What is *Free Software?*

Free Software Foundation

Updated 9 June 2026 · gnu.org/philosophy/free-sw.html

GNU

"Free software" means software that respects users' freedom and community. Roughly, it means that **the users have the freedom to run, copy, distribute, study, change and improve the software**. Thus, "free software" is a matter of liberty, not price. To understand the concept, you should think of "free" as in "free speech," not as in "free beer." We sometimes call it "libre software," borrowing the French or Spanish word for "free" as in freedom, to show we do not mean the software is gratis.

You may have paid money to get copies of a free program, or you may have obtained copies at no charge. But regardless of how you got your copies, you always have the freedom to copy and change the software, even to sell copies.

We campaign for these freedoms because everyone deserves them. With these freedoms, the users (both individually and collectively) control the program and what it does for them. When users don't control the program, we call it a "nonfree" or "proprietary" program. The nonfree program controls the users, and the developer controls the program; this makes the program an instrument of unjust power.

"Open source" is something different: it has a very different philosophy based on different values. Its practical definition is different too, but nearly all open source programs are in fact free. We explain the difference in Why "Open Source" misses the point of Free Software.

Have a question about free software licensing not answered here? See our other licensing resources, and if necessary contact the FSF Compliance Lab at licensing@fsf.org.

The Free Software Definition

The free software definition presents the criteria for whether a particular software program qualifies as free software. From time to time we revise this definition, to clarify it or to resolve questions about subtle issues. See the History section below for a list of changes that affect the definition of free software.

The four essential freedoms

A program is free software if the program's users (referred to individually here as "you") have the four essential freedoms:[1]

- The freedom to run the program as you wish, for any purpose (freedom 0).
- The freedom to study how the program works, and change it so it does your computing as you wish (freedom 1). Access to the source code is a precondition for this.
- The freedom to redistribute copies so you can help others (freedom 2).
- The freedom to distribute copies of your modified versions to others (freedom 3). By doing this you can give the whole community a chance to benefit from your changes. Access to the source code is a precondition for this.

A program is free software if it gives users adequately all of these freedoms. Otherwise, it is nonfree. While we can distinguish various nonfree distribution schemes in terms of how far they fall short of being free, we consider them all equally unethical.

In any given scenario, these freedoms must apply to whatever code we plan to make use of, or lead others to make use of. For instance, consider a program A which automatically launches a program B to handle some cases. If we plan to distribute A as it stands, that implies users will need B, so we need to judge whether both A and B are free. However, if we plan to modify A so that it doesn't use B, only A needs to be free; B is not pertinent to that plan.

Free software *can* be commercial

"Free software" does not mean "noncommercial." On the contrary, a free program must be available for commercial use, commercial development, and commercial distribution. This policy is of fundamental importance—without this, free software could not achieve its aims.

We want to invite everyone to use the GNU system, including businesses and their workers. That requires allowing commercial use. We hope that free replacement programs will supplant comparable proprietary programs, but they can't do that if businesses are forbidden to use them. We want commercial products that contain software to include the GNU system, and that would constitute commercial distribution for a price. Commercial development of free software is no longer unusual; such free commercial software is very important. Paid, professional support for free software fills an important need.

Thus, to exclude commercial use, commercial development or commercial distribution would hobble the free software community and obstruct its path to success. We must conclude that a program licensed with such restrictions does not qualify as free software.

A free program must offer the four freedoms to any would-be user that obtains a copy of the software, who has complied thus far with the conditions of the free license covering the software in any previous distribution of it. Putting some of the freedoms off limits to some users, or requiring that users pay, in money or in kind, to exercise them, is tantamount to not granting the freedoms in question, and thus renders the program nonfree.

Clarifying the Boundary Between Free and Nonfree

In the rest of this article we explain more precisely how far the various freedoms need to extend, on various issues, in order for a program to be free.

The freedom to run the program as you wish

The freedom to run the program means the freedom for any kind of person or organization to use it on any kind of computer system, for any kind of overall job and purpose, without being required to communicate about it with the developer or any other specific entity. In this freedom, it is the *user's* purpose that matters, not the *developer's* purpose; you as a user are free to run the program for your purposes, and if you distribute it to other people, they are then free to run it for their purposes, but you are not entitled to impose your purposes on them.

