

White Paper
January 2008

License to Code: Indemnifying Your Business Against Open Source Licensing Liabilities



ActiveState

Open Source in the Enterprise

Not long ago, open source software in the enterprise was a contentious topic. Conference panels and industry experts questioned the integrity of mixing open source with commercial and proprietary code. Would it unlock a Pandora's box of developer pain and sub-par software? Amidst the heated debate, open source tiptoed its way into the business of large-scale software development and settled in to stay.

Last year, a Forrester study showed 75% of large enterprises surveyed were either using or planning to use open source software.

Last year, a Forrester study showed 75% of large enterprises surveyed were either using or planning to use open source software. That was up from 60% the year before. In a study from *InfoWorld* magazine, 87% of respondents said their companies were already using open source in several projects—even for mission-critical applications. In 2007, The451 Group confirmed it: open source software is gaining worldwide adoption in commercial software companies.

And so, the tides turned. However, just because open source is ubiquitous doesn't mean it's risk-free. Many businesses widely use open source in commercial projects, but are blind to what's at stake when they distribute software that includes open source code. Continue reading and learn how to both protect your business and capitalize on opportunities when traversing the legal landscape of open source.

Everyone Is Doing It

Some of the biggest names in software development and manufacturing use open source software in their projects—Microsoft, Toshiba, even the U.S. Navy. There are many good reasons to do so.

For starters, open source languages decrease the expense of software development considerably.

For starters, open source languages decrease the expense of software development considerably. That's because you don't have to buy seat licenses for all your developers. However, being a cheap alternative to commercial software is not the only reason open source has caught on in the enterprise.

The ever-improving, extensible nature of open source languages appeals to developers. They know the code is constantly adapting, evolving and stabilizing. That's thanks to a dedicated, generous community whose passion for the code is rivaled only by a New York Yankees fans' passion for baseball.

Of course, another advantage of open source is being able to tap into the global open source community when things go sour. When problems arise—as they often do—enthusiastic members of the open source community see your headache as a riveting weekend project. They love the code; they fix the code; they are the code.

Why Open Source Isn't the Holy Grail

Despite all the benefits, any development team using open source for enterprise software development projects knows it's not perfect:

When you are on a tight development schedule, a lack of formal support or training can jeopardize a development project.

- Programmers in their basements can't fix your problem fast enough. The community of open source coders isn't on the clock, or your company's payroll. That means free 24/7 support doesn't exist for open source projects. When you are on a tight development schedule, a lack of formal support or training can jeopardize a development project.
- As a commercial software vendor, you must stand behind your product. That gets tricky if your software uses open source, because you don't have complete control over the entire code base. Yet, your company becomes accountable for the total quality of your product.
- If the open source software you're using is not top quality, it can cause a ripple effect that destabilizes your product and reflects poorly on your business.
- Most companies don't have policies in place to address administrative complexities that go along with using open source languages—like how to license your product. *InfoWorld's* survey indicates that only 41% of respondents had a policy in place for deploying and managing open source in their companies.

The Licensing Problem

In the *InfoWorld* survey, respondents said one of the major challenges of using open source languages in enterprise software development is understanding and enforcing open source licensing.

For many businesses, licensing is the granddaddy of policy problems. The confusion begins with terminology. When developers call software “free,” they mean users are free to run it, change it and redistribute copies with or without changes. As Richard Stallman—author of the first GPL open-source license—puts it, when you think of open source languages, “think of ‘free speech,’ not ‘free beer’¹.”

When using open source software is as easy as downloading it from the Internet, when is free really free?

¹ Stallman, Richard. “Why “Open Source” misses the point of Free Software.”

Licensing Myths

1. *Open source software is free and doesn't need to be licensed.* In truth, very little software is in the public domain.
2. *All open source projects use the same license.* Most open source software is a hodgepodge of different tools, all with their own licenses. There are plenty of unique licenses and licensing scenarios. For instance, we know of a case where an open-source license requires users to buy copies of the author's book for every member of their development team.
3. *License information is always in the fine print.* Sometimes you can find licensing information when you download the software. More often you need to sift through the source code looking for the license. Sometimes you even need to contact the community to answer your questions.
4. *It says "free," so it must be free.* Even when software is labeled as "free," always be sure to check the actual distribution terms to see if users really have all the freedoms that the term "free" implies.

Licensing 101

An open-source license is a copyright license for computer software that makes the source code available under terms that allow for modification and redistribution without having to pay the original author.²

Sounds like free software, but it's not. Open source code must be licensed. Although open-source licensing terms may have nothing to do with money, they will put restrictions on how you can distribute your product.

