A Quick Guide to GPLv3

Brett Smith
Free Software Foundation, Inc.
licensing@fsf.org

Introduction

After a year and a half of public consultation, thousands of comments, and four drafts, version 3 of the GNU General Public License (GPLv3) was finally published on June 29. While there's been a lot of discussion about the license since the first draft appeared, not many people have talked about the benefits that it provides developers. We've published this guide to fill that gap. We'll start with a brief refresher on free software, copyleft, and the goals of the GPL. We'll then review the major changes in the license to see how they advance those goals and benefit developers.

The Foundations of the GPL

Nobody should be restricted by the software they use. There are four freedoms that every user should have:

- the freedom to use the software for any purpose,
- the freedom to share the software with your friends and neighbors,
- the freedom to change the software to suit your needs, and
- the freedom to share the changes you make.

When a program offers users all of these freedoms, we call it free software.

Developers who write software can release it under the terms of the GNU GPL. When they do, it will be free software and stay free software, no matter who changes or distributes the program. We call this copyleft: the software is copyrighted, but instead of using those rights to restrict users like proprietary software does, we use them to ensure that every user has freedom.

We update the GPL to protect its copyleft from being undermined by legal or technological developments. The most recent version protects users from three recent threats:

- Tivoization: Some companies have created various different kinds of devices that run GPLed software, and then rigged the hardware so that they can change the software that's running, but you cannot. If a device can run arbitrary software, it's a general-purpose computer, and its owner should control what it does. When a device thwarts you from doing that, we call that tivoization.
- Laws prohibiting free software: Legislation like the Digital Millennium Copyright Act and the European Union Copyright Directive make it a crime to write or share software that can break DRM. These laws should not interfere with the rights the GPL grants you.
- Discriminatory patent deals: Microsoft has recently started telling people that they will not sue free software users for patent infringement—as long as you get the software from a vendor that's paying Microsoft for the privilege. Ultimately, Microsoft is trying to collect royalties for the use of free software, which interferes with users' freedom. No company should be able to do this.

Version 3 also has a number of improvements to make the license easier for everyone to use and understand. But even with all these changes, GPLv3 isn't a radical new license; instead it's an evolution of the previous version. Though a lot of text has changed, much of it simply clarifies what GPLv2 said. With that in mind, let's review the major changes in GPLv3, and talk about how they improve the license for users and developers.

Neutralizing Laws That Prohibit Free Software – But Not Forbidding DRM

You're probably familiar with the Digital Restrictions Management (DRM) on DVDs and other media. You're probably also familiar with the laws that make it illegal to write your own tools to bypass those restrictions, like the Digital Millennium Copyright Act and the European Union Copyright Directive. Nobody should be able to stop you from writing any code that you want, and GPLv3 protects this right for you.

It's always possible to use GPLed code to write software that implements DRM. However, if someone does that with code protected by GPLv3, section 3 says that the system will not count as an effective technological "protection" measure. This means that if you break the DRM, you'll be free to distribute your own software that does that, and you won't be threatened by the DMCA or similar laws.

As usual, the GNU GPL does not restrict what people do in software; it just stops them from restricting others.

Protecting Your Right to Tinker

Tivoization is a dangerous attempt to curtail users' freedom: the right to modify your software will become meaningless if none of your computers let you do it. GPLv3 stops tivoization by requiring the distributor to provide you with whatever information or data is necessary to install modified software on the device. This may be as simple as a set of instructions, or it may include special data such as cryptographic keys or information about how to bypass an integrity check in the hardware. It will depend on how the hardware was designed—but no matter what information you need, you must be able to get it.

This requirement is limited in scope. Distributors are still allowed to use cryptographic keys for any purpose, and they'll only be required to disclose a key if you need it to modify GPLed software on the device they gave you. The GNU Project itself uses GnuPG to prove the integrity of all the software on its FTP site, and measures like that are beneficial to users. GPLv3 does not stop people from using cryptography; we wouldn't want it to. It only stops people from taking away the rights that the license provides you—whether through patent law, technology, or any other means.

Stronger Protection Against Patent Threats

In the 17 years since GPLv2 was published, the software patent landscape has changed considerably, and free software licenses have developed new strategies to address them. GPLv3 reflects these changes too. Whenever someone conveys software covered by GPLv3 that they've written or modified, they must provide every recipient with any patent licenses necessary to exercise the rights that the GPL gives them. In addition to that, if any licensee tries to use a patent suit to stop another user from exercising those rights, their license will be terminated.

