



Selecting a License for Your Open Source Project

John Brockland

Partner

Cooley Godward Kronish LLP

March 25, 2008

Overview

- Introduction
- Key Provisions of Popular Licenses
- When to Consider Creating Your Own
- Considerations for Dual Licensing
- Copyleft and its Limitations
- Applying License Terms to Your Code

Introduction

- There are many choices (OSI lists over 70)
- But just a few key “moving parts”
- And just a few “most popular”

Introduction

- Key considerations
 - Motivation/Strategy for releasing open source
 - Strength of copyleft
 - Strength of attribution
 - Patent terms
 - Popularity
 - Compatibility with licenses for 3rd party code
 - Clarity and completeness of drafting

Key Provisions of Some Popular Licenses

- GPL (and family)
 - Copyleft: strong (works “based on” the Program)
 - AGPL: extends copyleft to network use
 - LGPL: permits “linking” without triggering copyleft
 - Attribution: preserve “appropriate copyright notice” and interactive “Appropriate Legal Notices”
 - Compatibility: generally low (but see GPLv3 Sections 7 and 13)
 - GPLv3: patent license and patent retaliation clause
 - Very popular

Key Provisions of Some Popular Licenses

- Apache 2.0
 - Copyleft: none
 - Attribution: retain attribution notices in source form, and in notice.txt, derivative source, or user interface
 - Compatibility: broad compatibility
 - Patent license and patent retaliation clause
 - Very popular

Key Provisions of Some Popular Licenses

- CDDL
 - Copyleft: limited (modifications)
 - Attribution: retain any copyright, patent or trademark notices contained within the software, and “any notices of licensing or any descriptive text giving attribution”
 - Compatibility: copyleft term enhances compatibility
 - Patent license and patent retaliation clause
 - Popular
 - Based on, and very similar to, Mozilla Public License

Key Provisions of Some Popular Licenses

- Eclipse Public License
 - Copyleft: limited (changes and additions, excluding new modules that are not derivative works)
 - Attribution: no removal or alteration of copyright notices
 - Compatibility: copyleft and binary distribution terms enhance compatibility
 - Patent license and patent retaliation clause
 - Popular

Create your own?

- Generally, No
 - Lose benefits of popularity
 - Waste time explaining
 - Impose external costs
- (Unless you have a good reason)
 - Tailor to specific technology (e.g., hardware)
 - Tailor to specific community
 - Customize existing license

Create your own?

- Some customization options
 - Use GPL Section 7
 - Add exceptions for compatibility
 - Attribution provisions (e.g., Terracotta)

License Compatibility

- When choosing a license, consider licenses for 3rd party code in the project
- FSF says: “In order to combine two programs (or substantial parts of them) into a larger work, you need to have permission to use both programs in this way. If the two programs' licenses permit this, they are compatible. If there is no way to satisfy both licenses at once, they are incompatible.”
- For the GPL, this means that the other license must permit derivatives to be licensed under the GPL.
- Some licenses have more lenient requirements
 - Eclipse Public License allows separately licensed components at the plug-in level
 - Apache license has no copyleft
- Analysis is much the same as when combining open source and proprietary code
- FLOSS exception can save licensees headaches

Dual Licensing

- Licensing software under both open source and proprietary terms
- Sufficient IP rights (compatibility, some exclusive rights)
- Reason for licensee to take commercial license
 - Avoid copyleft
 - Enhanced features
- Consider copyleft
 - Strong copyleft
 - No copyleft to speed adoption

Copyleft and Its Limitations

- Open source companies rely on copyleft
 - To carve out competitive niche
 - To promote dual licensing model
- Copyleft has limitations
 - Based on copyright law—strong but narrow
 - Lose trade secrets and contractual limitations that protect proprietary code
 - Generally requires distribution (but see AGPL)
 - Derivative work concept is not well defined for software
 - Litigation is expensive
 - *Jacobsen v. Katzer?*

GPL v2 or v3?

- GPLv3
 - Clears up some ambiguities
 - Allows some flexibility
 - Gaining popularity
 - Patent license
 - Contains surplus language
- GPLv2
 - Many years worth of “lore”
 - Linux

Applying Terms to Your Code

- Due diligence
 - Ownership
 - Sufficient license rights?
 - Conflicting agreements?
- Any code to remain proprietary?
 - Dual license
 - Open source is a one-way street

Applying Terms to Your Code

- Mechanics
 - Some licenses (e.g., GPL) have instructions
 - Most are based on “bare license” model
 - Notices
 - In source
 - License.txt
 - Download page
 - Third party notices
 - Disclaimers
 - Exceptions

Questions?

John Brockland
Cooley Godward Kronish LLP
jbrockland@cooley.com
650.843.5476

attorney advertisement