Legal Backgrounder

Washington Legal Foundation
Advocate for freedom and justice®
2009 Massachusetts Ave., NW
Washington, D.C. 20036
202.588.0302

Vol. 18 No. 20 June 6, 2003

OIL AND WATER, OR VINAIGRETTE?: "OPEN SOURCE" SOFTWARE MAY POSE LEGAL RISKS FOR COMMERCIAL USERS

by

Brian S. Kelly

Imagine that you are the CEO of a software company. You've helped your company navigate the treacherous waters of a down economy, and finally you can see that light at the end of the tunnel — a buyout to a well-heeled purchaser. What's even better, the initial offer reflects some of the hard work that everyone's put into the enterprise.

You endure grueling negotiations over the acquisition, but are reassured by the final endgame of a promising result for your employees and shareholders. You reach the eleventh hour of the process, when the purchaser calls you with some unexpected bad news. It seems that their due diligence on your company has revealed that deep in your product exists some code that was originally licensed to your company under the "GPL." Unfortunately, says the purchaser, the code has been in your product for a good while now, and several versions later, is scattered through much of the product. The purchaser has evaluated what it would take to tear out the GPL code and rewrite that functionality, and has factored that into the overall offer for the purchase of the company — which has now been cut by one third.

You hang up the phone, reach for the Excedrin, and wonder how to break the news to the employees and shareholders. How did this happen? And what is the GPL, anyway?

The rationale for open source software — the analysis of its basics and economics of open source licensing — has been well chronicled.[1] Instead, this LEGAL BACKGROUNDER notes the growing use of

[1] *See*, *e.g.*, Dr. David S. Evans, *Open Source Software Poses Challenges for Legal and Public Policy*, LEGAL BACKGROUNDER (Wash. Lgl. Fndt.) Jan. 31, 2003.

Brian S. Kelly is an intellectual property partner in the Washington, D.C. office of Fenwick & West LLP. His practice focuses on assisting vendors and consumers of technology with the identification, protection and commercialization of intellectual property assets.

open source software together with proprietary products and in corporate environments, and attempts to describe the unique problems caused by that activity and the impending lawsuits to come.

Open Source software has enjoyed explosive popularity in recent years. While this type of software (Linux in particular) may never live up to its initial hype as the compelling alternative to all things Microsoft, it has found its legs in certain industries — particularly in web servers, if not yet the desktop.[2]

Beyond Linux, it is worth noting that the popularity of Linux has inspired the open source licensing, of one flavor or another, of a growing number of disparate software products — from operating systems to applications to tools of most varieties.

Finally, the numbers of users in corporate America are growing. One recent survey found that over half of the programmers in corporate America currently use open source software in connection with their business. Of those, about forty percent use it without the knowledge of their boss.[3]

In most circumstances, these trends would be immensely encouraging to open source promoters (and to a large degree, they are). The problem occurs because at least some, and perhaps many, of those new users have incorrect assumptions about the legal impact of its use. Their assumption that open source code can be used freely (and often for free) is most often true. The companion assumption that such code can be modified and combined with proprietary code, and then have that combination redistributed in the traditional proprietary model, is unfortunately most often wrong.

The Different Models and the Viral Nature of the GPL. Open source licenses do come in a variety of flavors, with different licensing models being originated by everyone from the Free Software Foundation and Apache to the likes of Microsoft, IBM and Sun Microsystems. One of the oldest, and certainly the most predominant, is the General Public License, or "GPL." It is this license that will receive our focus here.

The GPL is a creature of the Free Software Foundation. It, like other open source licenses, gives the recipient a number of freedoms not found in traditional software licenses. It makes the source code of the software available to the user, does not restrict copying and redistribution of the software, permits the creation and redistribution of derivative works, and does not restrict classes of users. These rights are incredibly broad, and are wonderful for internal-use applications, or for software development tools.

The problem occurs with the distribution right, and what is commonly referred to as a "forcing" restriction. Under the GPL, if the user modifies the software and wishes to distribute the modification, then the entire work — including all modifications — cannot be redistributed under a typical proprietary license. The user is "forced" to redistribute the entire package under the GPL.

While this approach makes sense intuitively in circumstances where the modifications to the original product are comparatively nominal, the repercussions become troubling in other circumstances. There is a spectrum of possible "modifications" that could involve the GPL-licensed code in some way,

---

[2]For example, starting at a zero market share just three years ago, Linux has now grabbed 13.7% of the web server market, a market share expected to grow to over 25% by 2006. *See, e.g.,* Jim Kerstetter et al, *The Linux Uprising*, BUS.WK., Mar. 3, 2003, at 78.

[3]Ed Frauenheim, *Beyond Linux: Corporate Coders Step Up Their Use of Open Source*, WIRED, Dec. 2002, at 61.

  ISBN 1056 3059

and some of those uses lead to a counter-intuitive result. For example, what if a relatively small GPL product is dropped into a larger, existing proprietary product? What if the code to the original GPL product is not modified on its own, but is dropped into (or even simply linked with) an existing proprietary product?

