

Case studies

Legal Aspects – Master on Free Software 2012-13

Miguel Vidal

<http://flossystems.com>

Twitter: @mvidallopez

November 16th, 2012



(cc) 2008-2012 Miguel Vidal

This work is licensed under
a Creative Commons Attribution 3.0 License



<http://creativecommons.org/licenses/by/3.0>

- THIS PRESENTATION IS FOR INFORMATIONAL AND EDUCATIONAL PURPOSES ONLY, AND IS NOT LEGAL ADVICE!
- IT MUST NOT BE RELIED UPON AS A SUBSTITUTE FOR OBTAINING SPECIFIC LEGAL ADVICE FROM A LICENSED ATTORNEY.
- LEGAL ADVICE MUST BE PROVIDED IN THE COURSE OF AN ATTORNEY-CLIENT RELATIONSHIP SPECIFICALLY WITH REFERENCE TO ALL THE FACTS OF A PARTICULAR SITUATION AND THE LAW OF YOUR JURISDICTION.

Hire an attorney if you need legal advice

- Lesson 0: Presentation of the Course
- Lesson 1: Intellectual Property: basic concepts and legal framework
- Lesson 2: Legal Aspects of Free/Open Source Software
- Lesson 3: Free/Open Source software licenses
- Lesson 4: Free licenses for other intellectual works
- Lesson 5: Case studies

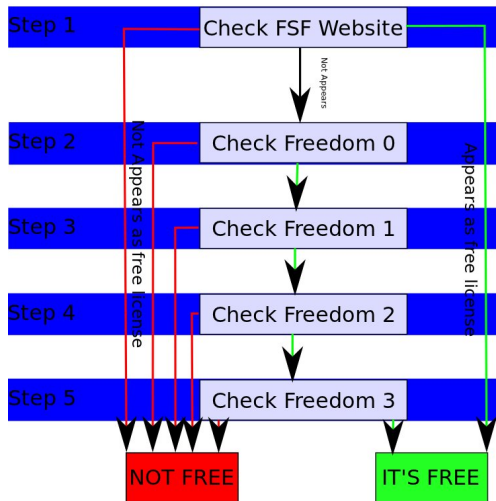
Choosing a free license: previous criteria

- Each project has its own goals and criteria related to licensing issues.
- The **main criterion** of differentiation is existence (or not) of **reciprocity pacts**.
- Other copyleft licenses have a limited effect, which applies only to the work (or component) original (“weak copyleft”).
- **Dual licensing** policies.

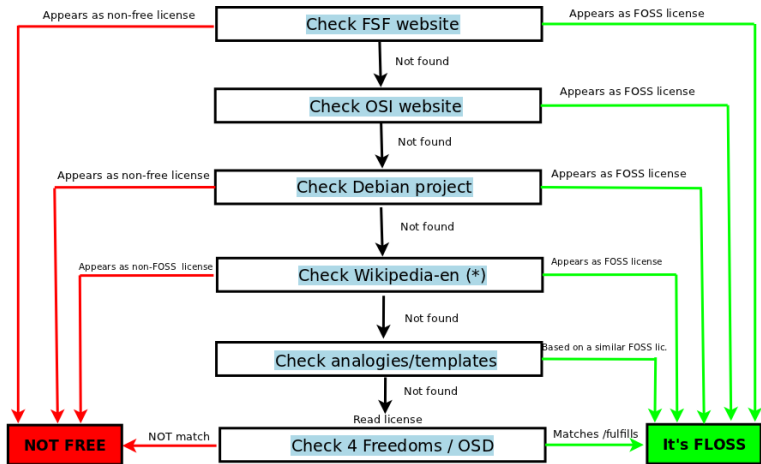
Choosing a free license: When?

- When we want to guarantee some basic freedoms, common to all free software.
- When we want a work achieves the highest use and dissemination (permissive licenses).
- When we want to maintain control over the evolution of the program (copyleft licenses).
- When we impose certain conditions or restrictions (the recognition of authorship, copyleft, lack of liability, extra warranties, trademarks, etc.)

How to know when a software license is free?



How to know when a software license is free?



[*] Warning: Wikipedia is NOT a primary source! (so you MUST check verifiable references in the wikiarticle)

(c) Copyright 2011 Miguel Vidal
Under CC-by 3.0 license

Choosing a free license: Cases

- Do I want to allow privatization of derivative works?