The theory behind open-source licensing is a good one. It ensures that open source languages are not exploited by organizations that use them in their projects, but otherwise would not give anything back to the community. The problem with open-source licenses is that they can be difficult to understand, which makes following their rules like navigating an uncharted trail.

GPL License: The Mother of All Open-Source Licenses

The vast majority of free software projects are licensed under the GPL License. The latest version of the GPL (version three) was released in June 2007. The GPL stipulates that you can only share, give away or sell your software if you include the source code with your application. In other words, when commercial software vendors integrate open source languages in their commercial products, they must include the open source code with their product, including any changes they made to it along the way.

Artistic License: For More 'Sensitive' Scenarios

Artistic License is another open-source license, and is commonly used for Perl implementations. The license is a reference to dramatic or poetic license. In this scenario, the copyright holder maintains some control over the kinds of changes that can be made to the code. Artistic License is notoriously confusing. Certain passages are so vague that it's rarely used on its own and is often paired with the GPL.

² "Open Source License." [Wikipedia](#)

Pick a License, Any License

GPL and Artistic License are two common open-source licenses, particularly for Perl implementations. However, many others make up the mix. Here's a small sample of some of the alternatives you need to be familiar with if you're using open source software in your company.

Title	Source	License	Clauses
No License	Open	None	Without a license, the code is copyrighted by default. People can read the code, but they have no legal right to use it. To use the code, you can contact the author directly and ask permission. Though, permission may not be enough. For copyrighted material, you often need to get a lawyer involved.
Public Domain	Open	Permissive	If code is in the public domain, anyone can use it for any purpose.
GPL License	Open	Copyleft	The ubiquitous open-source license. Requires all contributed code to be returned to the community.
Artistic License	Open	Copyleft, with some restrictions	Copyright holder maintains some semblance of artistic control over the development of the package, while giving the users of the package the right to use and distribute the Package in a more-or-less customary fashion, plus the right to make reasonable modifications.
LPGL License	Open	Mostly Copyleft	GPL with a twist. Open source software can be binary linked to proprietary programs in certain circumstances.
MIT/X11 License	Open	Permissive	Includes generic legal disclaimer of liability.
BSD License	Open	Permissive	Includes legal disclaimer of liability with explicitly named organization.
Apache License	Open	Permissive	Requires derivative works to provide notification of any licensed or proprietary code in a common location.
Eclipse Public License	Open	Permissive	Allows derivative works to choose their own license for their particular contributions.
Mozilla Public License	Open	Weak Copyleft	Allows liberal mixing with proprietary software.

Many IT managers don't realize that dependent modules are separately licensed.

Deciphering Module-Level Dependencies

Once you've got the basic license models under your belt, the next step is to decipher module-level dependencies. Open source languages are made up of thousands of libraries, modules, packages and frameworks, each with their own unique licenses. Many IT managers don't realize that dependent modules are separately licensed, so they don't fully grasp how to license their commercial implementations of open source software.

What are the Legal Risks of Using Open Source in Your Business?

What do all these licenses mean, and what happens if you don't abide by them?

The legal risk of working with open source languages is being hit with a breach of contract lawsuit or a patent infringement lawsuit.

Thanks to the open source community's rather polite culture, there haven't been many breaches of contract lawsuits to date, but they're not unheard of. Some of the high-profile cases have sparked concern about legal exposure when businesses use open source software in their projects, and have brought licensing front and center as a serious business risk.

You Can't Ignore the Problem

Some businesses, financial institutions in particular, think they're off the hook because they prohibit or discourage the use of open source languages in their software development. However, simply saying "no" to open source won't keep it out of your organization.

In today's development environment where budgets and timelines are tight, it's rare that a company can avoid open source altogether. Additionally, widespread standards that assume you're using open source can make 'abstinence' virtually impossible.

Licensing Mishaps

1. JBoss Blues

In 2006, FireStar Software slapped RedHat/JBoss with a patent infringement lawsuit. It claimed JBoss infringed on its technology patent.