The freedom to run the program as you wish means that you are not forbidden or stopped from making it run. This has nothing to do with what functionality the program has, whether it is technically capable of functioning in any given environment, or whether it is useful for any particular computing activity.

For example, if the code arbitrarily rejects certain meaningful inputs —or even fails unconditionally—that may make the program less useful, perhaps even totally useless, but it does not deny users the freedom to run the program, so it does not conflict with freedom 0. If the program is free, the users can overcome the loss of usefulness, because freedoms 1 and 3 permit users and communities to make and distribute modified versions without the arbitrary nuisance code.

"As you wish" includes, optionally, "not at all" if that is what you wish. So there is no need for a separate "freedom not to run a program."

Permission to run the program must not be conditional on what you do with other separate programs, not even if they are in use on the same computer or for the same general activity. "Separate programs" means that they have not been combined in a way that would make them count as a single larger work.

The freedom to study the source code and make changes

In order for freedoms 1 and 3 (the freedom to make changes and the freedom to publish the changed versions) to be meaningful, you need to have access to the source code of the program. Therefore, accessibility of source code is a necessary condition for free software. Obfuscated "source code" is not real source code and does not count as source code.

Source code is defined as the preferred form of the program for making changes in. Thus, whatever form a developer changes to develop the program is the source code of that developer's version. Placing a free license on a binary program does not make it source code—unless that binary program's developer always maintains it by patching that binary and has no real source code for it.

Freedom 1 includes the freedom to use your changed version in place of the original. If the program is delivered in a product designed to run someone else's modified versions but refuse to run yours—a practice known as "tivoization" or "lockdown," or (in its practitioners' perverse terminology) as "secure boot"—freedom 1 becomes an empty pretense rather than a practical reality. These binaries are not free software even if the source code they are compiled from is free.

One important way to modify a program is by merging in available free subroutines and modules. If the program's license says that you cannot merge in a suitably licensed existing module—for instance, if it requires you to be the copyright holder of any code you add—then the license is too restrictive to qualify as free.

Whether a change constitutes an improvement is a subjective matter. If your right to modify a program is limited, in substance, to changes that someone else considers an improvement, that program is not free.

One special case of freedom 1 is to delete the program's code so it returns after doing nothing, or make it invoke some other program. Thus, freedom 1 includes the "freedom to delete the program."

The freedom to redistribute if you wish: basic requirements

Freedom to distribute (freedoms 2 and 3) means you are free to redistribute copies, either with or without modifications, either gratis or charging a fee for distribution, to anyone anywhere. Being free to do these things means (among other things) that you do not have to ask or pay for permission to do so.

You should also have the freedom to make modifications and use them privately in your own work or play, without even mentioning that they exist. If you do publish your changes, you should not be required to notify anyone in particular, or in any particular way.

Freedom 3 includes the freedom to release your modified versions as free software. A free license may also permit other ways of releasing them; in other words, it does not have to be a copyleft license. However, a license that requires modified versions to be nonfree does not qualify as a free license.

The freedom to redistribute copies must include binary or executable forms of the program, as well as source code, for both modified and unmodified versions. (Distributing programs in runnable form is necessary for conveniently installable free operating systems.) It is OK if there is no way to produce a binary or executable form for a certain program (since some languages don't support that feature), but you must have the freedom to redistribute such forms should you find or develop a way to make them.

Copyleft

Certain kinds of rules about the manner of distributing free software are acceptable, when they don't conflict with the central freedoms. For example, copyleft (very simply stated) is the rule that when redistributing the program, you cannot add restrictions to deny other people the central freedoms. This rule does not conflict with the central freedoms; rather it protects them.

In the GNU project, we use copyleft to protect the four freedoms legally for everyone. We believe there are important reasons why it is better to use copyleft. However, noncopylefted free software is ethical too. See Categories of Free Software for a description of how "free software," "copylefted software" and other categories of software relate to each other.

Rules about packaging and distribution details

Rules about how to package a modified version are acceptable, if they don't substantively limit your freedom to release modified versions, or your freedom to make and use modified versions privately. Thus, it is acceptable for the license to require that you change the name of the modified version, remove a logo, or identify your modifications as yours. As long as these requirements are not so burdensome that they effectively hamper you from releasing your changes, they are acceptable; you're already making other changes to the program, so you won't have trouble making a few more.