Choosing a free license: Cases

- Do I want to allow privatization of derivative works?
- Do I want developers return their modifications to the community, or me as original author, in particular?

Choosing a free license: Cases

- Do I want to allow privatization of derivative works?
- Do I want developers return their modifications to the community, or me as original author, in particular?
- Do I want to allow licensees to merge or link their program with mine?

Choosing a free license: Cases

- Do I want to allow privatization of derivative works?
- Do I want developers return their modifications to the community, or me as original author, in particular?
- Do I want to allow licensees to merge or link their program with mine?
- Do I want widespread coverage, avoid any restriction and/or try to establish a standard?

Choosing a free license: Cases

- Do I want to allow privatization of derivative works?
- Do I want developers return their modifications to the community, or me as original author, in particular?
- Do I want to allow licensees to merge or link their program with mine?
- Do I want widespread coverage, avoid any restriction and/or try to establish a standard?
- Should my program run with one in particular? Have it any restrictions?

Choosing a free license: Cases

- Do I want to allow privatization of derivative works?
- Do I want developers return their modifications to the community, or me as original author, in particular?
- Do I want to allow licensees to merge or link their program with mine?
- Do I want widespread coverage, avoid any restriction and/or try to establish a standard?
- Should my program run with one in particular? Have it any restrictions?
- Is there risk that someone requiring a patent license over program?

Quick Reference For Choosing a Free Software License

License	Hackers like accepting code under it				
	Combine with proprietary and redistribute				
	Combine with GPL'ed code and redistribute				
	Must share source of redistributed version				
	Must include patent license with contribution				
v	v	v	v	v	v
---	---	---	---	---	---
permissive	Y	Y	Y	N	N
GNU LGPL	Y2	Y1	Y	Y	Y
GNU GPL	?2	N	Y	Y	Y
Mozilla PL 1.1	Y2	Y	N3	Y	Y

Exercise: Find mistakes

Licencia	Se puede mezclar con otro software no libre y redistribuir	Se pueden privatizar las obras agregadas	Se pueden privatizar las modificaciones	Hay que licenciar patentes
GPLv2/v3				SI
LGPLv2/v3	SI			SI
MPL	SI	SI		SI
BSD	SI	SI	SI	

Conventional Matrix

Free ↑ ↓ Restricted	GPL	Everything free all the time
	LGPL	Some things free
	BSD	Free but can be made proprietary
	Shared Source	Free to some users
	Proprietary License	It's not free; you have to pay for it.
	Proprietary Closed Source	It's not available.

Matrix Including Developer's Choice

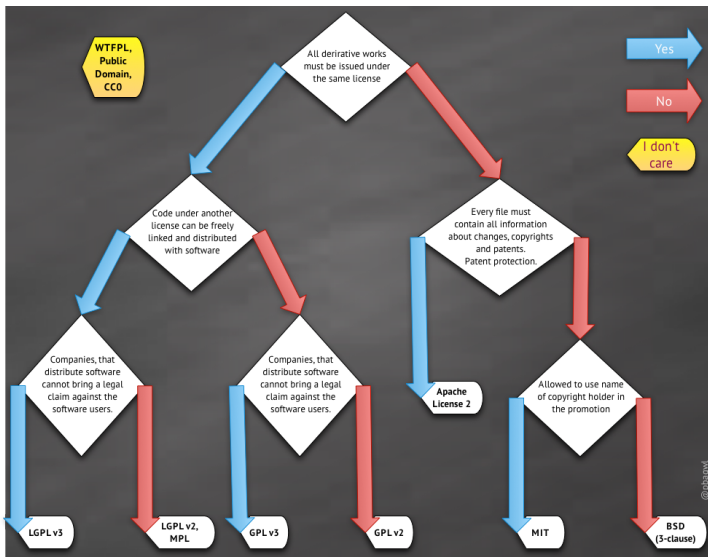
Free ↑ ↓ Restricted	BSD	Take the code and do what you want
	LGPL	Take the code, and do what you want with some.
	GPL	Take the code, but you have to make it open source.
	Shared Source	Take the code if you join our club.
	Proprietary License	Don't take anything unless you pay.
	Proprietary Closed Source	Take nothing

- Some members of the community refuse to accept GPL'ed source code into their projects.
- Other members of the community strongly prefer GPL'ed source code over other licenses.
- **Nobody refuses** to accept code under **permissive licenses** such as BSD, X11, MIT...

