrease in popularity for the GNU GPL family of licenses. We expect this trend to continue in years to follow.

GPL License Family Popularity Over Time

● 2016   ● 2017   ● 2018   ● 2019

	GPLv3	GPLv2	LGPL v2.1
2016	19%	14%	6%
2017	18%	11%	6%
2018	16%	10%	6%
2019	13%	10%	5%

The GPL was a trailblazer at the start of the open source revolution and is a prime example of the copyleft or viral license. When users incorporate a component licensed under one of the GPL licenses, they must release its source code, as well as the rights to modify and distribute the entire code. Not only that, but they are also required to release their source code under the same GPL license.

Back in the early days of open source adoption, the GPL license posed a real conundrum for businesses thinking about adopting open source or participating in the open source community. Many chose a dual licensing approach in an attempt to bridge the gap between the GPL license and their commercial needs.

While mountains of code have been open sourced since then, the numbers show that GNU GPL is avoided by many commercial entities that are taking a much more central place in the open source community year over year. With the wealth of open source licenses out there, it appears users are choosing the more permissive ones, that contain fewer requirements and restrictions.

OPEN SOURCE LICENSING IN 2020: WHAT DOES THE FUTURE HOLD?

While they didn't make it to our 2019 list of top ten open source licenses, or even top 20, this past year we continued to hear rumbles from the community about the open source licensing approaches. Changes to licensing in leading open source projects like Mongo DB and Redis are reminders that as open source usage grows, organizations are in the process of figuring out how to both embrace the open source community and update their business models to stay ahead.

As Michael DeHaan, the creator of the extremely popular Ansible has pointed out, open source developers and open source users may require a new open source licensing solution to ensure that the community continues to evolve.

One thing is certain from both the open source licensing headlines that we saw this past year, and from our research: both developers and commercial organizations are choosing the open source components that enable them to create products that can thrive in the open source ecosystem. The community is doing its best to make sure open source is easy to adopt and comply by. It's up to organizations to keep up their end of the bargain and make sure that they know which open source licenses they are using and that they are keeping up with their requirements.

TOP OPEN SOURCE LICENSES

LICENSES

Cheat Sheet

GNU General Public License (GPL)

The GNU's General Public License used to be the most popular open source license around. Richard Stallman created the GPL to protect the GNU software from becoming proprietary, and it is a specific implementation of the "copyleft" concept that he came up with.

Since GPL is a copyleft license, any software that is written based on any GPL component must be released as open source. The result is that projects that use any GPL open source component (regardless of its percentage in the entire code) are required to release the complete source code and all of the rights to modify and distribute the entire code.

Here are the answers to the most common questions regarding GPL:

1

What are the GPL terms and conditions?

When you use a GPL component in your software your entire software is considered a 'work based on' GPL, and therefore you are not allowed to claim patents or copyright on the software. Moreover, you are obligated to display a copyright notice, disclaimer of warranty, intact GPL notices, and a copy of the GPL.

You are not allowed to change the license or introduce additional terms and conditions.
You are under the reciprocity obligation, which means you are obligated to release the source code and all of the rights to modify and distribute the entire code.

2

Is GPL enforceable?

GPL is enforceable as it's essentially a copyright license. The copyright holders of the GPL software can choose to enforce the GPL on the distributed or derivative works of the software.
For example, the FSF holds the copyrights on many pieces of the GNU system, such as the GNU Compiler Collection. As the copyright holder, it can enforce the copyleft requirements of the GNU General Public License (GPL) if copyright infringement occurs on that software.

3

What is the difference between the GPLv2 and the GPLv3?

There has always been some confusion regarding what constitutes a 'work based on' another work, which in turn triggers the GPL reciprocity obligation. The FSF tried to add more clarity to GPLv3 as to when the reciprocity obligation is triggered. The FSF even wrote a new GPL license, the Affero license, to address a specific confusion referred to as the "ASP loophole".

The FSF also wanted to increase the compatibility of the GPLv3 with other licenses. To combine two codes into a larger work, both the programs must permit it. If such rights are granted by both the programs' licenses, they are compatible. By making the GPLv3 more compatible, the FSF expanded development options.

