Using GPL code in your software: the essentials

Technology Law | May 2016

A huge amount of useful software is freely available under the **GPL open source licence**. This article gives an overview of the GPL and important issues for developers who may wish to incorporate GPL-licenced code in their own programs.

1. GPL (the GNU General Public License)

The GPL is a "copyleft" software licence[1] published by the US-based Free Software Foundation.[2] This means that where a program or other work licensed under the GPL is *incorporated* (see section 4 below) in a second work, then that second work (if distributed to others) must also be licensed on the terms of the GPL (or a compatible license).

In general terms, if you use GPL-licensed code in your program that you wish to distribute to others, then you must also license your program under the GPL.

For example, if a developer writes and distributes Program A that incorporates a third party's Library B which is licensed under the GPL, Program A must also be licensed under the GPL. It should be noted that Program A does not automatically become licensed under the GPL just because it incorporates Library B, but it would breach the terms of Library B's licence if Program A was **distributed** under a non-GPL licence.

This has implications for proprietary and/or commercial software developers, as many popular and important libraries are licensed under the GPL. If a developer wishes to incorporate a GPL-licensed library into their proprietary program distributed to customers, they must either:

* Be prepared to license their own program under the GPL (which may not be a desired or viable option for commercial proprietary software); or

* Acquire a licence for the library that permits its inclusion in the proprietary program (usually described as a "commercial licence"), if such a commercial licence is available from the relevant licensor(s); or

* Adopt an approach where the GPL-licensed library is not actually incorporated into the proprietary program, but is provided as a separate independent utility that can be configured to work at "arms-length" with the proprietary program.

2. LGPL (the Lesser GNU General Public License)

The **LGPL** (the *Lesser* GPL) is a modified version of the GPL that has an exemption allowing a library to be *dynamically linked* (see section 4) with a program without requiring the program itself to be licensed under the GPL or LPGL. This allows proprietary developers to utilise an LGPL-licensed library in a program without affecting its proprietary licensing.

3. Interpreting the GPL

While the GPL is widely used around the world, interpreting it is somewhat of a challenge for a number of reasons:

---

[1] See https://en.wikipedia.org/wiki/Copyleft

[2] The official GPL website is at http://www.gnu.org/licenses/licenses.html. This article refers to the latest version of the GPL (version 3) which is available at http://www.gnu.org/licenses/gpl-3.0.html.

- It is not written for any specific scenario (as the name says, it is a "general" licence) and as a result may not be as clear (in legal terms) as a situation-specific license, and may not be consistent with other representations of the licensor. Some key terms are undefined (or only vaguely defined) and there is some debate amongst the open source community and commentators over the interpretation of various aspects of the licence;

- Furthermore, as the GPL is a template licence, it can be difficult to ascertain the intention of the parties to resolve any ambiguity or uncertainty. The FSF itself has published a variety of documents over the years describing and explaining the licences, and making assertions about how certain provisions should or would be interpreted (presumably under US law), but the legal status and accuracy of these extra-contractual documents in relation to any particular use of a licence in a given jurisdiction is uncertain;

- There are no governing law or jurisdiction provisions in the licences (the FSF says this is deliberate to avoid any particular US state from imposing additional restrictions) so it can be unclear which jurisdiction's laws (including laws of interpretation) would apply; and

- There have been relatively few Court cases involving GPL licences worldwide, and therefore little useful precedent for guidance. There have been no Court decisions involving the GPL in New Zealand, Australia or the UK.

4. Incorporating GPL-licensed code in proprietary programs

A key question that arises when using GPL licensed software with proprietary programs is: At what point does a GPL-licensed library become 'incorporated' into my proprietary program so as to require my program to be distributed under the GPL? The actual term used in the GPL is "work based on the Program", but this leaves some grey areas.

The answer is straightforward where the library's source code (or a substantial part of it) has been directly copied into a second program's source code. In such cases, the GPL code is clearly incorporated into the second program, and the second program must be distributed under the GPL to comply with its use of the GPL-licensed source code.

A library may also be incorporated into a second program by way of *linking*[3] a compiled version of it to the second program. Linking can be static or dynamic:

- Static linking involves compiling the library into the program itself, essentially hardwiring it into the main program;

- Dynamic linking involves the main program loading the library (often contained in a separate file, possibly shared by multiple programs) when the main program is run.

The kind of linking used can have different consequences depending on whether library is licensed under the GPL or the **LGPL** (see section 2):[4]

- Under the GPL, either static or dynamic linking requires the main program to be distributed under the GPL, with the result that linking a GPL-licensed library is incompatible (in licensing terms) with a proprietary program.

