



Tech
Law
Garden

Free and Open Source Software

Overview and Best Practices

© 2014 Tennille Christensen, all rights reserved.

First, the Copyright

- In the US and under most countries' laws, copyright vests in the creator of a work of original authorship immediately upon creation of the work.
- Today, most software is originally authored in the form of source code.
- A file full of source code contains an expression of an idea of how to tell a computer to do something written in the programming language selected by the author.
- This is analogous to a poem about a particular topic (e.g. Dickinson, Shelly, Neruda, Bronte, etc. on Death)

Exclusive Copyright Rights

17 USC § 106

Subject to sections [107](#) through [122](#), the owner of copyright under this title has the exclusive rights to do and to authorize any of the following:

- (1) to reproduce the copyrighted work in copies or phonorecords;
- (2) to prepare derivative works based upon the copyrighted work;
- (3) to distribute copies or phonorecords of the copyrighted work to the public by sale or other transfer of ownership, or by rental, lease, or lending;
- (4) in the case of literary, musical, dramatic, and choreographic works, pantomimes, and motion pictures and other audiovisual works, to perform the copyrighted work publicly;
- (5) in the case of literary, musical, dramatic, and choreographic works, pantomimes, and pictorial, graphic, or sculptural works, including the individual images of a motion picture or other audiovisual work, to display the copyrighted work publicly; and
- (6) in the case of sound recordings, to perform the copyrighted work publicly by means of a digital audio transmission.

Source Code Is Essentially Useless Without a Copyright License

Four types of software (this is not *exactly* correct)

- Assembly (machine code instructions that are tightly tied to the hardware where it is executed, assembled into machine code Ex: X86, PowerPC)
- Executable Code (compiled from source code into either assembly (which is compiled into machine code) or machine code by a compiler. Ex: C, C++)
- Byte code (compiled from source code into Byte Code, which is interpreted by the language's virtual machine into processor instructions at runtime. Ex: Java, Ruby, Python)
- Script code (human readable source that is interpreted and executed by a runtime engine directly from the script source at runtime. Ex: Javascript, HTML, PHP, Perl)

All human readable software files require the computer to make **copies** and **derivative works** in order for the processor to run the software. Client-server software (web/mobile-based applications) also requires that software be **distributed** to the client in order for the system to run it.

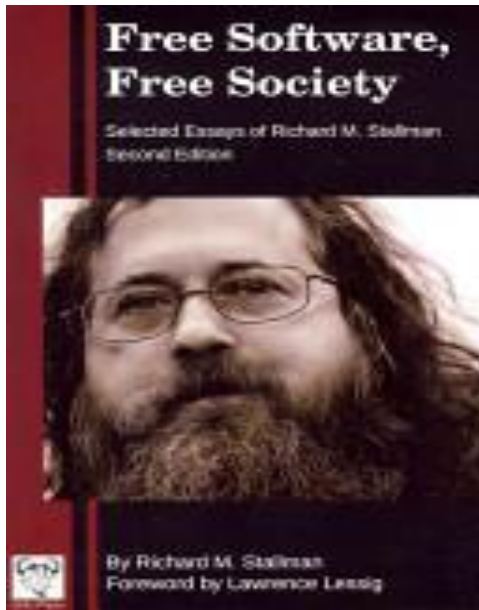
In the Beginning They Shared

- Without a copyright license to software, you do not have any legal right to use the software.
- In the 1950s and 1960s, however, most computer business models were based around hardware, and software wasn't seen as needing protection. It was often freely shared, used, and modified, and copyright rights were not enforced.
- However, as time marched on, licensing, maintenance and support of software became a more viable business model.

Then Things Changed

- Proprietary software became the norm
- Companies started distributing software in compiled or executable form only.
- Without the source code, it was much more difficult for users to review and modify software.
- In 1980, the US Congress enacted Section 117 of the US copyright act, making it **very** clear that computer programs were protected works under the Act. (See also: Don't Copy That Floppy, <http://www.youtube.com/watch?v=up863eQKGUI> 1992)
- Under the Bern Convention, WIPO and TRIPS, almost all nations recognize that software is copyrightable.

RMS Got Grumpy



- Many in the academic software world and the computer hobbyist world felt that software should be “free”.
- In 1983, Richard M. Stallman, a longtime member of the hacker community and a professor at MIT in the Artificial Intelligence Laboratory announced the GNU Project.



by
Richard Stallman

The GNU Manifesto

- Free Software Definition:

Free as in speech, not free as in beer.

- Copyleft:

GNU is not in the public domain. Everyone will be permitted to modify and redistribute GNU, but no distributor will be allowed to restrict its further redistribution. That is to say, proprietary modifications will not be allowed. I want to make sure that all versions of GNU remain free.

- Founded the Free Software Foundation (FSF)

Copyleft Freedoms

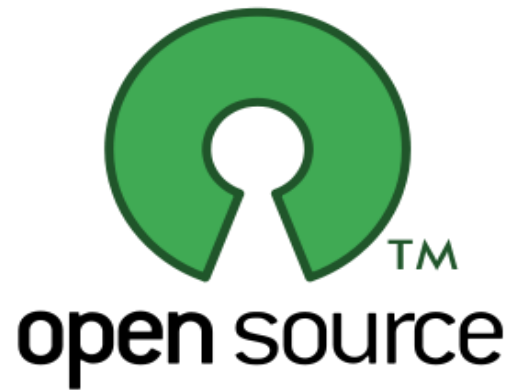
- Freedom 0 – the freedom to use the work,
- Freedom 1 – the freedom to study the work,
- Freedom 2 – the freedom to copy and share the work with others,
- Freedom 3 – the freedom to modify the work, and the freedom to distribute modified and therefore derivative works.