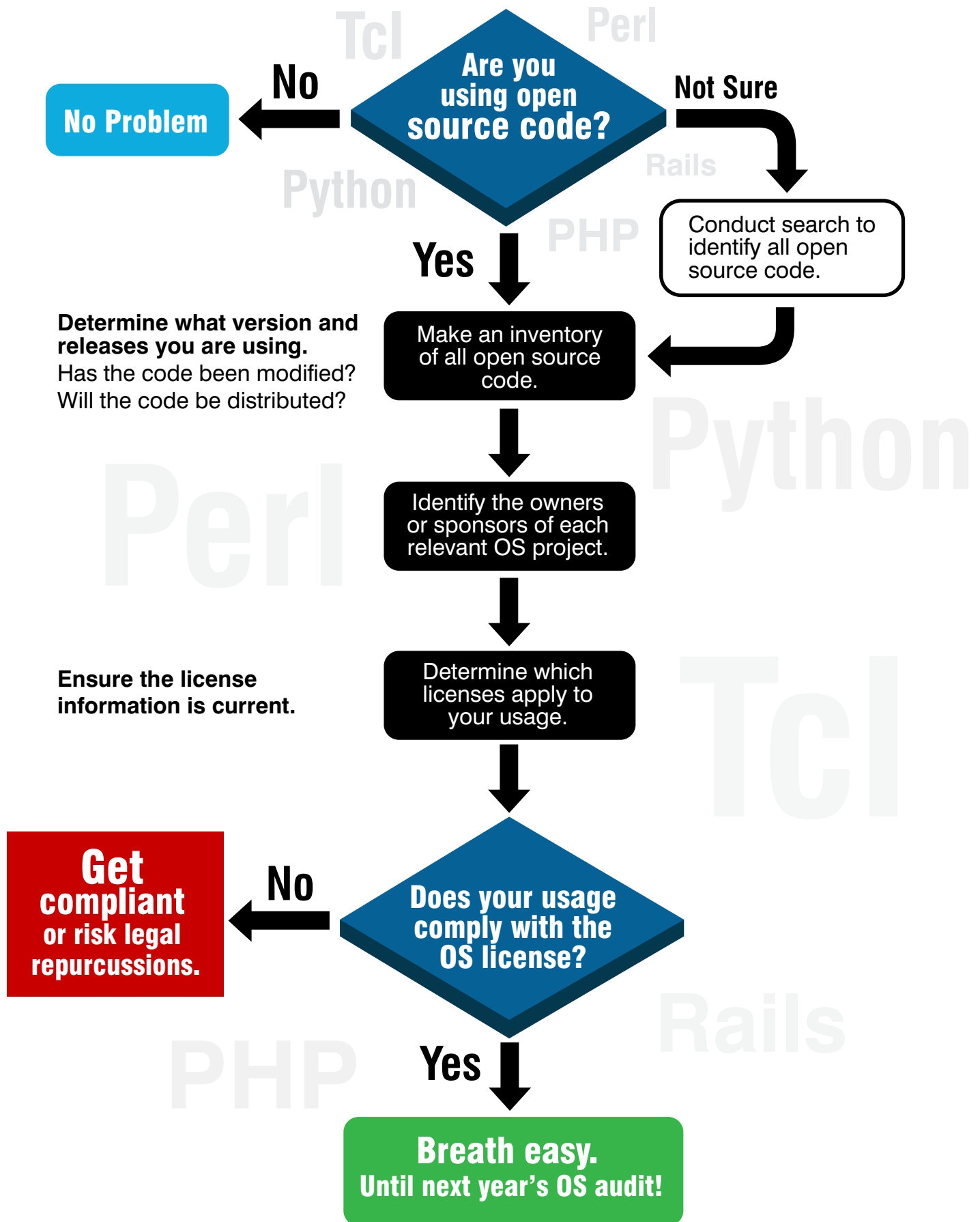
The results of this case are pending, but it could set the stage for more open-source license infringement law suits.

2. Sorry, Sony

In 2005, Sony woke-up to a PR nightmare when the open source community claimed the entertainment giant's controversial copy-protection software infringed on the copyrights to several open source software projects.

The irony, of course, is that Sony's copy-protection software prevents users from copying a CD more than twice.

How to run an open source audit for your business



Make Sure You're Covered

There's no single tool or strategy for making sure you choose the right open-source license for your project. To ensure that you're indemnified against legal action, you've got to be patient, diligent, detail oriented and, ultimately, right. For many, getting it right begins with an open source audit.

Typically, an audit follows these steps:

1. Evaluate the scope and nature of how your organization uses open source software. Where and how are you using it? Are you using it internally, or will it be distributed? Is it modified or unmodified? Has it interacted with other code—proprietary or open source?
2. Make a detailed inventory of which open source languages you're using. Note which version and release numbers you're working with, and who owns or sponsors the project.
3. Then, research and learn about the licenses for all the open source projects your organization is using. Make a library of the names of all the licenses, and get copies of all these licenses.

Open source audits can be resource and time intensive. Once you've gone through the hassle of evaluating, documenting and learning, there are still plenty of questions. Is your audit accurate? Did open source usage change before you completed the audit? When will you have to do it all again?

If the thought of running a licensing audit makes you queasy, there's another surefire way to make sure your organization is indemnified against open-source licensing risks—pass the compliance hot potato over to a licensing expert.