- Almost nobody refuses to accept LGPL'ed code, except the Apache Foundation, saying that they think it would impose LGPL requirement upon the proprietary code (when they are linked via the Java class-loading mechanism).
- The FSF disagrees with this statement, asserting that such linking falls under section 6 of the LGPLv2 (“linking exception”).

- MPL 1.1 can be specifically amended to allow combining with GPL (section 13).
- You should also get your employer (if you work as a programmer) or school, to sign a “copyright disclaimer” for the program, if necessary.

Choosing a Free Software License



Applying a free license

- LICENSE or COPYING file.
- Copyright and license summary at the beginning of each source file.
- It should have at least the “copyright” notice and a link to the full version of the license.
- Also add information on how to contact you by electronic and paper mail.
- If the program is interactive (terminal), make it output a short copyright notice when it starts in an interactive mode.
- If it has a GUI, a menu can include copyright notice (or even the full license).

Applying a free license

- **File-by-file method**: widely adopted in the free software community during the past two decades.
- **“Single COPYRIGHT file”** method (FSF preferred)
- Question: advantages and disadvantages

Example: How to Apply the GPL to Your Work

<one line with program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see
<<http://www.gnu.org/licenses/>>.

Example: How to Apply the Apache License to Your Work

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the ‘‘License’’);
you may not use this file except in compliance with the License.
You may obtain a copy of the License at
<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing,
software distributed under the License is distributed on an ‘‘AS
IS’’ BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express or implied. See the License for the specific language
governing permissions and limitations under the License.

Example: How to Apply the ISC License to Your Work (1)

- Below is an example license to be used for new code in OpenBSD, modeled after the ISC license.
- It is important to specify the year of the copyright. Additional years should be separated by a comma, e.g. Copyright (c) 2003, 2004
- If you add extra text to the body of the license, be careful not to add further restrictions.

Example: How to Apply the ISC License to Your Work

```
/*
 * Copyright (c) CCYY YOUR NAME HERE <user@your.dom.ain>
 *
 * Permission to use, copy, modify, and distribute this software
 * for any purpose with or without fee is hereby granted, provided
 * that the above copyright notice and this permission notice appear
 * in all copies.
 *
 * THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS
 * ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING
 * ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS.
 * IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL,
 * DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY
 * DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR
 * PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR
 * OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
 * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
 */
```

Maintaining Permissive-Licensed Files in a GPL-Licensed Project

- Take care to comply with the minimal conditions of the permissive license.
- Full preservation of copyright, permission, and warranty disclaimer notices.
- Failure to do so may result in infringement of the copyright.

Adding GPL'd modifications to permissive-licensed files

- The developer of the GPL'd project must comply with the notice preservation requirement.
- He should place a new copyright notice and permission notice above the existing one
- He should make clear that the developer has modified the file.

Adding GPL'd modifications to permissive-licensed files

```
/*  
* Copyright (c) 2007 GPL Project Developer Who Made Changes <gpl@example.org>  
*  
* This file is free software: you may copy, redistribute and/or modify it  
* under the terms of the GNU General Public License as published by the  
* Free Software Foundation, either version 2 of the License, or (at your  
* option) any later version.  
*  
* This file is distributed in the hope that it will be useful, but  
* WITHOUT ANY WARRANTY; without even the implied warranty of  
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU  
* General Public License for more details.  
*  
* You should have received a copy of the GNU General Public License  
* along with this program. If not, see <http://www.gnu.org/licenses>.  
*
```

