

About the LGPL

[teem](#)

NOTE: Prior to the release of version 1.9, the licensing terms for Teem changed from the LGPL, to LGPL plus exceptions which facilitate static linking (following [FLTK's](#) example). This changes addresses legitimate concerns about the practicality of using LGPL's code in binary-only applications. Specifically, Section 6 of the LGPL no longer applies to binary works linking with Teem. The licensing changes also renders moot many of the technical details of the write-up below (especially point #4), which I originally wrote in an effort to allay common fears and misconceptions about the LGPL.

It is also interesting to note, however, that while the text of the GPL and LGPL assert that linking with the library creates a derived work, some legal experts believe that linking creates a compilation, in the copyright sense, and not a derived work. This has implications for how people who normally resist reciprocal licenses like LGPL should understand their scope. Lawrence Rosen's [Open Source Licensing](#) covers this issue in Chapters [6](#) and [12](#)

My software, [Teem](#), is released under the [GNU Lesser General Public License](#) ("LGPL"), for a variety of reasons. Most importantly, this license best implements to the notion of "share and share alike" which is fundamental to academic research. Also, because the development of this software was largely funded by the tax paying American public, I feel obligated to prevent situations where my software is used but can not be inspected, understood, or altered at the level of source code.

It is important to understand the difference between the LGPL and the [GNU General Public License](#) ("GPL"). Some people dismiss all GNU licenses as "viral", which is both unfortunate and false. The most important difference between GPL and LGPL is that if your software uses Teem, you are in no way required to release the source for your software. However, because my software is not in the [public domain](#), or under a "[BSD-Style](#)" [free license](#), there are certain requirements. See [this page](#) for clarification on the different kinds of licenses. Keep in mind, however, that the LGPL-related requirements kick in **only** when you start distributing your software outside of your school, lab, or company.

After reading through the LGPL very carefully, I wrote up the following description of it, and the (minimal) responsibilities you have if you release software which uses Teem. Others have found it helpful because it clarifies whether or not the LGPL makes a distinction between static versus dynamic linking (the answer is no, and yes). Also, it enumerates exactly what you'd need to do if you release software that uses Teem (not much). Please read on.

The notion of a "derived work" is central to copyright law, because copyright concerns itself not just with a work, but with anything which is a modified version of the original work, or a translated version of the work, or which is (or contains) a significant portion of the work, and so on.

If you link my library with your program, the resulting binary is derived from my library (in the copyright sense) because the binary contains translated portions of my code. The GNU licenses make no distinction between static and dynamic linking **insofar as the definition of a derived work, as applied to compiled software, is concerned**. From the Preamble of the LGPL: "When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library."

One reason that the LGPL is a longer and more complicated document than the GPL is that the LGPL makes a distinction between different kinds of derived works, which the GPL does not make. Section 2(b) of the GPL applies to **any** derived work that you distribute or publish, requiring that the derived work be licensed under the GPL. [Read the GPL](#), for exactly what that entails; it involves distribution of your program's source code. The LGPL also says that it applies to all derived works. But the basic thing to understand is that **how** the LGPL applies in fact depends on the nature of the derived work:

1. If the derived work entails a modification of Teem itself or some portion of it, then you have a "work based on the library". Section 2(c) of the LGPL applies to that derived work as soon as you distribute it, requiring that the work be licensed under the LGPL.
2. If the derived work is a binary created by linking against Teem (or your modification of it), then you have a "work which uses the library". Section 6 of the LGPL applies to this work as soon as you distribute it, saying that the derived work can be distributed "under the terms of your choice, provided that ...", and goes on to describe the responsibilities of someone distributing the derived work.

The important distinction here is that those responsibilities, which I'll describe below, amount to something **less** than the full LGPL. Those responsibilities are mandated and defined by the LGPL (since the LGPL governs the distribution of any derived work), but they are **not the same as the LGPL**. Most confusion about the LGPL revolves around this basic property: Section 6 of the LGPL constrains (mildly) how you distribute compiled software using a LGPL'd library, but because those constraints are not the same as the LGPL, the LGPL does not propagate itself in the same way that the GPL does. This is in contrast to what is required when the derived work involves modification of my source, in which case the full LGPL does apply. However, if you have found bugs or API deficiencies in my software, please tell me, so that everyone can benefit. Since this software is how I do research, I'll be supporting it for quite a while.

So, if you distribute a program that links against Teem, without modifying Teem, Section 6 of the LGPL requires you to do four things:

1. Give notice that your program uses Teem, and that Teem is covered by LGPL. All this means is that in your documentation or descriptive material, wherever you other list other libraries that you depend on, you also mention Teem, and that its under LGPL. You would also mention Teem in an enumeration of the different licenses that govern different parts of your software, if this is the case.
2. Supply a copy of the LGPL. This is available online in [HTML](#) and [plain text](#). Its also the `Teem/src/LICENSE.txt` file in the Teem distribution.
3. If there is a section of your documentation that gives copyright notices for other code you use, or if copyright notices are displayed as part of the program's operation, then you need to show Teem's copyright notice as well:

Copyright (C) 2005 Gordon Kindlmann
Copyright (C) 2004, 2003, 2002, 2001, 2000, 1999, 1998 University of Utah

4. The interesting part: You need to make it possible for users of your program to modify my Teem libraries, and use those modified Teem libraries in your program. The principle of copyleft is that you pass on to others the same freedoms as were available to you. The five possible ways to facilitate this, listed in Section 6, can be summarized as two alternatives:
 1. If you **statically link** against Teem libraries, you need to accompany the distribution of your program with all source files for the required Teem libraries, and all the object (.o) files (or source files), for your program. This way someone could re-link your various object files together with modified Teem libraries to produce their own version of your program. You don't have to release any source for your program if you don't want to. (This much is from Section 6(a).) If you don't supply these materials alongside the distribution of your program, then they need to be conveyed or made available by some other means. (This is Sections 6(c)-6(e))
 2. If you only **dynamically link** against shared libraries, you're done! The nature of shared libraries means that your program will slurp up the Teem libraries (original or modified) at run-time. **You don't even need to distribute Teem source.** (This is Section 6(b))

I mentioned above that the LGPL does not distinguish between static and dynamic linking insofar as the definition of a derivative work is concerned. Section 6 of LGPL says that the distinction between static and dynamic **does** matter in the context of how to comply with LGPL, when you're releasing a program that uses Teem without modifying it. And this is how I imagine most everyone will use my libraries-- just linking against them as distributed.

I feel that the four points of compliance, described above, are clear, fair, and very easy to deal with, especially if you take the shared library route. Also, if you've read this far, please help by clearing up misunderstandings about the LGPL when you encounter them.