What's Wrong With Free As In Beer?

In the mid 1980s, some in the hacker, hobbyist, and academic software communities propagated permissive licenses on the software they wrote:

- BSD licenses (4 clause, 2 clause, FreeBSD)
- MIT license and derivatives thereof

AS IS, ALL WARRANTIES DISCLAIMED, USE AT YOUR OWN RISK



The Open Source Initiative

- Founded in February 1998, prompted by Netscape publishing the source code for Netscape Communicator.
- Chose the term open source to "dump the moralizing and confrontational attitude that had been associated with 'free software' in the past and sell the idea strictly on the same pragmatic, business-case grounds that had motivated Netscape."

The Free and Open Source Software World Today

Strong Copyleft

AGPL (use), GPL (distribution), ShareAlike, Sleepycat, etc.

Weak Copyleft

LGPL, MPL, EPL, etc.

Permissive

(often with some additional minor restrictions)

APL, MIT, BSD, etc.

Commercial Best Practices

FOSS (and third party software) Policy

- Copyleft position (may vary based on where/how the software will be used)
- Default rules on use of software under popular licenses
- Does the Company allow employees to contribute to FOSS projects?

FOSS (and third party software) Process

- Steps before evaluation/use
- Steps before check-in of FOSS/3rd party software into the company's source code tree
- Steps before public release or distribution of software containing or linking to FOSS
- Steps before contributing FOSS to the public

Engineers Generally Don't Understand All Of the Legal Requirements

- Very common for an engineer to say, “This software is Free. Or Open Source.”
- Less common for the software to be as free as the engineer thinks.
- Even the Free As In Beer Licenses have attribution and disclaimer requirements.
- Software release teams are often shocked that they have to compile a list of all open source and make the list, copyright holders, and licenses available in connection with each release.

There Are Many Licenses

Don't underestimate creative copyright holders:

- Hidden copyleft or commercial restrictions in license formats that visually resemble popular permissive licenses
- This software shall be used for Good, not Evil.
- The Do What the Fuck You Want Public License
- The Rude License (BSD-style, but modified to be full of curse words – very effective to help understand the attribution requirements that are inherent in BSD-style licenses).

(The next slide contains vulgar language)

The Fucking License

Copyright (c) 2012 [YOUR NAME HERE]. All rights reserved. So don't fuck with me.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- If you redistribute the source code, it must include a copy of this fucking license. You can't fucking remove this license from the source. It fucking stays or I will fucking hurt you.
- If you redistribute in binary form, the same goes. Put a fucking copy of this license in the fucking documentation or something. People need to read this shit.
- And no, you fucking well can't use the name of the author to endorse or promote your shitty software or products. Seriously, why the fuck would you do that anyway?

THIS SOFTWARE IS PROVIDED "AS FUCKING IS". OH, YOU WANT A WARRANTY? OH, YOU THINK JUST BY GETTING HOLD OF THIS SHIT IMPLIES A WARRANTY, MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE? YOU THINK YOUR FUCKING LAME "SAAS" WEBSITE IS GOING TO MAKE YOU A MILLIONAIRE, AND YOU WANT TO BLAME ME IF SHIT GOES WHACK? WELL FUCK YOU! IT FUCKING WELL DOES NOT! I TOTALLY DISCLAIM ANY OF THAT SHIT.

LISTEN HARD: THERE IS NO FUCKING WAY THAT I WILL BE HELD LIABLE FOR ANY OF THE CRAP YOU WANT TO TRY AND PIN ON ME:

1. I DISCLAIM DAMAGES OF ANY TYPE - DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL. WHO THE FUCK KNEW THAT THERE WERE SO MANY TYPES OF DAMAGES? AND YET HERE I AM DISCLAIMING THEM ALL.
2. IF YOU USE THIS SOFTWARE AND END UP HAVING TO BUY NEW SHIT, OR HIRE SOME ACNE-RIDDEN NERD TO FIX STUFF THAT WENT WRONG, OR IF SHIT GETS FUCKED UP SO BAD YOU HAVE DOWNTIME, WIPE YOUR DATA, LOSE PROFITS, OR SUDDENLY CAN'T ACCESS REDDIT.COM, I DON'T FUCKING CARE AND IT'S NOT MY FAULT. CONSIDER IT ALL DISCLAIMED, MUTHAFUCKER.
3. I DON'T GIVE A SHIT IF YOU HAVE SOME FUCKING "THEORY" OF LIABILITY. YOUR THEORY OF LIABILITY IS NOT GOING TO SAVE YOUR ASS. YOU KNOW WHY? BECAUSE I'M FUCKING DISCLAIMING LIABILITY RIGHT HERE; RIGHT NOW. THAT INCLUDES CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) AND THAT APPLIES TO ALL THE DAMAGES I MENTIONED ABOVE. EVEN IF I FUCKING TOLD YOU. EVEN IF I SAID "HEY, IF YOU USE THIS SOFTWARE, YOUR COMPUTER WILL BLOW UP". WHICH IT MIGHT.

THE FUCKING END



Tech Law Garden

Questions?

Please feel free to reach out to me directly at
[tennille\[AT\]techlawgarden.com](mailto:tennille@techlawgarden.com)