**Aim: OPEN SOURCE LICENSE**

**What is license?**

* The verb license or grant license means to give permission. Or license ([American English](https://en.wikipedia.org/wiki/American_and_British_English_spelling_differences)) refers to that permission as well as to the document recording that permission.
* A license may be granted by a party ("licensor") to another party ("licensee") as an element of an agreement between those parties. A shorthand definition of a license is "an authorization (by the licensor) to use the licensed material (by the licensee)."

**Popular Licenses**

The following OSI-approved licenses are popular that are widely used.

* Apache License 2.0
* BSD 3-clause New or Revised license
* BSD 2- clause simplified or free BSD license
* MIT License
* Mozilla Public License 2.0
* Eclipse Public License

**Apache License**

The Apache License (ASL) is a [permissive](https://en.wikipedia.org/wiki/Permissive_free_software_licence) [free software license](https://en.wikipedia.org/wiki/Free_software_license) written by the [Apache Software Foundation](https://en.wikipedia.org/wiki/Apache_Software_Foundation) (ASF). The Apache License requires preservation of the [copyright](https://en.wikipedia.org/wiki/Copyright) notice and [disclaimer](https://en.wikipedia.org/wiki/Disclaimer). Like other [free software licenses](https://en.wikipedia.org/wiki/Free_software_license), the license allows the user of the software the freedom to use the software for any purpose, to distribute it, to modify it.

**What Problem Does It Solve?**

Apache License 2.0.:work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as “Not a Contribution”.

**GNU General Public License**

The GNU General Public License (GNU GPL or GPL) is a widely used free software license, which guarantees end users the freedom to run, study, share and modify the software. The license was originally written by Richard Stallman of the Free Software Foundation (FSF) for the GNU Project, and grants the recipients of a computer program the rights of the Free Software Definition. The GPL is a copy left license, which means that derivative work can only be distributed under the same license terms. This is in distinction to permissive free software licenses, of which the BSD licenses and the MIT License are widely used examples.

**What Problem Does It Solve?**

Software under the GPL may be run for all purposes, including commercial purposes and even as a tool for creating proprietary software, for example when using GPL-licensed compilers. Users or companies who distribute GPL-

licensed works (e.g. software), may charge a fee for copies or give them free of charge. This distinguishes the GPL from shareware software licenses that allow copying for personal use but prohibit commercial distribution, or proprietary licenses where copying is prohibited by copyright law. The FSF argues that freedom-respecting free software should also not restrict commercial use and distribution (including redistribution): the GPL explicitly states that GPL works may be sold at any price.

**MIT License**

The MIT License is a permissive free software license originating at the Massachusetts Institute of Technology (MIT). As a permissive license, it puts only very limited restriction on reuse and has therefore excellent license compatibility. The MIT license permits reuse within proprietary software provided that all copies of the licensed software include a copy of the MIT License terms and the copyright notice. The MIT license is also compatible with many copy left licenses, such as the GNU General Public License (GPL);

**What Problem Does It Solve?**

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions: The above copyright notice and this permission notice shall be included in all

Copies or substantial portions of the Software.

**History**

In the early days of computing, programmers and developers shared software in order to learn from each other and evolve the field of computing. Eventually the open source notion moved to the way side of commercialization of software in the years 1970-1980. In 1997, [Eric Raymond](https://en.wikipedia.org/wiki/Eric_S._Raymond) published [The Cathedral and the Bazaar](https://en.wikipedia.org/wiki/The_Cathedral_and_the_Bazaar), a reflective analysis of the hacker community and free software principles. The paper received significant attention in early 1998, and was one factor in motivating [Netscape Communications Corporation](https://en.wikipedia.org/wiki/Netscape_Communications_Corporation) to release their popular [Netscape Communicator](https://en.wikipedia.org/wiki/Netscape_Communicator) Internet suite as [free software](https://en.wikipedia.org/wiki/Free_software). This source code subsequently became the basis behind [Sea Monkey](https://en.wikipedia.org/wiki/SeaMonkey), [Mozilla Firefox](https://en.wikipedia.org/wiki/Mozilla_Firefox), [Thunderbird](https://en.wikipedia.org/wiki/Mozilla_Thunderbird) and [KompoZer](https://en.wikipedia.org/wiki/KompoZer" \o "KompoZer).