Rules that "if you make your version available in this way, you must make it available in that way also" can be acceptable too, on the same condition. An example of such an acceptable rule is one saying that if you have distributed a modified version and a previous developer asks for a copy of it, you must send one. (Note that such a rule still leaves you the choice of whether to distribute your version at all.) Rules that require release of source code to the users for versions that you put into public use are also acceptable.

A special issue arises when a license requires changing the name by which the program will be invoked from other programs. That effectively hampers you from releasing your changed version so that it can replace the original when invoked by those other programs. This sort of requirement is acceptable only if there's a suitable aliasing facility that allows you to specify the original program's name as an alias for the modified version.

Export regulations

Sometimes government export control regulations and trade sanctions can constrain your freedom to distribute copies of programs internationally. Software developers do not have the power to eliminate or override these restrictions, but what they can and must do is refuse to impose them as conditions of use of the program. In this way, the restrictions will not affect activities and people outside the jurisdictions of these governments. Thus, free software licenses must not require obedience to any nontrivial export regulations as a condition of exercising any of the essential freedoms.

Merely mentioning the existence of export regulations, without making them a condition of the license itself, is acceptable since it does not restrict users. If an export regulation is actually trivial for free software, then requiring it as a condition is not an actual problem; however, it is a potential problem, since a later change in export law could make the requirement nontrivial and thus render the software nonfree.

Legal considerations

In order for these freedoms to be real, they must be permanent and irrevocable as long as you do nothing wrong; if the developer of the software has the power to revoke the license, or retroactively add restrictions to its terms, without your doing anything wrong to give cause, the software is not free.

A free license may not require compliance with the license of a nonfree program. Thus, for instance, if a license requires you to comply with the licenses of "all the programs you use," in the case of a user that runs nonfree programs this would require compliance with the licenses of those nonfree programs; that makes the license nonfree.

It is acceptable for a free license to specify which jurisdiction's law applies, or where litigation must be done, or both.

Contract-based licenses

Most free software licenses are based on copyright, and there are limits on what kinds of requirements can be imposed through copyright. If a copyright-based license respects freedom in the ways described above, it is unlikely to have some other sort of problem that we never anticipated (though this does happen occasionally). However, some free software licenses are based on contracts, and contracts can impose a much larger range of possible restrictions. That means there are many possible ways such a license could be unacceptably restrictive and nonfree.

We can't possibly list all the ways that might happen. If a contract-based license restricts the user in an unusual way that copyright-based licenses cannot, and which isn't mentioned here as legitimate, we will have to think about it, and we will probably conclude it is nonfree.

The Free Software Definition in Practice

How we interpret these criteria

Note that criteria such as those stated in this free software definition require careful thought for their interpretation. To decide whether a specific software license qualifies as a free software license, we judge it based on these criteria to determine whether it fits their spirit as well as the precise words. If a license includes unconscionable restrictions, we reject it, even if we did not anticipate the issue in these criteria. Sometimes a license requirement raises an issue that calls for extensive thought, including discussions with a lawyer, before we can decide if the requirement is acceptable. When we reach a conclusion about a new issue, we often update these criteria to make it easier to see why certain licenses do or don't qualify.

Get help with free licenses

If you are interested in whether a specific license qualifies as a free software license, see our list of licenses. If the license you are concerned with is not listed there, you can ask us about it by sending us email at <licensing@gnu.org>.

If you are contemplating writing a new license, please contact the Free Software Foundation first by writing to that address. The proliferation of different free software licenses means increased work for users in understanding the licenses; we may be able to help you find an existing free software license that meets your needs.

If that isn't possible, if you really need a new license, with our help you can ensure that the license really is a free software license and avoid various practical problems.

Use the right words when talking about free software

When talking about free software, it is best to avoid using terms like "give away" or "for free," because those terms imply that the issue is about price, not freedom. Some common terms such as "piracy" embody opinions we hope you won't endorse. See Confusing Words and Phrases that are Worth Avoiding for a discussion of these terms. We also have a list of proper translations of "free software" into various languages.

Another group uses the term "open source" to mean something close (but not identical) to "free software." We prefer the term "free software" because, once you have heard that it refers to freedom rather than price, it calls to mind freedom. The word "open" never refers to freedom.