The third difference between the two versions is that the GPLv3 was written in an attempt to increase usage worldwide. The language used in GPLv3 to describe the license rights was modified to ensure that international laws will interpret it as the FSF intended, unlike the language used in GPLv2, which is considered very US-centric. GPLv3 also allows developers to add local disclaimers, which helps to increase its usage outside the US.

4

Can you mix a GPL License with other licenses?

It's often believed that the code covered by the GPL license cannot be mixed with code covered by other open source software licenses. While restrictions exist, it is actually possible under both GPLv2 and GPLv3. The new language used in the GPLv3 establishes this even more clearly. The FSF has stated explicitly that GPLv3 is compatible with the Apache 2.0 license. There is, however, an issue with the original BSD license as it imposes a specific requirement that is not in the GPL (the requirement on advertisements of the program).

GPL GNU GPL with Classpath Exception

In some cases, the author can choose to the release code under the GNU GPL with a classpath exception. Here are the answers for the most common question regarding the GPL classpath exception:

1 **What is the GNU Classpath exception?**

The GNU GPL requires that every work based on the program — that is, every derivative of the original program or any modifications one introduces to it — be subject to the GPL. As such, it may cover your original code if you combined it with a GPL module.

The classpath exception permits linking a GPL library with an independent module "which is not derived from or based on the library", without subjecting the resulting program to the GPL. The independent module can be your own proprietary program. Therefore, the classpath exception enables the use of GPL'ed license components in a certain way without risking the integrity of your Intellectual Property.

In addition, the resulting executable can be copied and redistributed under a license of your choice – as long as you meet the terms and conditions that govern the existing modules you're using.

Essentially the classpath exception protects you from having to release your project under the GNU license, if you link to a GPL with classpath exception library — protecting you from having to publically open your entire source code.

2 **How should I link a GPL with classpath exception components to my software?**

You can either link the modules statically or dynamically. The GNU GPL classpath exception permits both methods.

3 **Do I have to extend the classpath exception downstream?**

If you use the GPL library as is, then you must. However, if you modify the GPL with classpath exception library, you may choose whether to extend the exception to your modified library. This is not compulsory. If you don't want to extend the exception, you don't have to include the exception statement in your modified library.

4 **Who can apply the classpath exception to a library?**

Only the copyright owner — usually, the developer of the library — can choose to release his or her program under the GPL with a classpath exception.

5 **What's the difference between the GPL with a Classpath Exception and the GNU GPL as such?**

The GPLv3, is a major revision of the GPLv2. It introduced changes to the license terminologies, discussed patent rights in detail, addressed compatibility issues with other open source licenses, and more. Like the GPLv2, it subjects any work that is derived from the GPL program to the terms and conditions of the GPL, in what is often called the "reciprocal" nature of the GPL.

The GNU GPL with the Classpath exception is a special case of the GNU GPL that allows developers to link to GPL classpath exception licenses library to different programs irrespective of their licenses, without subjecting the "derived" result to the terms and conditions of the GPL.

The Apache License

The Apache License is an open source software license released by the Apache Software Foundation (ASF). It's a popular and widely deployed license backed by a strong community. The Apache License allows you to freely use, modify, and distribute any Apache licensed product. However, while doing so you're required to follow the terms of the Apache License.

Here are the answers for the most common questions regarding the Apache license

1 What are the Apache License terms and conditions?

The Apache License is a permissive open source software license — so you can release your modified version of the Apache-licensed product under any license of your choice. You can freely use, modify, distribute and sell a software licensed under this license without worrying about its use: personal, internal or commercial.

This license explicitly grants rights to users that can be applied to both copyrights and patents, unlike other permissive licenses that are applicable only to copyrights and not patents. The rights given are perpetual, worldwide, irrevocable, but also non-exclusive — so you can use the licensed work, and so can anyone else. If you redistribute software with any Apache licensed components, you must include a copy of the license, provide a clear Apache License attribution, and add modification notices to all the files that you modify.