## History of the Apache License

Beginning in 1995, the Apache Group (later the Apache Software Foundation) released successive versions of their well-known hatted server. Their initial license was essentially the same as the old BSD license, with only the names of the organizations changed. When Berkeley accepted the argument put to it by the Free Software Foundation and retired their advertising clause from the BSD license, Apache did likewise and created the Apache License v1.1 - a slight variation on the modified BSD license. In 2004 Apache decided to depart from the BSD model a little more radically, and produced the Apache License.

**History of the GPL License**

[The GPL (The GNU General Public License)](http://www.gnu.org/copyleft/gpl.html), created by [Richard Stallman](http://www.stallman.org/), serves as the de facto constitution for the Free Software movement. It covers the majority of Free Software/Open Source software and has become the legal and philosophical cornerstone of the Free Software community. Because of its fundamental nature, it is useful to document the historical evolution of the GPL.

First, the GPL comes from the philosophy of [Richard Stallman](http://www.stallman.org/) and [the GNU Project](http://www.gnu.org/). The general history of GNU is covered [here](http://www.gnu.org/gnu/thegnuproject.html). In this page we will be focusing on the specific events in the creation of the GPL.

In the early years (1984 to 1988), the GNU Project did not have a single license to cover all its software.  What led Stallman to the creation of this copy left license was his experience with [James Gosling](http://java.sun.com/people/jag/), creator of News and the [Java programming language](http://java.sun.com/), and [UniPress](http://www.unipress.com/), over Emacs. While Stallman created the first Emacs in 1975, Gosling wrote the first C-based Emacs(Gosling Emacs) running on Unix in 1982. Gosling initally allowed free distribution of the Gosling Emacs source code, which Stallman used in early 1985 in the first version (15.34) of [GNU Emacs](http://www.gnu.org/software/emacs/).  Gosling later sold rights to Gosling Emacs to UniPress, and Gosling Emacs became [UniPressEmacs](http://www.unipress.com/cat/emacs.html).   UniPress threatened Stallman to stop distributing the Gosling source code, and Stallman was forced to comply. He later replace these parts with his own code. (Emacs version 16.56).  (See the Emacs [Timeline](http://www.jwz.org/doc/emacs-timeline.html)) To prevent free code from being proprietarized in this manner in the future, Stallman invented the GPL. Detailed description of this event can be found in [Stallman's 1986 speech at the Royal Institute of Technology, Sweden:](http://www.gnu.org/philosophy/stallman-kth.html)

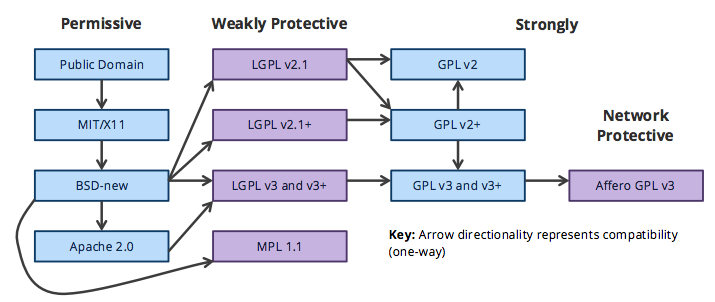
**History of the MIT License**

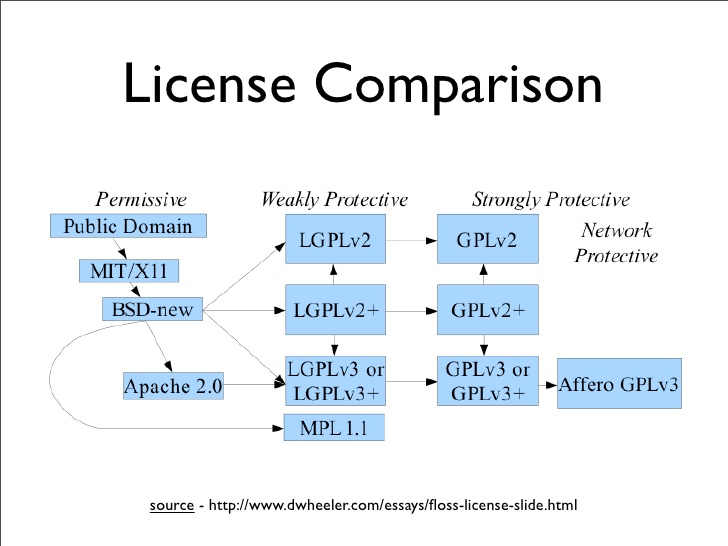
This course examines the history of MIT through the lens of the broader history of science and technology, and vice versa. The course covers the founding of MIT in 1861 and goes through the present, including such topics as William Barton Rogers, educational philosophy, biographies of MIT students and professors, intellectual and organizational development, the role of science, changing laboratories and practices, and MIT's relationship with Boston, the federal government, and industry. Assignments include short papers, presentations, and final paper. A number of classes are concurrent with the MIT150 Symposia.