Certainly proprietary software developers would prefer the answer to be no. It is in these latter cases that developers may download a small component licensed under the GPL, and (without awareness of the forcing clause) drop it into a commercial product due for redistribution, perhaps without even mentioning this fact to the management team. Unfortunately for them, while the language in the GPL is contradictory and ambiguous, overall it seems to suggest that in each of the above cases, the GPL would force redistribution of the entire, combined product under the GPL. It is this conclusion that essentially creates parallel licensing regimes — one for traditional software licenses that do not contain forcing clauses, and a separate one that accommodates forcing clauses. It is the common assumption that the GPL causes no problems with "mixing" the two regimes that traps the unwary.

Interpretations and Marketplace Reactions. Certainly the Free Software Foundation would agree with the conclusion that the GPL would require "forcing" the linked code to be licensed under the GPL. Recall that the FSF is the custodian of the GPL, and even for those products created by others where the author chooses to use the GPL as a licensing structure, the FSF's views are important. One of their stated goals is not just to encourage the use of open source software, but to make all software available at essentially no cost.

It is important to note, though, that even within the open source community there is not uniform agreement on the question of whether proprietary products that are simply linked to GPL-licensed code are forced to be open. Most commentators agree that where the links between the two are "static" — meaning that the connections between the two products are established when the product is compiled — then those kinds of links are deep enough, and the product is integrated enough, so that the forcing clause probably applies here.

If the links between the two are "dynamic" — meaning that the connections between the two products are made on a run-time basis, as the software is actually operating — then the evangelists within the community disagree. Richard Stallman of the FSF, for example, has argued that all linking — even dynamic linking — creates a derivative work that requires licensing the combination under this GPL. But Linus Torvalds, the contributor of the kernel for Linux, sees it differently. His opinion is that certain kinds of dynamic linking, such as when a proprietary application makes run-time calls on operating system functions, do not create a derivative work as contemplated by the GPL, and therefore should not be covered by the forcing clause of the GPL. (It is this interpretation as applied to Linux that may account in part for the growing acceptance and use of Linux and Linux-compatible applications in the corporate world).

To complicate the issue further, other commentators have argued that the type of linking is not determinative at all. Instead, practitioners should look at what kinds of files the "proprietary" contribution interacts with. If the application is linked to a GPL library, says this argument, then the application itself must be licensed under the GPL, and an application linked to a non-GPL library can remain proprietary.[4]

---

[4]*See, e.g.,* Jason B. Wacha, *Open Source, Free Software and the General Public License: An Overview of the History and Current Status of One of the Fastest Growing and Most Misunderstood Methods of Technology Licensing*, (continued...)

  ISBN 1056 3059

While not dispositive in itself, sometimes watching the reactions of other industry players in their licensing regimes can be helpful. Companies like Microsoft and Oracle have recently been placing "anti-open source" clauses in the licenses for their development tools that include run-time redistributables. The common theme of such licenses is that they prohibit a licensee from using the vendor product to build an application that also includes open source code, if the resulting product could be interpreted to make the redistributables subject to the terms of the GPL.

The reason that this paper has spent so much time trying to read the tea leaves of opinions of the open source community, and have not referred to case law on the GPL, is that to date there is no such case law. Though a few cases involving disputes over the GPL has been initiated, each to date had been settled — so there are no decisions on the books interpreting the GPL or what it means. Inevitably, that will change.

The Impending Suits. It will change in one of two ways. First, it could change once a user of GPL code has incorporated it into a proprietary product, and either disagrees with the "forcing" clause of the GPL code, or cannot afford to remove the offending GPL code as other potential defendants have done in the past, and is willing to litigate over it.

Other potential changes could occur not from a defendant unsatisfied with the GPL, but from a plaintiff. One high-profile suit has already been filed, by SCO Group — the inheritor of portions of the intellectual property to Unix — against IBM. SCO's suit alleges that IBM misappropriated SCO's trade secrets in Unix and incorporated such trade secrets into IBM's contributions to Linux. While at press time it will be difficult to predict the course of this suit, similar cases could lead to an interpretation of the GPL and also highlight another aspect — and potential weakness — of the GPL and open source projects generally: the risk of infringement. The GPL, in itself, does not include any warranties of title or indemnities for IP infringement, although a distributor can choose to offer it on its own if desired. Critics of open source products have pointed to this aspect of open source development as a unique deficiency: that because of the nature of open source development, a distributor will necessarily have less knowledge of or control over potential misappropriations that may be made by individual contributors to the project. This is a theoretical risk with all open source products, though with regard to Linux it appears that we may learn the real risk sooner rather than later.

Summary. We are in a unique time. The popularity of Linux, and of open source programming and licensing structures generally, are experiencing explosive growth not only for personal programming but also in the corporate world. But that corporate world is only slowly coming to the realization that the GPL represents a different kind of licensing regime. Mixing the oil of GPL code and the water of proprietary products is often problematic, and avoiding the forcing clause of the GPL will require a case-by-case analysis of the degree of integration between the two products. Looking to case law for reassurance about where this line lies, unfortunately, is currently of no help, though the battles are expected to come, and come soon. Given the rapidly increasing stakes in the open source world, the sooner we get that clarity, the better.

---

[4](...continued)
NEW MATTER, State Bar of California, Vol. 27 Issue 2/3, Summer/Fall 2002, at 14, 17. Mr. Wacha is the General Counsel for MontaVista Software, a Linux distributor.

ISBN 1056 3059