No-Brainer Compliance: Out-of-the-Box Licensing

Swift adoption of open source in the enterprise has created a demand for stable distributions and comprehensive support and maintenance for open source deployments.

Swift adoption of open source in the enterprise has created a demand for stable distributions and comprehensive support and maintenance for open source deployments. As such, third-party companies began offering safety nets for businesses deploying open source software. So, when the opensource community does not provide answers to common developer pains, companies like ActiveState jump in with superior commercial offerings.

ActiveState recognized early on that businesses distributing commercial implementations of open source were taking big risks when it came to code stability, unreliable technical support and potential license infringement. So, it developed enterprise-level open source distributions that have become renowned for quality and are now the de-facto standards for millions of

developers around the world. Like all open source code, ActiveState language distributions are provided free to the community.

Protect Your Business with ActiveState OEM Licensing

When you use ActiveState enterprise distributions—ActivePerl, ActivePython or ActiveTcl—you can obtain OEM licensing packages that will protect your company from legal exposure and improve the open source development process. Here's how it works.

Protect Your Organization

ActiveState OEM Licensing puts an end to legal paranoia because your company is no longer responsible for licensing the open source components of your ActiveState software. ActiveState enterprise distributions are guaranteed to comply with all licensing requirements, so you can deploy your software worry-free.

Cost Effective Licensing

At first glance, managing open-source licensing on your own may seem the most affordable option. However, it often adds to spiraling costs that go hand-in-hand with building your own open source distributions. ActiveState OEM Licensing can be a more cost effective way to license Perl, Python and Tcl builds because it guarantees instant, quality builds and fixed licensing expenditure.

Save on Time and Resources

OEM licensing is an appealing alternative to the dreaded open-source audit. ActiveState's licensing solutions provide instant, royalty-free distribution rights. Out-of-the-box licensing saves you time, personnel resources and the headaches that accompany managing license compliance on your own.

Faster to Market

You can save days and months of developer time with quality, pre-configured, pre-tested Perl, Python and Tcl builds. Enterprise distributions aid faster software development by eliminating the time it takes to build, test and maintain your own distributions. As well as getting to market faster, ActiveState's quality guarantee boosts confidence in your commercial software project.

Access to Open Source Experts

OEM Licensing gives your developers access to ActiveState's open-source experts. ActiveState developers are supporters and contributors to open

ActiveState enterprise distributions are guaranteed to comply with all licensing requirements, so you can deploy your software worry-free.

ActiveState offers unlimited support incidents each year for companies that don't want to gamble on their commercial software projects.

source languages and maintain healthy, active relationships with the open source community. Your developers can lean on ActiveState language experts for design and development advice.

On-Demand Professional Services

Counting on the open source community for technical support can be a crapshoot. ActiveState offers unlimited support incidents each year for companies that don't want to gamble on their commercial software projects. Other professional services include technical support for your developers backed by Service Level Agreements and co-marketing packages that capitalize on the industry-recognized quality of ActiveState's enterprise builds to enhance your product's value.

Conclusion

Building great software is hard enough without having to wrestle with the multi-headed monster of open-source licensing. Negotiating the labyrinth of licensing alternatives is time-consuming, risky and often outside of your staff's core competencies. Plus, the legal risks associated with failing the compliance test are significant.

ActiveState OEM Licensing offers peace of mind for enterprises working with open source software. ActivePerl, ActivePython and ActiveTcl are comprehensive software, support and maintenance packages for organizations of all sizes that depend on Perl, Python and Tcl. Safeguard your company's applications with guaranteed, quality-assured binaries and mitigate risk with world-class support for your critical systems.

Learn more at http://www.activestate.com/business_solutions/.

Who is ActiveState?

ActiveState has been a player in enterprise-level open source distributions since 1997. The company's developers are supporters and contributors to open source languages—including Perl, Python and Tcl—and maintain healthy, active relationships with the open source community.

ActiveState creates professional software development tools, programming language distributions and business solutions for dynamic languages, and practical tools and applications for social networking platforms. ActiveState solves complex problems and produces cool and useful applications that help people live more productive and enjoyable lives. ActiveState technology “just works.” ActiveState is owned by its employees and Pender Financial Group, a private merchant bank focused on technology in British Columbia. For more information, visit www.activestate.com.

ActiveState



ActiveState Software Inc.

1700-409 Granville Street
Vancouver, BC V6C 1T2

Phone: +1.778.786.1100
Fax: +1.778.786.1133

Sales

enterprisesales@activestate.com
Phone: +1.778.786.1101

Toll-free in North America
1.866.510.2914