

# Open source, open world

## New possibilities for computer software in business

By Thomas M. Pitegoff

**E**ver heard the term “open source” referring to computer programs? Is the company you’re advising using it? If you think your background on this subject is a bit weak, read on.

Open source software has become an important component of corporate computer systems. Open source software differs from commercial software in the manner in which it is distributed. Commercial or proprietary software is typically distributed for consideration under a license that protects the proprietary rights of the publisher by limiting modification and copying. Open source software is distributed freely along with its underlying source code, allowing programmers to enhance and modify the software to suit a customer’s specific needs.

Proprietary software is written by programmers who agree to maintain the confidentiality of the code. Open source software is written as a collaborative effort, open to any programmer who desires to contribute. The open source license ensures that the software is distributed with its source code, which allows others to look under the hood and make modifications, something that is not possible with proprietary software.

Source code is a set of computer instructions written by a programmer

in a programming language that can be read by human beings. Once the source code is written, a programmer “compiles” it, using a compiler program, into binary or “object code” so that it can run on computers. Object code appears as strings of ones and zeros.

The source code of proprietary software distributed under a traditional software license remains secret. Software companies strive to maintain their competitive edge by keeping the source code out of the hands of potential competitors. While source code is protected by copyright and may be protected by patent, in practice source code is protected by trade secrecy. Even if the source code copyright is registered with the U.S. Copyright Office, only a small portion of the code need be filed on registration, while most of it typically remains secret. Traditional software license agreements commonly prohibit reverse engineering that would enable a programmer to decipher the source code.

When a company acquires a license to use proprietary software that is important to its business, that company wants to know that its licensor will endure as a going concern so that it can fix bugs in the software and enhance and upgrade it. A licensee with adequate bargaining power may succeed in having the source code placed in escrow with an independent software escrow company, thereby providing some assurance that the source

code will be available to the licensee in the event that the licensor goes out of business, is acquired or decides to stop supporting the software.

Open source turns the traditional software business model on its head. There is no need for a source code escrow. Both the source code and the object code are free and freely available over the Internet. This allows any company to customize the software to its needs. Anyone in the world can improve the code for the benefit of everyone.

The open source approach promotes rapid software development. It also promotes reliability and stability in the software by means of independent peer review and widespread testing. This process also requires that the source code be well documented, something that is too often missing in software programs developed in-house. With a successful open source project, there is a large community of computer professionals who can support the software.

The open source license uses a novel inversion of intellectual property law to promote rather than restrict the copying, use and modification of software. Instead of guarding exclusive rights, the open source license ensures free access, use and modification of software. While free, open source software nevertheless relies on intellectual property rights.

Microsoft has argued that open source software undermines intellectual property. Specifically, the company has attacked one open source

---

*Pitegoff is the founder of Pitegoff Law Office in White Plains, N.Y.*

license that requires those who modify the software to distribute the source code of the derivative work freely. The open source community sees this as a scare tactic in response to the rapid expansion of the popular Linux operating system, the most successful of all open source projects.

While Microsoft's campaign against open source software may be an exercise in hyperbole, there is nevertheless a grain of truth in it. Corporations are rightfully concerned that their programmers may modify open source software using proprietary code. If that happens, the open source license can require that this proprietary code be licensed to the public.

Notwithstanding the risk, the role of open source software in business has grown exponentially in recent years, along with the expansion of the Internet. Open source has resulted in enhanced competition and innovation and raises the hope of market-based competition to Microsoft's monopoly. Open source enables programmers around the world to collaborate on productive projects. It allows for standardization without monopolization. Even Microsoft has taken the small step of making the code of its next Windows version available to a limited group of customers under stringent conditions.

Open source is not about to destroy proprietary or commercial software. Proprietary and open source software can and do coexist peacefully in many corporate computer systems. Companies that develop, sell and integrate software, as well as companies that do not sell software but desire to

improve, enhance or modify software, can and will continue to use proprietary software when it gives them a competitive advantage. In some situations, though, these companies find open source useful. Ideally, a company should consider *both* open source and proprietary software for any given project and use the one that will do the job best.

One reason for the success of open source software has been the explosive growth of the Web, which speeds contributions by programmers around the world. Another reason is the obvious fact that it's free, allowing users to install it on countless computers without corporate budget approval. Installing traditional software can be very expensive, especially for a large company.

An incidental benefit of open source is that it solves a company's software piracy problem. Unauthorized copying of proprietary software is a violation of copyright law and of most software licenses. Piracy costs software companies billions of dollars in lost revenue each year. Yet it is rampant in companies of all sizes, including law firms. [See "May I See Your License," *ABA Journal*, April 2001, page 74.]

Linux, the most widely used open source program, is the fastest-growing operating system for computer servers. It is secure and stable, and because it requires far less computer memory than Windows, it runs well on older machines that would be incapable of running Windows. Linux is even being used on some handheld devices. While Linux is more common on servers than desktop computers, a growing number of desktop applications are available to Linux users.