You can choose to release the modified or derived products under different licenses, the unmodified parts of the software, however, must retain the Apache License, and you cannot name your modified version in any way that suggests that the final product is endorsed or created by the ASF.

Additionally, if you want to add a copyright statement about all the modifications that you've done to any Apache licensed software; you are free to do so. Since the Apache License doesn't require you to release the modified code under the same license, you can choose to add specific license terms and conditions that govern how others use, reproduce, or distribute your modified code.

2 What is the difference between the different versions?

The Apache Group (later named the Apache Software Foundation) released the first version of its license in 1995, but it's rare that you'll come across components that still carry this license.

In 2000, when Berkeley accepted the argument put to it by the Free Software Foundation and retired their advertising clause from the BSD license and formed the modified BSD license, Apache did likewise and created the Apache License version 1.1.

Removing the advertising clause meant that the advertising materials of the derivative works of any Apache licensed product were no longer required to include the Apache License attribution. It became ok to include the attribution in the documentation alone.

In 2004, the ASF decided to depart from the BSD model a little more radically and produced the Apache License version 2.0 by granting patents rights and defining solid definitions of the concepts it uses to make it more coherent.

3 What is the difference between the Apache License 2.0 and the GNU GPL?

The GNU GPL is a copyleft license. So software that uses any GPL-licensed component has to release its full source code and all rights to modify and distribute the entire code. The Apache License 2.0 doesn't impose any such terms. You're not forced to release your modified version. Besides, you can choose to release your modified version under a different license (however, you're required to retain the Apache License for the unmodified parts of the code).
a specific requirement that is not in the GPL (the requirement on advertisements of the program).

4 ## Is the Apache License compatible with the GNU GPL?

Apache License 2.0 is compatible with GPLv3, so you can freely mix the code that's released under these two licenses. The resulting software, however, must be released under GPLv3.

However, the Apache License 2.0 in incompatible with GPLv2 due to the restriction that terminates the grant of patent rights if the license sues over patent infringement.

Previous Apache versions, being heavily based on the BSD license, are compatible.

5 ## What is the difference between Apache License 2.0 and BSD?

The BSD license is another highly permissible license that allows you to modify and redistribute software licensed under the BSD license as you like. Earlier versions of the Apache License were identical to the original (and later the modified) BSD licenses, but Apache License 2.0 sets them apart. The key differences between the two licenses are:

- Explicit grant of patent rights: Apache License 2.0 explicitly lays down the grant of patent rights while using, modifying or distributing Apache licensed software; it also lists the circumstances when such grant gets withdrawn.

- Clear definitions of the used concepts: Apache License 2.0 explicitly defines all the terms and concepts that it uses. This leaves little scope for ambiguity.

- Reusable without rewording: Apache License 2.0 can be easily used by other projects without any rewording in the license document itself.

Microsoft Public Licenses (Ms-PL)

The Microsoft Public License is a free and open source software license released by Microsoft, which wrote it for its projects that were released as open source.

Here are the answers for the most common question regarding Ms-PL license:

1 **What are the Microsoft Public License (Ms-PL) terms and conditions?**

You are free to reproduce and distribute original or derivative works of any software licensed under the Ms-PL license. However, you may not use any contributors' names, logos, or trademarks when you do so. The Ms-PL protects the authors by explicitly not offering any express warranties or guarantees for using your code, so the author is not liable if the code doesn't work well in some cases.

When you distribute software (or its portion) under the Ms-PL, you're not required to distribute its source code. You may do so if you want to, but you're not obliged. However, you're required to retain all copyright, patent, trademark, and attribution notices that are originally present in the software. Additionally, if you distribute any portion of the software in its source code form, you may do so only under the Ms-PL by including a complete copy of this license with your distribution. If you distribute any portion of the software in its compiled or object code form, you may only do so under any other license that complies with the Ms-PL.

It is important to note that the Ms-PL terms and conditions document is very short, concise and written in a very clear language. Microsoft wanted to be very clear and direct with the open source community, which also helps the adoption rate (similar to the BSD license).