What this means for users and developers is that they'll be able to work with GPLv3-covered software without worrying that a desperate contributor will try to sue them for patent infringement later. With these changes, GPLv3 affords its users more defenses against patent aggression than any other free software license.

Clarifying License Compatibility

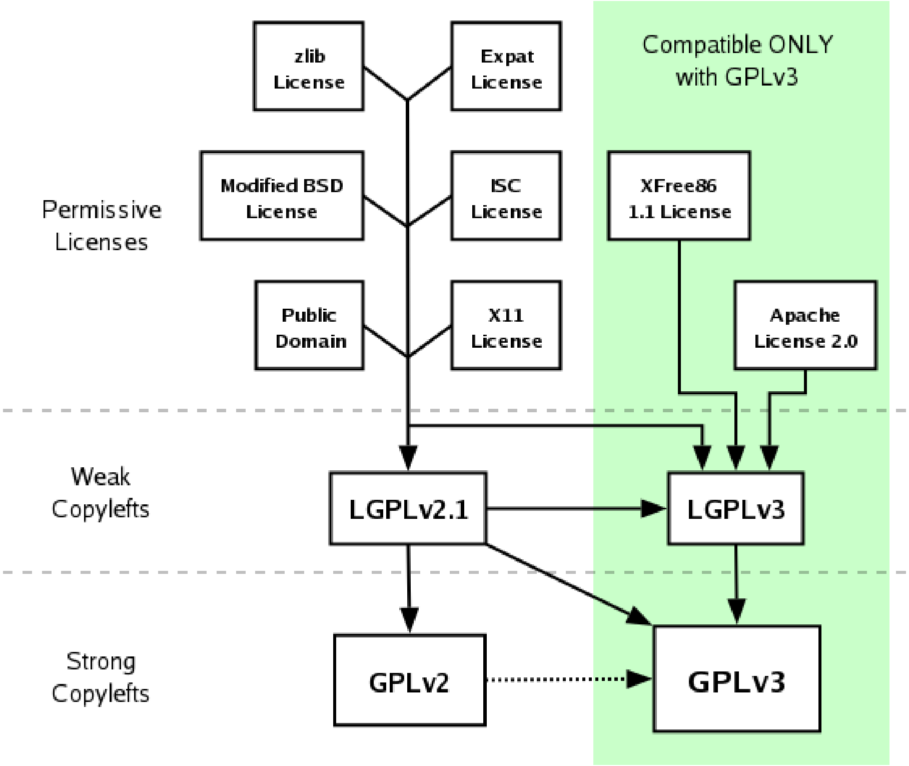
If you found some code and wanted to incorporate it into a GPLed project, GPLv2 said that the license on the other code was not allowed to have any restrictions that were not already in GPLv2. As long as that was the case, we said the license was GPL-compatible.

However, some licenses had requirements that weren't really restrictive, because they were so easy to comply with. For example, some licenses say that they don't give you permission to use certain trademarks. That's not really an additional restriction: if that clause wasn't there, you still wouldn't have permission to use the trademark. We always said those licenses were compatible with GPLv2, too.

Now, GPLv3 explicitly gives everyone permission to use code that has requirements like this. These new terms should help clear up misunderstandings about which licenses are GPL-compatible, why that is, and what you can do with GPL-compatible code.

New Compatible Licenses

In addition to clarifying the rules about licenses that are already GPL-compatible, GPLv3 is also newly compatible with a few other licenses. The Apache License 2.0 is a prime example. Lots of great free software is available under this license, with strong communities surrounding it. We hope that this change in GPLv3 will foster more cooperation and sharing within the free software community. The chart below helps illustrate some common compatibility relationships between different free software licenses:



Arrows pointing from one license to another indicate that the first license is compatible with the second. This is true even if you follow multiple arrows to get from one license to the other; so, for example, the ISC license is compatible with GPLv3. GPLv2 is compatible with GPLv3 if the program allows you to choose "any later version" of the GPL, which is the case for most software released under this license. This diagram is not comprehensive (see our licenses page for a more complete list of licenses compatible with GPLv2 and GPLv3), but plainly illustrates that GPLv3 is compatible with just about everything GPLv2 is, and then some.