IBM has become a strong advocate of Linux, pledging this year to spend more than \$1 billion promoting the open source operating system. Linux frees IBM from a dependence on Microsoft and allows IBM to offer more hardware and services to its customers for lower prices than when it sells proprietary operating systems.

IBM is not the only major computer company that supports Linux. Compaq, Dell and HP also offer Linux computers. An increasing number of enterprise applications also run under Linux. These include Oracle, SAP and Corel applications, as well as Lotus Domino and Tivo. Apache is a widely used open source Web server program that runs under Linux. Samba is an open source program that makes Linux look like a Windows file and a print server to networked computers running under Windows.

Open source developers commonly use one of several standard form licenses for their development projects because these licenses are familiar to the open source community. The range of standard open source licenses is broad enough to allow a project leader to select one that will be appropriate for the project. All of the open source licenses permit users to read the source code, redistribute, modify and use it freely.

Linux uses the General Public License, or the "GPL," the most widely used open source license. The GPL has an illustrious history. In 1984, Richard Stallman, the founder of the Free Software Foundation, began developing an open source operating system. He called it "GNU," a recursive acronym for "GNU's Not Unix." The Free Software Foundation formulated the GPL.

In 1991, Linus Torvalds, then an undergraduate student at the University of Helsinki, posted the kernel of an operating system on the Internet and invited the world to help him develop it. Other programmers responded and the collaborative development began, in part by combining the system's kernel with GNU. Linus Torvalds is still the project leader and chief engineer of Linux, although he now works for Transmeta, a computer chip manufacturer.

## Books on open source

Russell C. Pavlicek, *Embracing Insanity: Open Source Software Development* (2000).

Donald K. Rosenberg, *Open Source: The Unauthorized White Papers* (2000).

Eric S. Raymond, *The Cathedral & the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary* (1999, 2001).

The GPL allows licensees to use, copy, distribute and modify the software. Derivative works must be licensed as a whole at no charge to third parties. In other words, anyone who modifies or enhances software licensed under the GPL agrees to make the resulting code freely available under the same license, along with the entire source code. Any patents must also be licensed for everyone's free use.

The BSD license (Berkeley Software Distribution) is less restrictive than the GPL. It allows the licensee to alter the software and distribute the derivative version as object code without the corresponding source code. It also allows for the distribution of derivative works as commercial versions. Because it is less restrictive, the BSD license can result in "forking," or the development of various versions of the same software. BSD was the license under which the University of California at Berkeley distributed an open source version of the UNIX operating system in the 1980s. Even Microsoft uses the FreeBSD open-source operating system on servers that manage its Hotmail free e-mail service.

Under the Library (or Lesser) General Public License (LGPL), a proprietary software program such as a device driver or a library of software functions can be loaded by the Linux kernel without bringing the proprietary driver or module within the GPL. In this way, a manufacturer can maintain the secrecy of its hardware device while making it available to work with Linux and other open source programs. Other variations of open source license include the Mozilla Public License, the Sun Community Source License, the IBM Public License, the MIT or X Consortium license, the Perl Artistic License and the FRS license.

The license that is of the most concern to businesses is the GPL because any program containing a software element licensed under the GPL, and any program derived from GPL software, must itself be distributed under the GPL. Because of this characteristic, the GPL is sometimes referred to as a "viral license."

## Open source Web sites

Open Source Initiative – [www.opensource.org](http://www.opensource.org)  
Apache Software Foundation – [www.apache.org](http://www.apache.org)  
Free Software Foundation – [www.gnu.org](http://www.gnu.org) (or [www.fsf.org](http://www.fsf.org))  
Linux Documentation Project – [www.linuxdoc.org](http://www.linuxdoc.org)  
Metalab Linux Archive – [www.ibiblio.org/pub/linux](http://www.ibiblio.org/pub/linux)  
Debian Free Software Guidelines –  
[www.debian.org/social\\_contract.html](http://www.debian.org/social_contract.html)  
LinuxWorld – [www.linuxworld.com](http://www.linuxworld.com)  
Linux Today – <http://linuxtoday.com>  
Linux Online – [www.linux.org](http://www.linux.org)  
IBM – [www.ibm.com/linux](http://www.ibm.com/linux)  
Source Forge – <http://sourceforge.net>

Section 2(b) of the GPL provides that a licensee who distributes new software that "contains or is derived from" open source GPL software must distribute the new software under the same GPL:

You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this license.

A company can avoid the viral nature of the GPL by exercising care. Only those modifications and enhancements that are written into the GPL code or compiled with it become part of the GPL code. Section 2 of the GPL goes on to provide, in part, as follows:

If identifiable sections of that work are not derived from the program, and can be reasonably considered independent and separate works in themselves, then this license, and its terms, do not apply to those sections when you distribute them as separate works.