2 **Is Microsoft Public License (Ms-PL) considered copyleft?**

A Copyleft license offers the right to distribute modified and derivative versions of a program, provided that the same rights and freedoms are preserved for downstream recipients of those modifications and derivatives. Distribution of Ms-PL'ed software or its portion in its source code form, can only be done under the Ms-PL license. When you distribute the Ms-PL'ed software in compiled or object code form, the Ms-PL license lets you do so only under "a license that complies with" the Ms-PL. Hence, the Copyleft effect of Ms-PL is clear when choosing to distribute source code version of the modified or derivative Ms-PL software.

It seems that when distributing compiled or object code versions of modified or derivative Ms-PL software, the same rights and freedoms need not be passed through to downstream recipients, even though the Ms-PL text is not entirely clear on this point. This interpretation is supported by Microsoft, the steward of Ms-PL, who maintains that one may distribute compiled or object code versions of Ms-PL'ed software under terms of his or her choosing, which must not grant downstream recipients more rights (but can grant them less rights) to the Ms-PL'ed software than are granted to that person.

3 **What is the difference between Microsoft Public License (Ms-PL) and the Microsoft Reciprocal License (Ms-RL)?**

The Ms-RL license is a copyleft license that is more restrictive than the Ms-PL. It allows you to modify and distribute any Ms-RL'ed component as long as the modified source files are included and licensed under the Ms-RL.

However, you can license the other files of the software, which are entirely your own work, under any other compatible license you may choose.

4   **Is Microsoft Public License compatible with GNU GPL?**

No. The Microsoft Public License is not compatible with the GNU GPL. The incompatibility between GPL and Ms-PL stems from the fact that GPL is much more restrictive than the Ms-PL, for example, GPL's requirement to distribute the source code does not correspond with the Ms-PL clause that enables compiling the program without distributing the source code. Even the Ms-RL, which is a copyleft license, is not compatible with GPL.

It is believed that Microsoft deliberately crafted its open source licenses to be incompatible with the GPL since, like many other commercial companies, it disliked the fact that if you submit a code under this license, your code can then be taken into a proprietary black hole by someone else.

5   **How can you use a component licensed under the Microsoft Public License in your commercial project?**

The BSD license is another highly permissible license that allows you to modify and redistribute software licensed under the BSD license as you like. Earlier versions of the Apache License were identical to the original (and later the modified) BSD licenses, but Apache License 2.0 sets them apart. The key differences between the two licenses are:

BSD Berkeley Software Distribution (BSD)

BSD Licenses, or the original BSD License and its two variants - the Modified BSD License (3-clause), and the Simplified BSD License/FreeBSD License (2-clause) are a family of permissive free software licenses.

Here are the answers for the most common questions regarding BSD licenses:

1 ### What are the terms and conditions of the BSD Licenses?

The BSD License lets you freely modify and distribute your software's code in the source or binary format as long as you retain a copy of the copyright notice, list of conditions, and the disclaimer.

The original BSD License or the 4-clause BSD License also contains an advertising clause and a non-endorsement clause (detailed explanation about these clauses are offered in the following questions). The modified BSD License or the 3-clause BSD License was created by removing the advertising clause from the original BSD License. Further, the FreeBSD version or the 2-clause BSD License was created by removing the non-endorsement clause from the modified BSD License or the 3-clause BSD License.

In addition, the resulting executable can be copied and redistributed under a license of your choice – as long as you meet the terms and conditions that govern the existing modules you're using.

Essentially the classpath exception protects you from having to release your project under the GNU license, if you link to a GPL with classpath exception library — protecting you from having to publically open your entire source code.

2 ### What is the difference between the original 4-clause BSD License and the Modified 3-clause BSD License?

The advertising clause from the original BSD License requires users to acknowledge the original authors of any used BSD-licensed components in all advertising materials mentioning features or use of their software. This clause was criticized for several reasons. It also made the original BSD License incompatible with the GNU GPL.

