



e-Primers on Free/Open Source Software

Free and Open Source Software Licensing Primer

Shun Ling Chen

**International Open Source Network (IOSN)
An Initiative of UNDP's Asia-Pacific Development Information Programme**

<http://www.iosn.net>

Published by
the United Nations Development Programme's
Asia-Pacific Development Information Programme (UNDP-APDIP)
Kuala Lumpur, Malaysia

Web: <http://www.apdip.net/>

Email: info@apdip.net

© UNDP-APDIP 2004

The material in this book may be reproduced, republished and incorporated into further works provided acknowledgment is given to UNDP-APDIP. This publication is released under the Creative Commons Attribution 2.0 license.

For full details of the license, please refer to the following:

<http://creativecommons.org/licenses/by/2.0/legalcode>

ISBN:

Table of Contents

Introduction.....	4
An overview of Intellectual Property Rights.....	5
How is software protected?	6
Copyright Basics.....	6
What can be copyrighted?.....	6
Is there anything required to be done to get a work protected?.....	6
What are the rights granted to the copyright holder?.....	6
Copyright law protection has been expanded largely with time.	7
The expansion of copyright protection.....	7
From national to international protection.....	7
Protection upon creation, the abolishment of formality requirements.....	8
Copyright law, a balance between public and private interests.....	8
Software and copyright protection.....	9
Extension of copyright law protection to software in 1980s	9
Copyright Protects Both Source and Object Code under TRIPs.....	9
Users' Rights Disguised in Proprietary Licensing Models	10
What is FOSS? How is FOSS different from proprietary software?.....	10
Free Software.....	11
Background: transition in information technology industry and legal institution	11
Richard Stallman on a stark moral decision.....	11
Free Software Definition.....	11
Creating a Free Software Environment.....	12
Open Source Software.....	13
Open Source Definition	13
OSI approved Licenses.....	14
Free or Restrictive?	14
How to make the source free/open?.....	15
GNU General Public License (GNU GPL or GPL).....	24
Copyleft	24
Major terms and conditions of GPL	25
User's freedoms.....	25
No Warranty.....	25

License Issued Directly from the Author.....	25
GNU Lesser General Public License (GNU LGPL or LGPL).....	26
Major terms and conditions of LGPL.....	26
Works based on the Library	27
Works that uses the Library.....	27
BSD Style Licenses.....	28
Multiple Licensing.....	28
Now, source code is open, how about documentations?	29
GNU Free Documentation License (GNU FDL).....	29
Creative Commons Licenses.....	29
Scenarios.....	33
End user (individual/business/government)	33
Developer (individual, business).....	34
When starting a new project.....	35
When modifying an existing module.....	36
When integrating different FOSS modules into one service.....	37
Vendor/ Producer (business).....	38
Government Sponsored Projects.....	39
Government Funded FOSS Projects- Cases from Asia-Pacific	40
FOSS Project under Government-affiliated Research Institute— Multi-Lingual	
Editor, Japan.....	40
National FOSS Project – Taiwan.....	41
Online Legal Resources and Materials.....	42
Glossary.....	43
Copyleft.....	43
Copyright.....	43
Copyright holder.....	43
Derivative work.....	43
Fair Use.....	44
License.....	44
Multiple Licensing.....	44
Public Domain.....	44
Source Code.....	44
Sublicense.....	45
Warranty Disclaimer.....	45

Introduction

A general introduction to the FOSS, its benefit and its development, has been explained in the first primer of this series of primers focused on the FOSS movement, and is available from the International Open Source Network website at:

http://www.iosn.net/downloads/foss_primer_current.pdf

Under the current legal structure, licenses are the key factors how we can possibly make the source of a program free/open. As David Wheeler said, free/ open source software (FOSS) programs the programs whose licenses give users the freedom to run the program for any purpose, to study and modify the program, and to redistribute copies of either the original or modified program without having to pay royalties to previous developers.¹

Starting from mid 1980s, the birth of GNU General Public License (GNU GPL or GPL) has allowed a divergent model of software development.² Following the GNU GPL, various FOSS licenses have been drafted and adopted by FOSS communities, academic institutes, and commercial companies. The number is growing rapidly. In early 2003, 43 licenses were recognized by the Open Source Initiative (OSI) as open source licenses, and in July 2004, the number has amounted to 54. The number will still be growing.³

FOSS licenses are many, legal issues related are numerous, and possible legal disputes are uncountable. Nevertheless, the aim of this primer is to provide a brief introduction on licensing issues involved with using the different FOSS licenses.

