# LICENSE TO CODE

HOW TO MITIGATE
OPEN SOURCE
LICENSE RISKS

ActiveState®

# LICENSE TO CODE

## ENTERPRISES RELY ON OPEN SOURCE

A little over a decade ago, it was commonplace to hear industry experts questioning the integrity of mixing open source software (OSS) with commercial and proprietary code. Today, however, the world's largest enterprises (and even the US Government) have recognized the benefits of leveraging third party code, which include:

- Speeding up the development process.
- Decreasing licensing costs.
- Leveraging a global resource of dedicated, passionate open source developers who often provide faster innovation than commercial vendors can.

As a result, it's not uncommon to find modern software that is comprised of up to 90% third party code, the bulk of which is open source. According to Gartner, as much as 95% of IT organizations leverage open-source software within their mission-critical applications. [1] Open source security vendor, Black Duck Software also points to the fact that at least 78% of all enterprises use open source software offerings in some capacity. [2]

**95%**

According to Gartner, as much as 95% of IT organizations leverage open-source software within their mission-critical applications.

## COMMERCIAL USE NEEDS ENTERPRISE PROTECTION

However, just because open source is ubiquitous doesn't mean it's risk-free:

- Commercial software vendors must stand behind their product. That gets tricky if your software uses open source, because (depending on the license) you may not have complete control over the entire code base.
- Most businesses use open source in their commercial projects, but may not be aware of their obligations when they re-distribute it.
- Using open source software for internal-facing projects doesn't exempt you from the obligations inherent in its license.
- Open source software that is outdated or poorly maintained can destabilize your mission-critical application, or even provide exploitable attack vectors for bad actors.

Continue reading and learn how to both protect your business and capitalize on opportunities when traversing the legal landscape of open source.

# LICENSE TO CODE

## FREE HAS A PRICE

Even today, enterprises are challenged to track, enforce and verify proper open source licensing in their products. The confusion often begins with terminology. When developers label software as "free," they mean users are free to run it, change it and redistribute copies with or without changes. However, it doesn't necessarily mean "free of any obligations." As Richard Stallman— author of the first GPL open source license— puts it, when you think of open source, "think of 'free speech,' not 'free beer.'"

A typical 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. Because there's no money changing hands or contracts being signed, OSS really does sound like free software. But incorporating OSS into your project often comes with licensing terms that place obligations on how you can distribute your product.

For example, the grandfather of open source licenses, the GNU General Public License (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.

***Because there's no money changing hands or contracts being signed, OSS really does sound like free software. But incorporating OSS into your project often comes with licensing terms that place obligations 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 can make following their rules like navigating an uncharted trail.

## TRAVERSING THE LEGAL LANDSCAPE

Open source licenses come in all shapes and sizes. The following table includes many of the most common 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 must contact the author directly and ask permission, but you will likely want 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 | Requires all contributed code to be returned to the community. |
| LPGL License | Open | Mostly Copyleft | GPL with a twist. Open source software can be binary linked to proprietary programs in certain circumstances. |
| MIT | Open | Permissive | Includes generic legal disclaimer of liability. |
| BSD 2 (“Free”) | Open | Permissive | Includes legal disclaimer of liability with explicitly named organization. |
| BSD 3 (“New”) | Open | Permissive | Identical to BSD 2, but also prohibits the use of contributor names in endorsing derived works. |
| Apache 2.0 | Open | Permissive | Requires derivative works to provide notification of any licensed or proprietary code in a common location. |
| Eclipse Public | Open | Permissive | Allows derivative works to choose their own license for their particular contributions. |
| Mozilla Public | Open | Copyleft | Allows liberal mixing with proprietary software. |

Adapted from “Pick a License, Any License” and “Licenses & Standards”

# LICENSE TO CODE

### LICENSING MYTHS

**1.** Open source software doesn't need to be contractually licensed. In truth, very little software is in the public domain.

**2.** All open source projects use the same license. In reality, most OSS is made up of numerous libraries, packages and frameworks, each with unique licenses that must be dealt with separately.