Basically, the BSD License authors expected developers to include the acknowledgment in their copyright notices. However, due to misunderstanding the license (and even with malicious intent, in some cases), developers started replacing the acknowledgment text by adding their own or their organizations' names. This led to situations where developers were required to list too many attributions, each corresponding with a BSD-licensed component used in their software. Following the feedback, in 1999, the advertising clause that appears in the original BSD License was removed to create the Modified 3-clause BSD License.

3 ### What is the difference between the Modified 3-clause BSD License and Simplified 2-clause BSD License?

The Simplified 2-clause BSD License further toned down the 3-clause BSD License by removing the non-endorsement clause. This clause ensured that users could not make it sound like their software was endorsed by any of the acknowledged developers or organizations.

It also introduced a disclaimer about views and opinions expressed in the software to be those of the authors and not of the FreeBSD project.

4   ## Are the BSD Licenses compatible with GPL?

As mentioned earlier, it was the advertisement clause in the original BSD license that made it incompatible with the GNU GPL. The newer versions of the original BSD licenses, i.e., the 3-clause and the 2-clause variants are compatible with GPL.

5   ## What are the differences between the Modified BSD License and the MIT License?

MIT is one of the most permissive free software licenses. Basically, you can do whatever you want with software licensed under the MIT license - as long as you add a copy of the original MIT license and copyright notice to it. Its simplicity is the reason behind its high adoption rate among developers.

If you use the MIT license, you can use it as-is. But if you use any of the BSD licenses, you're still required to modify the license copy to suit the project at hand. In addition, the Modified BSD License, thanks to its non-endorsement clause, protects you from having your name involved in a project unless that's what you want.

CDDL Common Development and Distribution License (CDDL)

Common Development and Distribution License (CDDL) is an open source license published by Sun Microsystems to replace the Sun Public License (SPL). The CDDL license is considered by Sun (now Oracle) to be SPL version 2. It is inspired by the Mozilla Public License (MPL).

Sun used to release its free software and its open source projects under its Sun Public License (SPL) before it turned to rely upon the CDDL in 2004. CDDL is often dubbed as a cleaned-up version of the MPL and is made to facilitate reusability.

Here are some common questions about CDDL:

1 **What are the Common Development and Distribution License (CDDL) terms and conditions?**

You're free to reproduce and distribute any original or derivative works of any software licensed under the CDDL. However, you must not remove or make any changes to any copyright, patent or trademark notices in the software. You must also retain any notices of licensing or any descriptive text giving attribution to any contributor or the initial developer.

When you distribute your software in an executable form (any form other than source code), you are required to make the source code available as well under the CDDL. The executable form may be released under the CDDL or any CDDL compatible licenses.

The source code that you have to make available includes your contributions as long as they are an addition to, deletion from or modification of the contents of a file containing the original software — or new files that contain parts of the original program. That means that if your additions are made in separate and independent files that do not contain the original code, you do not have to release it under the CDDL. You may do that if you choose to, but you're not obligated.

In addition, you must include a copy of the CDDL with any source code that you distribute. For each modification that you make, you must identify yourself as the modifier by including a notice in your modified files.

2 **Is the CDDL considered copyleft?**

The CDDL is considered a weak copyleft license. A copyleft license, like the GNU GPL, the MPL or the Eclipse License, requires that you give down-the-stream users of the program the same rights that you received. For that purpose, you are required to distribute the program — including any modified and extended versions of it - under the same license. This means that using such a copyleft licensed component in your code will require you to release your entire program as open source. Essentially, it means you distribute the original or modified software under the same license that it originally carried.

The CDDL requires you to release the source code of only the CDDL licensed components that you use or modify in your code under the CDDL. If you distribute your software in its executable form, you must include the source code form but the executable can be distributed either under the CDDL or under a compatible license.

3

Does the CDDL grant patent rights?

Yes, it does. Any contributor grants you the right to use the patents that his contribution embodies. CDDL takes a very clear stand on patents — you can use, modify, and redistribute CDDL licensed components without any concerns about any patents that the code contributors might hold on the contributed technology. The CDDL discourages patent litigation against developers by terminating the usage rights of anyone who initiates a patent claim against a developer about the code that they have contributed.