Using this distinction, companies distribute proprietary files, programs and drivers on the same CD-ROM as a GPL program. The GPL does not attach to these independent components, even if they work in conjunction with the GPL program. The GPL also does not attach to proprietary applications that run under Linux.

As long as a company is doing nothing more than installing open source software on its systems and using it, there is no legal issue. All open source licenses allow for the free use of the software.

At the user level, there is a superficial similarity between the management of open source licenses and the management of shrink wrap licenses. Neither the open source license nor the shrink wrap license is signed by the parties. Both can easily be installed on any PC without management's knowledge. For these reasons, just keeping track of these licenses is a challenge. That is particularly true of open source licenses, which can be obtained without any company expenditure. The difference between the two licenses, though, is fundamental. With shrink wrap licenses, a company must monitor the distribution of unauthorized copies, not the failure to distribute code.

Corporate users may be concerned by the fact that open source licenses conspicuously disclaim all warranties. What if a company is damaged by its use of open source software? Could the company challenge the open source warranty disclaimer? If the company succeeds in its argument that the disclaimer is unenforceable, who stands behind an implied warranty when the software is downloaded from the Internet for free?

In reality, there is little need to be concerned with the open source warranty disclaimer, which makes sense

when the software is distributed for free. Users can draw on an army of open source vendors who will provide the required support for a fee. In fact, most commercial software licenses also conspicuously disclaim all or essentially all warranties. In practice, it is likely that a bug in an open source program will be fixed more quickly than one in a commercial software program because of the speed of the Internet and the pride of the developers. Moreover, a corporate user can often have the bug fixed in-house because the source code is available. With proprietary software, the company would have to wait for the next release of the software.

Computer companies and any enterprise with an IT department that can modify and enhance open source

etary code that the company does not want to publish, or using code licensed from a third party. Incorporating proprietary third-party software into GPL software would constitute a violation of the third party's license and copyrights, and possibly the third party's patent, and would contaminate the resulting code.

A company working on a GPL development project should follow the GPL requirement of releasing the derivative work to others. If the company has any hesitation doing so, it should not work on a development project with GPL code.

What if a company uses GPL material and does not subsequently make the derivative work available to others?

What if the company decides to sell the

seeks to stop poachers of open source software. The fear of bad publicity may be reason enough to observe the terms of the open source license.


Open source licenses also raise issues over which a company may have little control. For example, what if the open source software used by the company already includes infringing material from an earlier "contributor"? To what extent would the company be liable for copyright or patent infringement by a programmer higher in the open source development chain? Would the innocent company with the deep pockets suffer? The answer is not at all clear.

What about damages? Because the intellectual property owner receives no compensation, the open source licensor would suffer no actual monetary loss. Even if damages are available, such as statutory damages in copyright, how would damages be allocated? Countless programmers contribute to a GPL project.

The fact that broad rights are granted with an open source license might also make it difficult to obtain an injunction. The GPL contains one basic requirement. Modified software must be freely redistributed with its source code under the same license. Who is irreparably harmed if a licensee distributes a derivative work without the source code and license?

Does U.S. law even apply to the GPL? The Linux project was coordinated by a project leader in Helsinki, Finland, with contributions from programmers around the world. Open source is becoming increasingly popular in a number of countries. The GPL is silent on the question of governing law.

While these are interesting questions, they are likely to remain theoretical for some time. Unless the courts or Congress declare open source licenses unenforceable, companies using open source software should assume that these licenses are enforceable and act accordingly. Even though open source licenses pose risks, the benefits of using open source software usually outweigh the risks.

There is a growing place for open source software in business. 

## To manage the risk, a company needs controls.

software must be concerned about the viral nature of the GPL. They need internal policies regarding the use of open source software. These companies typically require their employees to sign agreements assigning all inventions to the employer, including all computer code developed by the employee. Because the company owns and is responsible for the work product, company counsel should be involved in all development efforts. Employees should understand that the determination whether to proceed with any GPL project is a company decision, not that of an individual employee.

Companies that have the capability of modifying and enhancing software are rightfully concerned that their own proprietary software may inadvertently become open source as a result of the company's use of GPL software, thereby licensing their software to the public without compensation to the company. To manage this risk, a company needs to have controls in place.

Programmers must not modify or enhance a GPL program using propri-

etary code as a proprietary program? This could happen without the knowledge of management or counsel, because open source software is so easily accessible.

Open source licenses have not yet been tested in court. For this reason, it is not clear whether the open source license is enforceable. If it is enforceable, a breach of the open source license would constitute copyright infringement. But whose copyrights are infringed? One might argue that the open source license is so broad that it is tantamount to placing the software in the public domain. The entire copyright in the software is essentially given away free with the license. If so, how can users be bound by the obligation to disclose the code of derivative works and freely distribute them?

Arguably, the open source license is enforceable to the same extent that a click wrap or shrink wrap license is enforceable. Even if the open source license is not enforceable, misuse of the GPL can result in bad publicity for the company that violates its terms. The Free Software Foundation actively