**3.** License information is always in the fine print. In practice, sometimes the license is downloaded with the software; sometimes it's in the source code, and other times you may need to contact the author/ community to answer your license questions.

### WHAT ARE MY RISKS?

The most common result of improperly implemented open source licenses is being hit with a patent infringement or breach of contract lawsuit.

Thanks to the open source community's rather polite culture, there have only been a few high-profile lawsuits to date. Based on the 2006 ruling by the Federal Circuit Court of Appeals, open source license violations are typically treated as copyright claims, and primarily settled out of court for undisclosed sums of money. Organizations such as gpl-violations.org have been raising the public awareness of OSS licensing and helping OSS authors pursue businesses that violate the GPL license since 2004. These cases have been generally settled out of court, and the settlements typically re-invested into open source software projects, providing support for the community.

However, times are changing. For example, Linux has over 15,000 contributors whose code is covered under a single GPL license, with each contributors' individual rights existing side-by-side with rights in the project as a whole. As a result, each author would generally own the copyright on their individual contributions, but not the Linux kernel in its entirety.

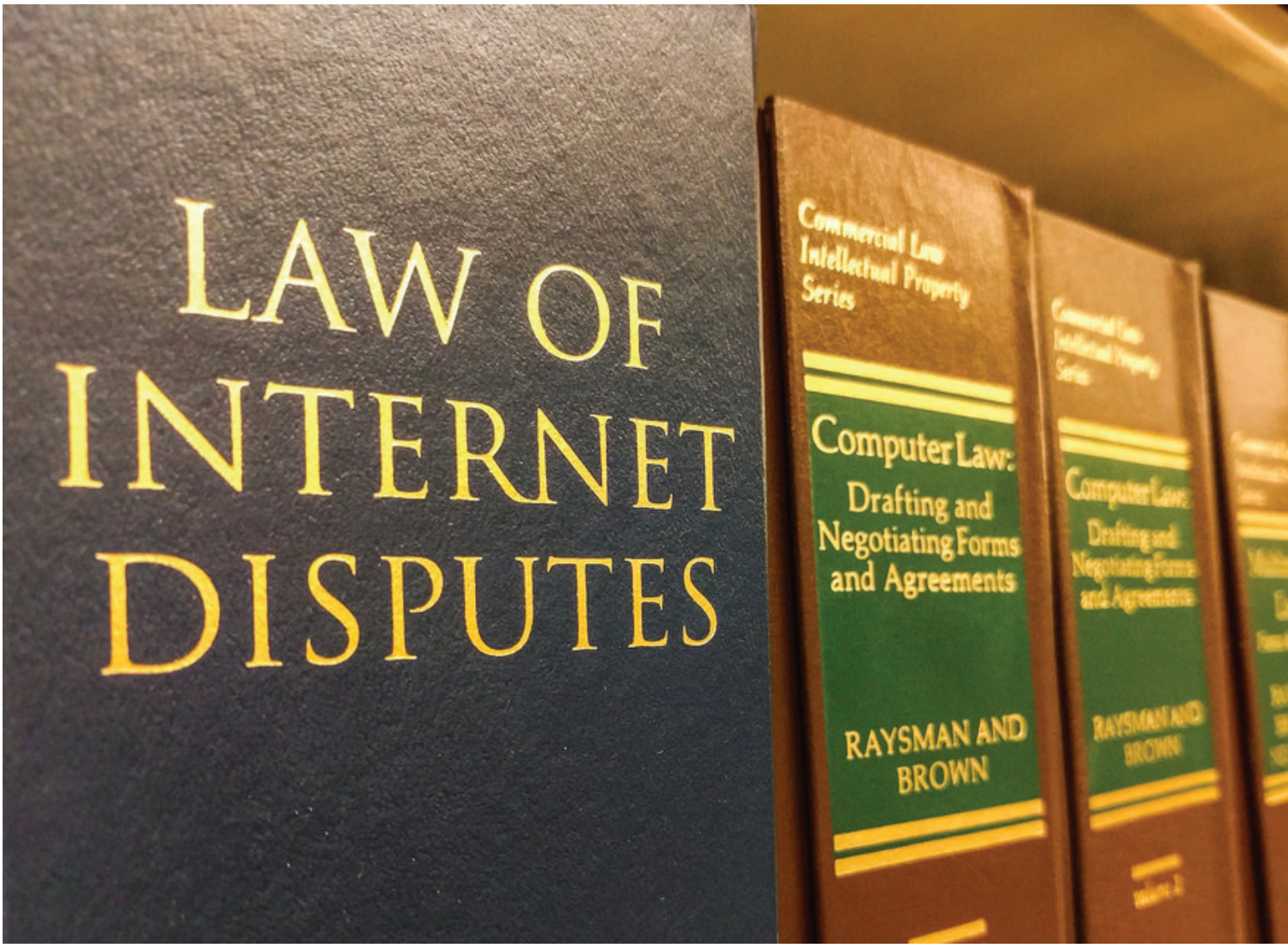
That supposition is now being challenged by Linux contributor Patrick McHardy in his case against Smart TV vendor, Geniatech. While McHardy's pattern of behavior since 2014 has been generally consistent with copyright profiteering based on his individual contributions, a German court is currently (Oct 2017) leaning toward ruling that McHardy can also claim Linux co-authorship, potentially opening Geniatech to a significantly large penalty as a result of their license infringement.