4

What is the difference between CDDL version 1.0 and CDDL version 1.1?

CDDL version 1.1 was submitted a year after the first draft in early January 2005. It includes some corrections that prevent the CDDL from being in conflict with European Copyright law and to allow single developers to use the CDDL for their work.

5

What is the difference between the CDDL and the GNU GPL and is it compatible with them?

The GNU GPL requires that you connect it to any program that is a derivative work of the original software. That means that you have to make its source code available. This is considered strong reciprocity. The CDDL takes a softer approach. As we have seen, if your additions are made in independent files that do not contain any part of the original program — then these files are not subject to the CDDL. It means, amongst other things, that you do not have to release these files' source code.

Furthermore, the GPL takes a tough stand on changing the license's terms and conditions. While certain additions are permitted under GPL 3, the general rule is that no other changes can be introduced. The CDDL only subjects the source code version of the software to its provisions. The executable version can be distributed under the terms of any other license that you choose, provided that it is in compliance with the terms of the CDDL and that the license for the executable does not attempt to limit or alter the recipients' rights in the source code form of the program.

Due to these differences, the CDDL is not considered compatible with the GNU GPL.

eclipse Berkeley Software Distribution (BSD)

The Eclipse Public License (EPL) is an open source license developed by the Eclipse Foundation. It's derived from the Common Public License (CPL). The Eclipse codebase is now available under the EPL was formerly licensed under the CPL.

Here are some questions and answers regarding the EPL:

1 **What are the terms and conditions of the Eclipse Public License?**

The EPL license is a copyleft license. If you modify an EPL'ed component and distribute it in the source code form as part of your program, you're required to disclose the modified code under the EPL. If you distribute such a program in its object code form, you're required to state that the source code can be made available to the recipient upon request. You're also required to share the method for requesting the source code.

The Eclipse Foundation makes it clear that, in their opinion, "merely interfacing or interoperating" with an Eclipse plugin does not make your code a derivative work of the plugin.

If you redistribute a program with an EPL component, you are obligated to include the full license text and the copyrights.

The EPL protects the author from possible lawsuits or damages caused if a company used their component in a commercial product. It also offers a patent grant.

2 **Is the EPL considered a copyleft license?**

Yes, the EPL is considered a weak copyleft license.

Weak copyleft licenses require you to disclose your source on the source code, but not on binaries and therefore you can compile covered sources with others and distribute the resulting (merged) binaries under the license of your choice. With strong copyleft licenses, like the GPL family, you are obligated to reuse the same license in case of re-distribution of copies or derivatives on both source and binaries.

3 **What is the difference between the Eclipse Public License and IBM's Common Public License (CPL)?**

The EPL revises the CPL by deleting the first sentence in the 7th section of the original CPL that was believed to be overly broad and non-conducive to the growth of the Eclipse ecosystem. The removed content explained how the CPL handled patent retaliation.

4 **What is the difference between the Eclipse Public License and the GNU GPL?**

The GNU GPL family of licenses has a strong copyleft clause requiring users to release their software's full source code irrespective of the percentage of the GPL'ed code included. The EPL, on the other hand, doesn't require you to open source your entire code. You're only required to open source any included modified EPL'ed components when distributing in the source code form and make the source code available upon request when distributing in the object form.

5 ## Is the Eclipse Public License compatible with the GNU GPL?

The EPL is not compatible with the GNU GPL. The GPL requires the user to release the entire software and that the distributor not "impose any further restrictions on the recipients' exercise of the rights granted". The EPL, however, requires that anyone distributing the work grant every recipient a license to any patents that they might hold that cover the modifications they have made.

Because this is a "further restriction" on the recipients, the distribution of such a combined work does not satisfy the GPL.

*The author of this white paper is not a lawyer, and you should not interpret this as legal advice of any kind. Information is provided on an as-is basis. For a legal consultation, please contact your legal advisor.