Adding GPL'd modifications to permissive-licensed files

```
/*  
* Copyright (c) 2007 GPL Project Developer Who Made Changes <gpl@example.org>  
*  
* This file is free software: you may copy, redistribute and/or modify it  
* under the terms of the GNU General Public License as published by the  
* Free Software Foundation, either version 2 of the License, or (at your  
* option) any later version.  
*  
* This file is distributed in the hope that it will be useful, but  
* WITHOUT ANY WARRANTY; without even the implied warranty of  
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU  
* General Public License for more details.  
*  
* You should have received a copy of the GNU General Public License  
* along with this program. If not, see <http://www.gnu.org/licenses>.  
*  
* This file incorporates work covered by the following copyright and  
* permission notice:  
*  
* Copyright (c) YEARS_LIST, Permissive Contributor1 <contrib1@example.net>  
* Copyright (c) YEARS_LIST, Permissive Contributor2 <contrib2@example.net>  
*  
* Permission to use, copy, modify, and/or distribute this software  
* for any purpose with or without fee is hereby granted, provided  
* that the above copyright notice and this permission notice appear  
* in all copies.  
*
```


Mixing GPL and BSD code just takes care

- 1 Identify all contributors.
- 2 Identify which contributions create a copyright interest.
- 3 Secure permissions from current copyright holders.
- 4 Create a system for tracking permissions on future contributions.
- 5 Publicize the new licensing policy.

Distribute software under two different sets of terms and conditions. Motivations:

- License compatibility (Perl, Mozilla/Firefox, MySQL).
- Market segregation based business models (MySQL Enterprise)
- Allows the holder to offer customisations, early releases, generate other derivative works or grant rights to third parties to redistribute proprietary versions.

Dual-licensed under the GPL and a permissive license?

- Developers sometimes attempt to explicitly dual licensing to be available under permissive terms as well as to be incorporable into GPL'd projects.
- But developers should simply use the permissive license without any further reference to the GPL (Software Freedom Law Center).

- Two licenses are incompatible if it is not possible combining both works in compliance with the terms of both licenses at the same time.
- It affects to distribution, not the use.
- If two licenses are free does, it doesn't imply are compatible.
- Copyleft licenses are mutually incompatible, unless compatibility is declared explicitly.
- Support for “linking”: even if not allowed mix or integrate software with different licenses, maybe it can be linked.

Forking (1)

- A piece of software is modified and developed separately by another team development, and distributed under a different name, and maybe other license.
- The forks can be possible **only** with free/open source software.
- The GPL software has tendency to avoid forking (we must keep the original license).
- The BSD-style licenses are forked easily.

Forking (and 2)

- **Forkaphobia**: Some people consider forking a bad thing (waste efforts, bitter disputes...).
- But there are successful cases: XOrg/XFree86, 386BSD, OpenBSD, Gnu-Emacs/XEmacs, changes of license (GForge, OpenSSH).
- If forking is easy, experimentation with ideas can be pursued while still remaining downstream.
- But if forking is difficult, experimenters are reduced to dissenters (resulting in divorce).
- **Forking paradox**: “the easier it is to fork the software, the more difficult it is to fork the community” (Bryan Cantrill)

- Warranty disclaimers and limitation of liability clauses are common in software.
- There are legal doubts about the effectiveness of these clauses: **would not apply to consumers.**
- This clauses are valid when there is **no commercial service.**
- The proprietary licenses using similar terms: is a myth to accept greater responsibility.
- It must be considered legal guarantees that apply to both open source and proprietary software.
- The free licenses allow (sometimes) to add extra warranty clauses.

Anti-patterns: mistakes (B. Cantrill)

- No source! Don't do this: you gain nothing and you lose credibility (perhaps forever). Ex. HP webOS
- Forkaphobia: FOSS that cannot be forked has no vitality.
- Governance orgy (constitutions, elections...): corrosive effects, factionalize and undermine community.
- Competitive paranoia: companies that adopt your technology **are not your competitors!**

Anti-collaborative licensing?

- **Competitive paranoia**: anti-collaborative licensing erect walls **within** open source software.
- Strong copyleft (and affero clauses) excludes competitors... **but also collaborators!** It prevents cross-pollination across open source code bases!
- And prevent integration of FLOSS technologies! (ex. DTrace, ZFS and Linux).
- If you want a collaborative copyleft license, **consider weak copyleft** (MPL, CDDL, LGPL...).

- Case study: Apache License v2 and GPL (v2 and v3) (in)compatibility.

Exercise 2: Case study

- Case study: GPL and CDDL incompatibility.

Exercise 3: Case study

- Case study: The Atheros Case.

Case studies

Legal Aspects – Master on Free Software 2012-13

Miguel Vidal

<http://flossystems.com>

Twitter: @mvidallopez

November 16th, 2012

