

How do you choose an open-source license? [closed]

Asked 16 years, 3 months ago Modified 15 years, 6 months ago

Viewed 4k times



26



Closed. This question is [off-topic](#). It is not currently accepting answers.



Want to improve this question? [Update the question](#) so it's [on-topic](#) for Stack Overflow.

Closed 11 years ago.

[Improve this question](#)

I'm a software engineer, not a lawyer, and my university doesn't offer any courses geared toward licensing software. In fact, their law-related courses are lacking (but slowly growing in number). Where can I go to learn about open-source licenses and how to choose them?

open-source

licensing

Share

Improve this question

Follow

edited Sep 10, 2008 at 5:13



Community Bot

1 • 1

asked Sep 3, 2008 at 15:12



Thomas Owens

116k ● 99 ● 317 ● 436

10 Answers

Sorted by:

Highest score (default)



17

There are lots described here:

<http://www.gnu.org/licenses/license-list.html#SoftwareLicenses>



The decision of which one to use can be political, but should ultimately be determined by your plans/desires for the software. If you want to ensure it is always free then choose GPL or another "Copyleft" license. If you don't mind some commercial use, choose another one that's compatible with that.

Share Improve this answer

answered Sep 3, 2008 at 15:16

Follow



Jason Diller

3,368 ● 2 ● 26 ● 25

-
- 2 Watch out for the two meanings of "free" in this context; you CAN sell GPL'd software, and it can even be used in a commercial context. See gnu.org/philosophy/selling.html – Neil Williams May 28, 2009 at 20:53
-



16

I almost always end up usign MIT or BSD (they're equivalent), since it



1. Is the most liberal license out there. It just says you're not responsible for any kind of trouble, and optionally forces people to include a copyright notice of your original work in derivatives.
2. It allows closed source derivatives, which is something I see as a good thing: companies sometimes don't have the possibility to do their work under the GPL (they may themselves use products or components from a third party with restricted licenses).

That, and the GNU/GPL bunch are generally extremists when you encounter them in the wild.

Share Improve this answer

answered Sep 3, 2008 at 15:20

Follow



[wvdschel](#)

11.8k ● 14 ● 42 ● 45

-
- 6 The Boost license is even more liberal than MIT/BSD, as it doesn't require copyright notice in distributed binary or library code. – [Setjmp](#) Mar 17, 2010 at 20:17
-



This can create endless discussion, but there is one tenet I would hold to whenever deciding what license to use:

12

DON'T CREATE A NEW ONE!!



No matter how persuasive your legal guy's arguments that, because no current license exactly meets your project's unique needs, you should write your own, or even just "slightly modify" an existing one, treat him like a programmer coming to you arguing that he just **HAS** to use a GOTO statement because nothing else in the language will work.



Other advice:

- Choose one which has major usage (see <http://freshmeat.net/stats/#license>)
- See David A. Wheeler's discussion of why to choose a license compatible with the GPL - <http://www.dwheeler.com/essays/gpl-compatible.html>.

Share Improve this answer

answered Sep 16, 2008 at 1:50

Follow



Perry

226 ● 1 ● 3

2 +100 (if I could) for the don't create a new one advice.
– [jmucchiello](#) May 28, 2009 at 20:52



If you are looking for information regarding free and open source licenses a useful comparison chart:

8

http://en.wikipedia.org/wiki/Comparison_of_free_software_licences



Share Improve this answer

answered Sep 3, 2008 at 15:18



Follow



[jwarzech](#)

6,665 ● 11 ● 53 ● 74



5

You could always just use the best one of all, the [WTFPL](#). I use this on most of my school projects since they aren't that great anyways.



Share Improve this answer

answered Sep 3, 2008 at 18:11



Follow



[Brian Paden](#)

1,351 ● 4 ● 15 ● 25



I personally go for the Poetic license, but this is also a good one. – [Chris Lutz](#) May 28, 2009 at 20:54

1 +1 There's nothing as simple and clear as the WTFPL.
– [Xiong Chiamiov](#) Aug 26, 2009 at 22:22



3

Wikipedia, of course, has basically all the information you would ever need to know. But the hard part is to know where to start. I'd recommend starting off by reading about the [Apache License](#) and the [GNU GPL](#), which are two popular sides to the same story, each offering different freedoms to the people associated with the code.





But here it is in a nutshell: Apache License lets anyone do anything with your code, including taking it and using it in a closed source product. It gives whoever is taking the code the **freedom** to do what they want with it.

The GNU GPL, on the other hand, allows your code *only* to be used in a project that is also distributed under the GPL. In this case you might write some code and prevent a proprietary company from using your work. Here, you're giving **freedom** to the code itself that it will always be used for "free" purposes.

Share Improve this answer
Follow

answered Sep 3, 2008 at 15:20



[Will Robertson](#)

64.4k ● 33 ● 102 ● 117



3



I'm slightly surprised to see no mention of the [Open Source Initiative](#) as a source of information about which open source licences exist. It probably doesn't do the comparisons, so the other sites are also worth checking.

Share Improve this answer
Follow

answered Sep 29, 2008 at 17:52



[Jonathan Leffler](#)

752k ● 145 ● 946 ● 1.3k



1

More pragmatic reasons can also influence your choice of license - if you want to use a GPL library, you must use GPL yourself, or if you intend your software to be part of a larger project then you need to look at their requirements.



Share Improve this answer

answered Sep 3, 2008 at 15:21

Follow



Daniel James

3,939 ● 23 ● 30



1



I've recently begun investigating the type of licensing to apply to a rather substantial piece of work. The number of choices and the content, restrictions (or not) and limitations of all the open-source licenses is bewildering.

I've found a couple good links in the answers posted, but I didn't see anything pointing to the [Open Source Initiative's alphabetical list of licenses](#), so I've included it here.



Share Improve this answer

answered May 28, 2009 at 20:49

Follow



Gary Chambers

1,267 ● 1 ● 10 ● 10



0



We had a similar dilemma. At our company we decided invest lots of time on a framework, with the eventual hope of releasing it to the open source community. The business is built using open source tools (apache, php, etc.), it was time to give back. We decided on an LGPL/MPL dual license. That way, we could incorporate fixes/improvements from the community, while still protecting applications (particularly ours) running on top of it from being forced to go open source as well.



Share Improve this answer

answered Sep 3, 2008 at 15:20

Follow



John O