Another notable case is that of Korean developer Hancom, which is being sued for including open source software called GhostScript in one of their products without

abiding by the GPL license. This case has highlighted a key issue which has been an open question for at least a decade: whether an open source license represents a legal contract. While the issue has yet to be resolved, in April 2017 a US district court did "set the precedent that licenses like the GNU GPL can be treated like legal contracts, and developers can legitimately sue when those contracts are breached."

These kinds of precedence indicate that the open source universe may be becoming less collaborative and more litigious.

***An April 2017 US district court ruling on the question of whether an open source license constitutes a legal contract may soon be resolved in favor of open source developers, granting them the ability to sue for breach of contract when an OSS license is misused.***



# LICENSE TO CODE

## DOES DIY RISK MANAGEMENT WORK?

Surprisingly, even in this day and age of ubiquitous open source, there's still no single tool or strategy for ensuring compliance with open source licensing. While some enterprises continuing to do track licenses manually using spreadsheets, a number of Software Composition Analysis (SCA) tools are available to help you discover undisclosed open source software in your projects, and identify their associated licenses. However, the only way to ensure that the licenses have been properly implemented is to manually verify each one. As a result, to better manage legal risk, enterprises typically perform periodic license audits.

**1.** Run a SCA tool in order to identify the OSS included in each project, and their associated licenses.

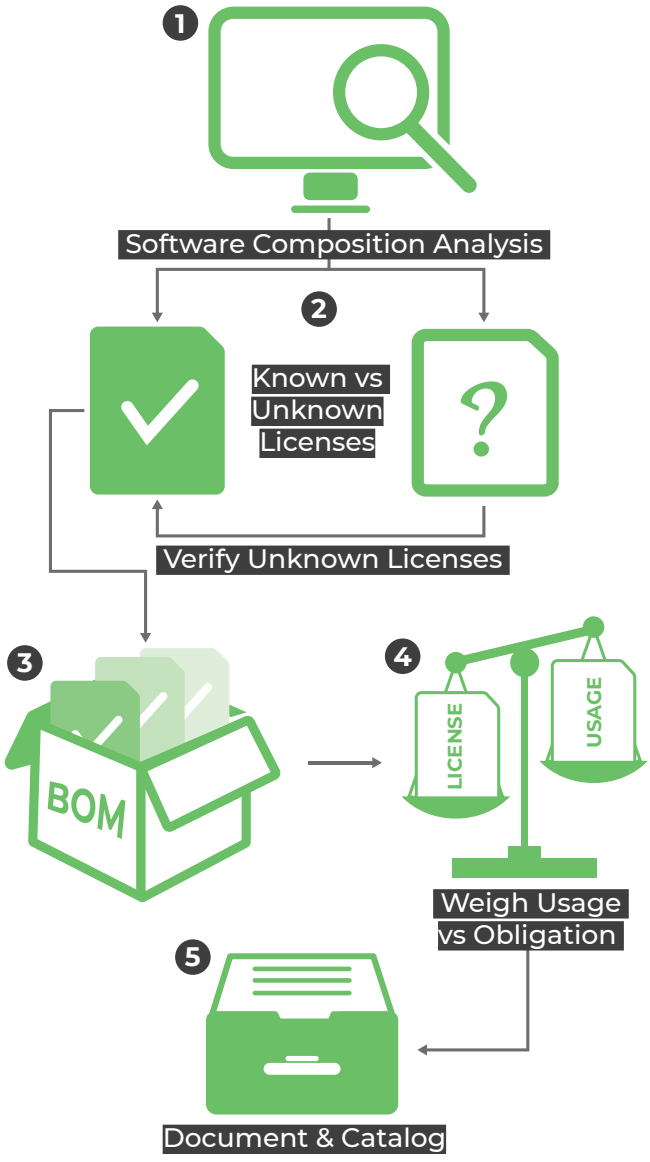
**2.** Manually investigate unknown or unidentified licenses.

**3.** Create an inventory or Bill of Materials (BOM) for all licenses in each project.
- **a.** Identify project owners and verify the resulting BOM with their expectations.

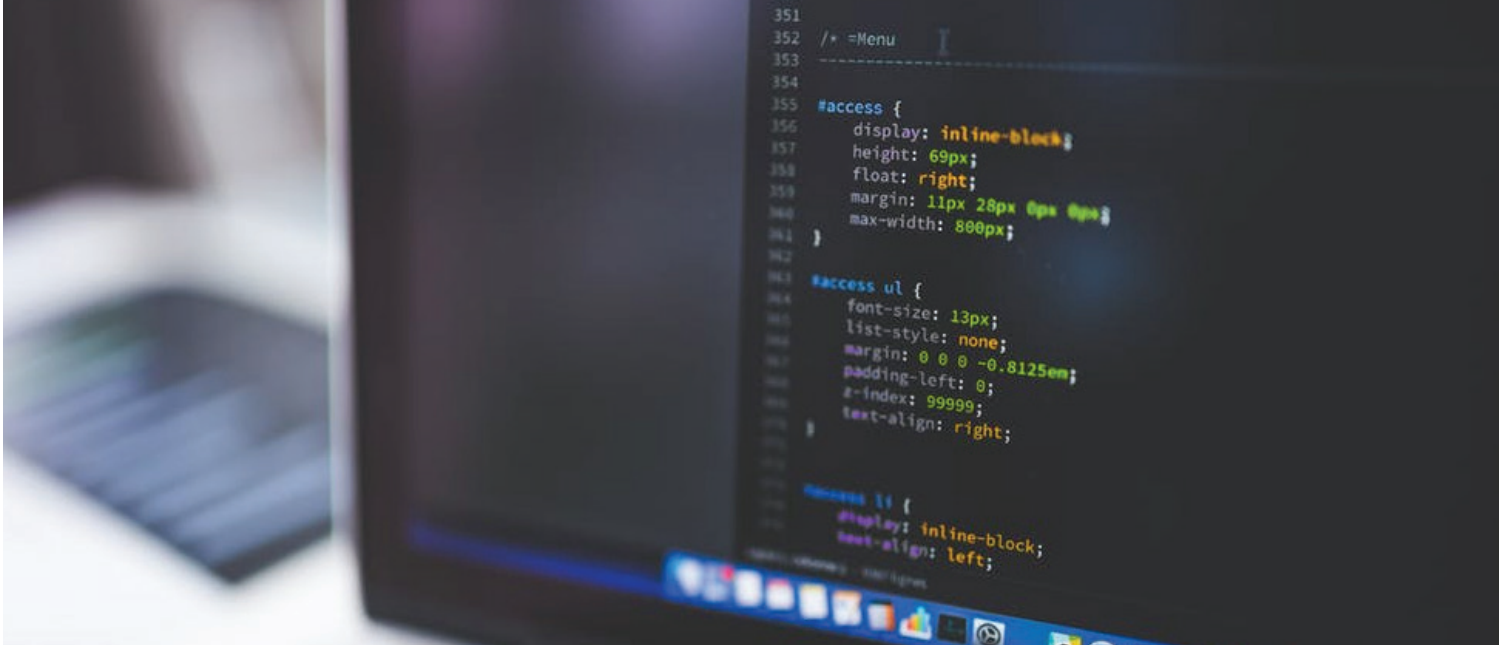
**4.** Ensure usage complies with license obligations:
- **a.** Will the code be used internally or will it be distributed?
- **b.** Has the code been modified?
- **c.** Can it be deployed with commercial custom code, or other (variously licensed) open source code?