The primer will begin with a brief overview on intellectual property rights, then moves onto the idea and the development of copyright protection, the main category of intellectual property. Software is protected mainly by copyright, and it is also under which the FOSS licenses are structured. After the introduction to the legal structure, the primer will briefly explain how the FOSS movement uses licenses as a way to create a different model of programming.

FOSS licenses are many and they might differ from each other in very different ways. Due to page

¹ Wheeler, David, "Why OSS/FS? Look at the Numbers!" [home page online]; available from http://www.dwheeler.com/oss_fs_why.html, Internet; accessed on November 7, 2003.

² GNU is a recursive acronym for "Gnu's not Unix". Available from <http://www.fsf.org/>, Homepage, Internet, Accessed 5 July, 2004

³ For example, there were 5 licenses approved in Feb 2004, and 2 licenses were added to the list of approved licenses again in June 2004. available from <http://opensource.org/weblog/2004/01/03#newsblog> and <http://opensource.org/weblog/2004/06/03#Jun2-04>, Internet, Accessed 5 July, 2004.

limits, only the more pervasively adopted licenses will be shortly mentioned here, and the focuses would be on three most pervasive licenses, the GPL, the GNU Lesser General Public License (LGPL) and the Berkeley Software Distribution (BSD) style licenses. Their importance lies not only in the large numbers of FOSS projects licensed under, but they also represent very different styles of FOSS licensing.

In the last section of the primer, some scenarios will be provided for end-users, developers and vendors to remind them of some related copyrights issues. And with the growth of attention paid on FOSS development from governments, the primer also include 2 cases regarding to government sponsored FOSS projects.

An overview of Intellectual Property Rights

Nowadays, the intangible products of the human beings' creative activities are considered to be a kind of property and protected as their tangible counterparts. The idea of intellectual property rights has been generally accepted and legal institutes are built to offer protection to their possessors. Although copyrights, patent, trademark and trade secret all fall into the greater category of intellectual property, the essence of each differs from each other.

Trade secret is protected to avoid being accessed by its owner's competing business entities. It can be done through a variety of civil and commercial means, such as confidentiality agreements or non-disclosure agreements signed by those who are given accesses to those special knowledge and information.⁴

Trademarks are the distinctive names, phrases, symbols, designs or pictures or styles used by business to identify itself and its products or services to its consumers. In many countries, colors, three dimensional marks, sounds and even smells are also capable of trademark protection.⁵

While trade secrets promotes competency of a business by keep certain information to the public, patents are designed to provide the inventor the monopoly of certain newly developed knowledge for a period of time (usually 20 years) in exchange of its disclosure. Typically, to gain such exclusive rights, the inventor is required to file a patent application, which will be reviewed by a designated patent examiner. Novelty of the invention is an essential element in granting a patent.⁶

⁴ Available from http://en.wikipedia.org/wiki/Trade_secret, Internet; accessed 27 July 2004

⁵ Available from <http://en.wikipedia.org/wiki/Trademark>, Internet; accessed 27 July 2004

⁶ Available from <http://en.wikipedia.org/wiki/Patents>, Internet; Accessed 27 July 2004

Copyright is applied to various kinds of human creative works, such as literary works, music compositions, paintings and also software. The copyright holder of a work is entitled to exclusive rights of the reproduction, modification, distribution and public display of the work. Unlike patent protection, copyright law is applied to a work upon its creation, regardless of its novelty. But the ideas embraced in the work is not protected, copyright only prevents others from copying the copyright holder's particular way of expressing the ideas.⁷

How is software protected?

Software, like other literary works, is now protected under Copyright Law. Although in recently years it is argued that source code and algorithm should be patentable and has already been granted patents in some case, software patents are still questioned and contested by many, especially from the FOSS community. The primer is focusing on the FOSS licenses, the software patent is itself too complicated an issue and won't be addressed much here.

Copyright Basics

What can be copyrighted?

Copyright protects the expressions of ideas in different forms of works, including literary, dramatic, musical, artistic, and other intellectual works.⁸ Since 1980s, a software program, like other literary works, is protected under copyright law.⁹

The ideas expressed in such works themselves are not protected.¹⁰

Is there anything required to be done to get a work protected?

Nowadays, copyright law does not require formalities to gain copyright protection. The author does not need to publish, does not need to register, does not need to pay registration fee of any kind, and nor does he need to have a copyright notice attaching to it, the copyright of the work immediately becomes the property of the author who created the work. Copyright protection is automatically applied to a work once it is created.¹¹

What are the rights granted to the copyright holder?