---

[3] See https://en.wikipedia.org/wiki/Linker_%28computing%29

[4] There is another variant of the GPL, beyond the scope of this article, known as the Affero General Public Licence ("AGPL") that extends coverage to using software over a network – see https://en.wikipedia.org/wiki/Affero_General_Public_License.

- Under the LGPL, only static linking requires the main program to be distributed under the LGPL. Dynamic linking to LGPL-licensed software does **not** require the main program to be distributed under the LGPL (this distinction is the purpose of the LGPL licence). This means that proprietary programs can use dynamically-linked LGPL-licensed libraries while retaining their proprietary licensing.

Proprietary programs may also be able to use GPL-licensed software without licensing issues if the proprietary program and the GPL-licensed software are, in substance, separate programs that operate at 'arms-length'.

5. Potential GPL violations & consequences

As noted, inclusion of a GPL-licensed library in a proprietary program does not automatically render the proprietary program subject to the GPL, but it would be a breach of the library's GPL licence to **distribute** the proprietary program under a non-GPL licence.[5]

Clause 4 of the GPL states that such a breach automatically terminates the licence. That is, the proprietary developer's licence to use the GPL work in their program is automatically terminated if they distribute their program under a non-GPL licence. However, clause 4 adds the following protection for end-users of the proprietary program:

> However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

Together with clause 6 of the GPL (stating that every recipient of GPL-licensed software automatically receives a direct licence from the original licensor), this means that end-users of a proprietary program that improperly includes a GPL-licensed library are not affected (in terms of their right to use the program) by the automatic termination of the proprietary developer's GPL licence.

However, possible legal consequences for the proprietary developer who has breached the GPL may include:

- Liability to the licensor of the GPL-licensed work for damages (although it would be difficult for the copyright holder to prove any real "loss" if the software is freely licensed with no commercial option available);

- An "account of profits", which seeks to strip the infringer of profits arising from the unlawful use;

- An injunction preventing the infringer from further distributing the software in breach of the GPL, which in turn could result in lost revenue or liability under contracts;[6] or

- A legal declaration that the GPL has been breached.

Such consequences have proven rare, at least in part because it is expensive to pursue legal remedies and there is little financial incentive. Many GPL projects are non-commercial enterprises comprised of volunteers across multiple countries. However, organisations such as the FSF and other free software groups have taken on some cases as a matter of principle and to defend the integrity of open source licensing.[7]

---

[5] It is also unlikely, in our opinion, that an end-user could compel the proprietary developer to release their program under the GPL, although we repeat our observation that the GPL has never been tested in a New Zealand Court, nor to our knowledge in an Australian or UK Court.

[6] This occurred in one of the few court cases involving the GPL to date, in which a German court issued an injunction preventing a German company from distributing a product incorporating GPL code until it complied with the GPL: http://www.theregister.co.uk/2004/04/21/licence_germany/

[7] A German case brought by a free software organisation is currently in progress against multinational US company VMWare for alleged violation of the GPL: http://www.infoworld.com/article/2893695/open-source-software/vmware-heading-to-court-over-gpl-violations.html

Further (more likely) consequences may include:

- Problems arising from IP audits or due diligence projects, which could have significant implications for a proposed business/asset sale, valuation, joint-venture or merger;

- Breaches of IP warranties or other contractual obligations to end-users;

- Potential conflicts with end-user procurement policies (i.e. policies stating that all suppliers must fully comply with licensing requirements); and

- Negative publicity and reputational damage (there have been intermittent examples of "naming and shaming" GPL violators, as well as the cases taken by free software groups).[8]

Remedying a GPL licensing issue could be achieved by replacing the relevant code with a suitable non-copyleft open source alternative (for example, code licensed under the Apache, MIT or BSD licences)[9] or acquiring a commercial licence (if one is available). In some cases this may be straightforward, but in other cases it could be a difficult and costly exercise depending on the feasibility of replacing the GPL-licensed code, the cost of a commercially licensed version (if available), the nature and scale of code being replaced, etc.

It is therefore important that software developers:

- Identify uses of GPL-licensed code in proprietary software developments; and

- Ensure the GPL licensing terms are fully understood and consistent with the intended distribution and licensing model of the proprietary software.

For more information on matters covered in this article, please contact:

Guy Burgess
Partner

Email: guy.burgess@clendons.co.nz
DDI: +64 9 306 8003
Level 5, 2 Emily Place, Auckland Central, New Zealand

This article is necessarily brief and general in nature and is not legal or professional advice. Readers must seek legal advice relevant to their specific circumstances. Please contact the professional staff of Clendons for advice specific to your situation.

---

[8] http://www.itworld.com/article/2732025/open-source-tools/gpl-enforcement-sparks-community-flames.html

[9] https://en.wikipedia.org/wiki/Apache_License