**Open Source License Popularity**

*John T. Haller, Updated May 11, 2014 (Created August 28, 2009)*

Popularity of individual open source license usage is tracked by several entities with the results published online. I thought it would be interesting to combine the license types that are similar and map out the overall popularity along with OSI and FSF approval and GPL compatibility. As you can see, the GPL/LGPL is the most popular license by a wide margin with over 45% of software being licensed under one or more versions. The next 3 licenses are all GPL compatible in some way meaning 67% to 87% or more of all open source code is GPL compatible and can be added to GPL licensed projects. Note that you can't tell exactly as some version issues enter into the mix with the Apache and Perl licenses.

**DETAILED LICENSING MODEL**



# Impact of License

# Apache License

# Why we chose the Apache License

https://secure.gravatar.com/avatar/4cb777496236a3f8b86c2fb53b98bc44?s=40&d=mm&r=g By [Adam Jacob](https://blog.chef.io/author/adam/)[August 11, 2009January 11, 2011](https://blog.chef.io/2009/08/11/why-we-chose-the-apache-license/)

From time to time, we get asked why new contributors to any Opscode project have to sign a Contributor License Agreement (or a Corporate Contributor License Agreement.) While there is an explanation on the [Opscode Wiki](http://wiki.opscode.com/display/chef/How+to+Contribute), we’ve never gone into deep detail on how we made the choice of the Apache License, nor why we follow the Apache Foundation’s practice of requiring contributors to sign Contributor License Agreements (CLAs) (and Corporations to sign Corporate CLAs). So if you geek out about open source licensing, and the intersection between business requirements and license choices, read on.

A quick disclaimer – I am not a lawyer, and this post is intended to make sure we’re as clear as we can be with our community, friends, and partners about why we chose the Apache License. You should choose the license that is right for your project.

# Our Requirements

In a nutshell, we had three broad requirements for the license we released our open source software under:

* We are an Open Source business – meaning we want to make money from our Open Source software. We wanted a license that allowed us to build a business from our creations.
* We wanted anyone (or any company) whose problems were solved by our software to be able to use it, in any environment they wanted, in what ever way they wanted.
* We wanted to build an open and equal community – we didn’t want to reserve any rights for ourselves that we didn’t grant to the other people (or companies) that help build the software.

# Our Business

Really, every mainstream Open Source license will allow you to make money from your software. The real differentiation comes when you talk about how you make that money, and whether you allow others to make money in the same way. For example, a common practice in many commercial open source businesses is to license their software under a strong copy left (GPLV2, GPLv3 or AGPLv3) and require copyright assignment from contributors. The important part here is the copyright assignment: this allows the company to be the sole copyright holder for the work, which provides them the benefit of being able to re-license the work as they see fit. Companies using this model include [MySQL](http://www.mysql.com/about/legal/licensing/index.html) (GPLv2 w/FOSS exception, commercial license), [Funambol](http://www.funambol.com/solutions/licensing.php) (AGPLv3, commercial license), and [Hyperic](http://www.hyperic.com/partners/faq.html).

The intent of this model is clear: as the creators/originators of their respective products, each of those companies uses the copy left to ensure that the playing field is level for those contributing back to the project. For those who wish to be free from those restrictions, they can pay a licensing fee, and then can do with the software what they will. It’s a fine model, and I respect many companies who follow it.

For us, though, this model comes into conflict with our second and third requirements: that anyone whose problems are solved by our software can use it, and that we not reserve any rights for ourselves that we don’t grant to other people (or companies) who help us build the software. Companies like [Engine Yard](http://www.engineyard.com/) and [RightScale](http://www.rightscale.com/) already have commercial offerings around Chef – they did so without needing to ask our permission, and with a high degree of confidence around the contents of the Chef code base. They are both contributing back to the project – not because they are bound to by the license, but because the eco-system encourages them to do it. They aren’t just helping Opscode – they are helping each other, and anyone else who uses Chef.