**5.** Create or update an inventory of licenses and usage compliance.

Because open source audits can be resource and time intensive, many enterprises prefer to outsource it. But to ensure against legal action, you still have to be diligent, detail oriented and, ultimately, right when confirming the accuracy of the audit.




***According to Black Duck Software, as much as 95% of applications incorporate undisclosed open source.[4]***




## ENTERPRISE & OEM INDEMNIFICATION

The need for risk mitigation when it comes to open source litigation has created market demand for commercially backed open source that offers not only stable, comprehensive OSS distributions and SLA-backed support & maintenance, but also indemnification against copyright and breach of contract lawsuits.

***To ensure against legal action, you stil have to be diligent, detail oriented and, ultimately, right.***

For the past 20 years, ActiveState has been providing just that: enterprise-level, 100% open source-compatible 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 open source language distributions are provided free to the community.

With ActiveState Enterprise distributions you also gain indemnification against IP infringement lawsuits that will protect your company from legal exposure since your company is no longer responsible for licensing all the individual open source components of a language distribution. ActiveState Enterprise distributions and agreements override the open source licenses and offer the warranties, guarantees and indemnification large enterprises need, so you can deploy your code worry-free. For vendors that distribute applications and code externally to customers or partners, OEM licensing may be a better alternative to investing in periodic open source audits. ActiveState's licensing solutions provide instant, royalty-free distribution rights. Out-of-the-box licensing saves you time, resources and the headaches that accompany managing license compliance on your own.

***ActiveState Enterprise distributions and agreements override the open source licenses and offer the warranties, guarantees and indemnification large enterprises need.***

# LICENSE TO CODE


## SUMMARY

Building great software is hard enough without having to wrestle with open source licensing. Determining license compliance is time-consuming, risky and should not need to be a required core competency of your business. Plus, the legal risks associated with failing the compliance test can be significant.

ActiveState offers peace of mind for enterprises working with open source languages. ActivePerl, ActivePython, ActiveGo, ActiveRuby and ActiveTcl Enterprise distributions offer comprehensive licensing, indemnification and support & maintenance to mitigate the risks associated with licensing and distributing open source software with your products.


Contact ActiveState at **1.866.631.4581**, or **solutions@activestate.com** for a complimentary consultation with ActiveState's open source language experts.

***ActiveState offerings mitigate the risks associated with licensing and distributing open source software with your products.***


ActiveState®

website: www.activestate.com
Toll-free in NA: 1.866.631.4581
email: solutions@activestate.com




**ABOUT ACTIVESTATE**

ActiveState is a leader in providing commercial level open source language distributions. It provides commercial versions of Python, Tcl, Perl, Ruby and Go. More than two million developers and 97% of Fortune 1000 companies use ActiveState open source language builds including CA, Cisco, Pepsi, Lockheed Martin and NASA. To learn more, visit activestate.com