The "copyright" is actually a bundle of rights, including the right to reproduce the work, the right

⁷ Available from <http://en.wikipedia.org/wiki/Copyright>, Internet, Accessed 27 July 2004

⁸ Available from <http://www.wipo.int/copyright/en/faq/faqs.htm#rights>, Internet, Accessed 28 June 2004

⁹ Available from http://www.wipo.int/copyright/en/faq/faqs.htm#P39_5114, Internet, Accessed 28 June 2004

¹⁰ Available from <http://www.wipo.int/copyright/en/faq/faqs.htm#ideas>, Internet, Accessed 28 June 2004

¹¹ Available from <http://www.wipo.int/copyright/en/faq/faqs.htm#rights>, Internet, Accessed 28 June 2004

to prepare derivative works based upon the work, the right to distribute copies of a work, the right to perform and display a copyrighted work publicly, and some other kinds of rights defined in the copyright law.¹² Without an explicit expressed consent from the copyright holder, it is illegal for anyone to violate any of the copyright holder's rights.

Copyright law protection has been expanded largely with time.

The expansion of copyright protection

The first copyright legislation (Statute of Ann, 1710)

Compared to other legal institutions, copyright law came relatively late in human civilization. The first known copyright legislation was the Statute of Ann, enacted in 1710, the Great Britain.¹³ For a newly created work, the Statute of Ann protects the copyright holders the right to print and reprint the books and other writings for 14 years.

All-dimensional expansion of copyright protection

We can see from the Statute of Ann that at its initial age, the scope of copyright protection is quite limited. The works protected are limited to books and other writings", the rights granted to the copyright holder are limited to the "print" and the "reprint" of the work, and the length of the protection is limited to "14 years".

Nowadays, copyright law protects much more than that. The works protected now includes paintings, sculpture, music compositions, music recording, architecture, and software. The rights granted to the copyright holder now expand to the print, reprint, modification, public display, public perform, distribution of the work. And the life of copyright is lengthened to 50 years after the author has died. (In Europe and the US, it has been expanded to 70 years.)¹⁴

From national to international protection

The expansion of copyright protection affects not only the copyright law in one jurisdiction, but has become an international standard.

Berne Convention

In the late 19th century, while copyrighted works gradually became an important item in the international trade, the trans-national copyright protection gradually became a serious issue.

¹² Richard, Jones, "Copyright Protection for Computer Software in the United States", 2002, Available from <http://www.ladas.com/Patents/Computer/SoftwareAndCopyright/Softwa02.html>, Internet Accessed on 28 June 2004

¹³ Available from <http://www.copyrighthistory.com/anne.html>, Internet, Accessed on 28 June 2004

¹⁴ Little, Jonathan, "History of Copyright- A Chronology", 2002, Available from <http://www.musicjournal.org/01copyright.html>, Internet, Accessed on 28 June 2004

Started with its European signatories, the Berne Convention 1886 first introduced the national treatment principle. While Berne Convention required its signatories to uphold certain basic copyright protection, the national treatment principle protects the work of a foreign copyright holder the same way as it protects its nationals. Thus the Berne Convention has created an international standard for copyright protection.

However, without a dispute resolution mechanism, the protection Berne Convention could offer is relatively weak, for it would be too costly for a copyright holder to claim his rights in a foreign country where he believes his rights have been infringed.

More enforceable international standard - WTO and Trade Related Intellectual Property Agreements

In the 1990s, the World Trade Organization (WTO) and the Trade Related Intellectual Property Agreements (TRIPs) has become a more powerful structure for international copyright protection, for every economy that intends to become a member of WTO is required to sign the TRIPs, and that every TRIPs signatory must agree to comply with all of the key sections of the Bern Convention. And WTO also provides a dispute-settlement and enforcement mechanism for copyright infringements among member countries. Thus, the international copyright protection has become a more enforceable standard.¹⁵

Protection upon creation, the abolishment of formality requirements

As a principle set forth by the 1908 Berne Convention, copyright protection is applied to a work once it is created without requiring any formality.¹⁶ The author does not need to register, nor even publish a work, to enjoy the full-fledge copyright protection. Though the scope and terms of copyright law in different countries might have been changed with time, but for Berne Convention signatories, unless otherwise expressed explicitly by the author, the law assumes that the author claims all kinds of rights granted to him.