Beyond Software

Software manuals must be free, for the same reasons that software must be free, and because the manuals are in effect part of the software.

The same arguments also make sense for other kinds of works of practical use—that is to say, works that embody useful knowledge, such as educational works and reference works. Wikipedia is the best-known example.

Any kind of work *can* be free, and the definition of free software has been extended to a definition of free cultural works applicable to any kind of works.

History

From time to time we revise this Free Software Definition. Here is the list of substantive changes, along with links to show exactly what was changed.

Version 1.169
Explain more clearly why the four freedoms must apply to commercial activity. Explain why the four freedoms imply the freedom not to run the program and the freedom to delete it, so there is no need to state those as separate requirements.

Version 1.165
Clarify that arbitrary annoyances in the code do not negate freedom 0, and that freedoms 1 and 3 enable users to remove them.

Version 1.153
Clarify that freedom to run the program means nothing stops you from making it run.

Version 1.141
Clarify which code needs to be free.

Version 1.135
Say each time that freedom 0 is the freedom to run the program as you wish.

Version 1.134
Freedom 0 is not a matter of the program's functionality.

Version 1.131
A free license may not require compliance with a nonfree license of another program.

Version 1.129
State explicitly that choice of law and choice of forum specifications are allowed. (This was always our policy.)

Version 1.122
An export control requirement is a real problem if the requirement is nontrivial; otherwise it is only a potential problem.

Version 1.118
Clarification: the issue is limits on your right to modify, not on what modifications you have made. And modifications are not limited to "improvements"

Version 1.111
Clarify 1.77 by saying that only retroactive *restrictions* are unacceptable. The copyright holders can always grant additional *permission* for use of the work by releasing the work in another way in parallel.

Version 1.105
Reflect, in the brief statement of freedom 1, the point (already stated in version 1.80) that it includes really using your modified version for your computing.

Version 1.92
Clarify that obfuscated code does not qualify as source code.

Version 1.90
Clarify that freedom 3 means the right to distribute copies of your own modified or improved version, not a right to participate in someone else's development project.

Version 1.89
Freedom 3 includes the right to release modified versions as free software.

Version 1.80
Freedom 1 must be practical, not just theoretical; i.e., no tivoization.

Version 1.77
Clarify that all retroactive changes to the license are unacceptable, even if it's not described as a complete replacement.

Version 1.74
Four clarifications of points not explicit enough, or stated in some places but not reflected everywhere:

- "Improvements" does not mean the license can substantively limit what kinds of modified versions you can release. Freedom 3 includes distributing modified versions, not just changes.
- The right to merge in existing modules refers to those that are suitably licensed.
- Explicitly state the conclusion of the point about export controls.
- Imposing a license change constitutes revoking the old license.

Version 1.57
Add "Beyond Software" section.

Version 1.46
Clarify whose purpose is significant in the freedom to run the program for any purpose.

Version 1.41
Clarify wording about contract-based licenses.

Version 1.40
Explain that a free license must allow to you use other available free software to create your modifications.

Version 1.39
Note that it is acceptable for a license to require you to provide source for versions of the software you put into public use.

Version 1.31
Note that it is acceptable for a license to require you to identify yourself as the author of modifications. Other minor clarifications throughout the text.

Version 1.23
Address potential problems related to contract-based licenses.

Version 1.16
Explain why distribution of binaries is important.

Version 1.11
Note that a free license may require you to send a copy of versions you distribute to previous developers on request.

There are gaps in the version numbers shown above because there are other changes in this page that do not affect the definition or its interpretations. For instance, the list does not include changes in asides, formatting, spelling, punctuation, or other parts of the page. You can review the complete list of changes to the page through the cvsweb interface.

Footnote

1 The reason they are numbered 0, 1, 2 and 3 is historical. Around 1990 there were three freedoms, numbered 1, 2 and 3. Then we realized that the freedom to run the program needed to be mentioned explicitly. It was clearly more basic than the other three, so it properly should precede them. Rather than renumber the others, we made it freedom 0.

Copyright © 1996-2002, 2004-2019, 2021, 2022, 2023, 2024, 2026 Free Software Foundation, Inc.

This page is licensed under a Creative Commons Attribution-NoDerivatives 4.0 International License. creativecommons.org/licenses/by-nd/4.0/