The GNU Affero GPL version 3 has also been brought into the fold. The original Affero GPL was designed to ensure that all users of a web application would be able to receive its source. The GNU Affero GPL version 3 broadens this goal: it is applicable to all network-interactive software, so it will also work well for programs like game servers. The additional provision is also more flexible, so that if someone uses AGPLed source in an application without a network interface, they'll only have to provide source in the same sort of way the GPL has always required. By making these two licenses compatible, developers of network-interactive software will be able to strengthen their copyleft while still building on top of the mature body of GPLed code available to them.

More Ways for Developers to Provide Source

One of the fundamental requirements of the GPL is that when you distribute object code to users, you must also provide them with a way to get the source. GPLv2 gave you a few ways to do this, and GPLv3 keeps those intact with some clarification. It also offers you new ways to provide source when you convey object code over a network. For instance, when you host object code on a web or FTP server, you can simply provide instructions that tell visitors how to get the source from a third-party server. Thanks to this new option, fulfilling this requirement should be easier for many small distributors who only make a few changes to large bodies of source.

The new license also makes it much easier to convey object code via BitTorrent. First, people who are merely downloading or seeding the torrent are exempt from the license's requirements for conveying the software. Then, whoever starts the torrent can provide source by simply telling other torrent users where it is available on a public network server.

These new options help keep the GPL in line with community standards for offering source, without making it harder for users to get.

Less Source to Distribute: New System Libraries Exception

Both versions of the GPL require you to provide all the source necessary to build the software, including supporting libraries, compilation scripts, and so on. They also draw the line at System Libraries: you're not required to provide the source for certain core components of the operating system, such as the C library.

GPLv3 has adjusted the definition of System Library to include software that may not come directly with the operating system, but that all users of the software can reasonably be expected to have. For example, it now also includes the standard libraries of common programming languages such as Python and Ruby.

The new definition also makes it clear that you can combine GPLed software with GPL-incompatible System Libraries, such as OpenSolaris' C library, and distribute them both together. These changes will make life easier for free software distributors who want to provide these combinations to their users.

A Global License

GPLv2 talks about "distribution" a lot—when you share the program with someone else, you're distributing it. The license never says what distribution is, because the term was borrowed from United States copyright law. We expected that judges would look there for the definition. However, we later found out that copyright laws in other countries use the same word, but give it different meanings. Because of this, a judge in such a country might analyze GPLv2 differently than a judge in the United States.

GPLv3 uses a new term, "convey," and provides a definition for that term. "Convey" has the same meaning we intended for "distribute," but now that this is explained directly in the license, it should be easy for people everywhere to understand what we meant. There are other minor changes throughout the license that will also help ensure it is applied consistently worldwide.

When the Rules Are Broken: A Smooth Path to Compliance

Under GPLv2, if you violated the license in any way, your rights were automatically and permanently lost. The only way to get them back was to petition the copyright holder. While a strong defense against violations is valuable, this policy could cause a lot of headache when someone accidentally ran afoul of the rules. Asking all the copyright holders for a formal restoration of the license could be burdensome and costly: a typical GNU/Linux distribution draws upon the work of thousands.

GPLv3 offers a reprieve for good behavior: if you violate the license, you'll get your rights back once you stop the violation, unless a copyright holder contacts you within 60 days. After you receive such a notice, you can have your rights fully restored if you're a first-time violator and correct the violation within 30 days. Otherwise, you can work out the issue on a case-by-case basis with the copyright holders who contacted you, and your rights will be restored afterward.

Compliance with the GPL has always been the top priority of the FSF Compliance Lab and other groups enforcing the license worldwide. These changes ensure that compliance remains the top priority for enforcers, and gives violators incentive to comply.

The Latest and Greatest

Some of these changes probably seem less important to you than others. That's okay. Every project is different, and needs different things from its license. But odds are that a number of these improvements will help you and your work.

And taken as a whole, all these upgrades represent something more: we made a better copyleft. It does more to protect users' freedom, but it also enables more cooperation in the free software community. But updating the license is only part of the job: in order for people to get the benefits it offers, developers need to use GPLv3 for their projects, too. By releasing your own software under the new license, everyone who deals with it—users, other developers, distributors, even lawyers—will benefit. We hope you'll use GPLv3 for your next release.

If you'd like to learn more about upgrading your project to GPLv3, the FSF Compliance Lab would be happy to assist you. On our web site, you can find basic instructions for using the license, and an FAQ addressing common concerns that people have about it. If your situation is more complicated than that, please contact us and we'll do what we can to help you with your transition. Together, we can help protect freedom for all users.