Which brings us to how we feel about the relationship of our license in terms of helping us make money on our free software. The license sets up the level playing field upon which a truly fertile community can grow – of individual contributors and corporations. We believe that we can build a business that works for us, and our investors, through building true and lasting partnerships based on equality and mutual purpose. Our decision on the Apache License reflects that choice.

# GPL LICENSE

# The Impact of the GPL

So what effect has the GPL had on software development and the open source movement as a whole? It's actually had a tremendous impact.

First, many very common UNIX applications, such as GNU Emacs, have been released under the GPL, and are used by countless numbers of users every day.

Second, the open source software movement has taken several ideas promoted by the GPL and modified them slightly. The most important is the idea that software licensing should include access to source code. As we move into a more complex era of computing, this issue becomes important for multiple reasons:

* Stability and longevity. If your company invests in a proprietary software package for a mission-critical task, and the company that sold you the software subsequently goes out of business, what do you do? It might mean scrapping a large-scale deployment, which can be very costly. However, with source code, you could have internal company developers charged with maintaining the application, fixing bugs, and even developing new features.
* Security. By being able to review the source code, you can be certain that a mission-critical application is secure and doesn't contain any backdoors or other potentially devastating security flaws.

These are just a couple of reasons that have caused many people to advocate the inclusion of source code in software distribution.

However, some developers want to distribute source code, but don't want to forgo all distribution rights, as would be required under the GPL. As a result, there have been a host of competing open source licenses: the Apache License, the BSD License, the IBM Public License, the Sun Public License, and the QtPublic License, to name just a few. The [Open Source Initiative](http://www.opensource.org/licenses/) (OSI) reviews licenses for compliance with the goals of open source, and publishes approved licenses on the OSI web site.

There is even an effort to create open source or copy left-style licenses for creative content, such as writing, video, Flash animations, etc. This effort is spearheaded by the [Creative Commons](http://www.creativecommons.org/), which has a number of different licensing options available to the creative community.

The bottom line is that the GPL and licensing agreements from the open source community aren't a panacea for protecting individual liberties, and they're not an affront to the free market and the protection of intellectual property. They're simply another tool that developers have in their arsenal to control how they would like the software they create to be used.

# MIT LICENSE

# COMMUNITY IMPACT

## Community Collaboration

Since its founding, MIT has maintained a commitment to serving the local community as a good neighbor, as a resource for education and technology, and by contributing to the state and local economy.

 Some examples of how MIT strives to support its host community include offering expertise to nonprofits and programs via committee and board memberships; donation of funds; educational initiatives; facility use; and community service performed by students, faculty, and staff. The [MIT's Collaborations in Cambridge document](http://ogcr.mit.edu/sites/default/files/MITs%20Collaborations.pdf) outlines the Institute's efforts in this regard.

**Economic Impact**

The Institute, which employs nearly 10,000 individuals, and is the second largest employer in Cambridge, contributes significantly to the local and state economy, serving as a magnet for investment and fueling an innovation economy with the research, startups, and talent pool it generates. According to the 2009 [Entrepreneurial Impact: the Role of MIT](https://entrepreneurship.mit.edu/about/entrepreneurial-impact-report/) report by the Ewing Marion Kauffman Foundation, over 200 Cambridge companies have strong ties with the MIT community. The report also cites that many of these Cambridge startups have licensed technology from the [MIT Technology Licensing Office](http://web.mit.edu/tlo/www).

In Cambridge, MIT pays taxes on its commercial property and provides an annual payment in lieu of taxes (PILOT) for property that is legally tax exempt. In fiscal year 2012, the Institute continued this 83-year old tradition by providing the City of Cambridge with a voluntary PILOT contribution in the amount of $2.3 million. In addition, MIT paid or generated over $36.5 million in real estate taxes in fiscal year 2012, and in doing so represented 12.2 percent of the Cambridge tax revenue stream and was the City's largest taxpayer.

MIT also supports the financial base of Cambridge through other means, including payments for licenses, fees, and permits, totaling $7.2 million in fiscal year 2012. MIT is proud of its Cambridge First Purchasing Program, which resulted in the investment of over $58.4 million in Cambridge businesses in fiscal year 2012. This program, together with taxes paid, payments in lieu of taxes, and municipal fees, brought MIT's fiscal year 2012 economic contribution to the City to over $104.5 million.