The 1908 revision of Berne Convention has abolished the formality requirement of copyright protection. Since then, for Berne Convention Union countries, a work is protected upon its creation, without any need to register, so are unpublished works. But again, with the standardization of international copyright protection, copyright laws in different countries have been revised to comply with such standard. For example, anticipating joining Berne Convention Union, the US revised its Copyright Act and abolished formality requirements in 1976.¹⁷

¹⁵ Story, Alan, "Don't Ignore Copyright, the 'Sleeping Giant' on the TRIPs and International Educational Agenda", pp.132-33, in Global Intellectual Property Rights, Knowledge, Access and Development, Drahos, Peter and Ruth Mayne Eds., NY: MacMillan, 2002

¹⁶ Lessig, Lawrence, "Free Culture", Footnote 194, Available from <http://www.jus.uio.no/sisu/freeculture.lawrence.lessig/14>, Internet, Accessed on 29 June 2004

¹⁷ Little, Jonathan, "History of Copyright- A Chronology", 2002, Available from

Copyright law, a balance between public and private interests

Comparing to other legal institutions, copyright protection is a relatively new invention in human history. The development of copyright regulation with time also reflects the social and technological transformation regarding to human creative activity and its distribution. While granting exclusive private rights to authors or copyright holders has been considered as a way to promote human creative activity, copyright law bears the recognition of the larger public interest, particularly on education, research and access to information.¹⁸

Various ways has been adopted in order to acquire the balance between the public and private interests. In Statute of Ann, the law stated that the authorities could limit and settle the price of printed books according to the best of their judgments. In the US Constitution, under congressional decision, exclusive rights to their writings and discoveries could be granted to authors and inventors within limited time. In copyright law, fair use and other exceptions are to avoid the drawbacks of excessive exclusive right.

These legislative efforts are made to attain a balance between confronting interests. While recognizing the exclusive rights granted by the law might hinder the public accessibility of information and knowledge, certain limits, such as fair use and first sale doctrine, are imposed upon such exclusive private rights during its limited lifetime.

Software and copyright protection

Extension of copyright law protection to software in 1980s

The US Computer Software Copyright Act 1980 started to regard computer programs as works that could be protected by copyright law. Since then, it has become an international trend that the copyright protection could be applied to computer software. The WIPO Copyright Treaty (1996) also states that computer software is a kind of the works protected by copyright law.

Copyright Protects Both Source and Object Code under TRIPs

There's one thing that needs to be mentioned about copyright protections on software. Software could be expressed in both source code and object code. While the ideas expressed in the form of

<http://www.musicjournal.org/01copyright.html>, Internet, Accessed on 28 June 2004

¹⁸ As stated in the Preamble of WIPO Copyright Treaty, Available from <http://www.wipo.int/clea/docs/en/wo/wo033en.htm>, Internet, Accessed on 29 June 2004

source code could be perceived by trained programmers, they could not be understood by human beings when they're expressed in the form of object code. Since it is stated in the TRIPs that the copyright protection of computer software apply to both source code and object code forms,¹⁹ so in practice, proprietary software companies tend to release their product only in object forms, and take the source code of the product as their trade secret.²⁰

As we know that copyright law protects only the expressions of ideas but not the ideas themselves, when receiving a copy of a literary work, a music composition, the ideas expressed in the work could be perceived and might become the inspirations of other new works, and thus is able to contribute to the human intellectual development. But when software distributed only in the form of object code could also receive copyright protection, this means that the proprietary company could enjoy full-fledge copyright protection without delivering the ideas. The knowledge embraced in the making of the software is thus not accessible to the trained developer, not to say the general public. Therefore, the copyright protection on software in such cases might not be in line with the essence of copyright law, i.e. the acquirement of the balance between private and public interests.

Users' Rights Disguised in Proprietary Licensing Models

Under traditional proprietary licenses, source code is not accessible. Proprietary licenses might even forbid you from studying the program. E.g. even in the licenses for developers, Microsoft End User Agreement MSDN, the Microsoft Developer Network Subscription, reverse-engineering, decompilation and disassembly are not allowed except and only to the extent that it is expressly permitted by the applicable law.²¹

For end users, the proprietary licenses usually allow only one copy for each computer. That means, if the user has 1 desktop and 1 laptop, or 2 desktops, he will have to purchase 2 copies if he wants to run the program legally on his 2 machines. When there're bugs in the program he legally purchased, since he does not have access to the source code, and is not allowed to study the program, he will then stay vulnerable and passive to be notified by the proprietary companies or hackers on the existence of such bugs. Even if he is notified, he will not be able to legally debug by himself or use unofficial patches, since the modification of the program is not granted either.

Under traditional proprietary licensing model, end users are set completely bare-handed. The FOSS movement has contributed for a positive transformation of such situation, As stated by the Free Software Foundation (FSF) when it was founded in 1985, it is dedicated to promote users' rights to

¹⁹ Available from http://www.wto.org/english/docs_e/legal_e/legal_e.htm#TRIPs, Internet, Accessed on 28 June 2004

²⁰ Halligan, R. Mark, "How to Protect Intellectual Property Right in Computer Software", Available from <http://my.execpc.com/~mhallign/computer.html>, Internet, Accessed on 1 July 2004

²¹ Available from <http://www.msdn.net/EULA/NA/English.aspx>, Internet, Accessed 04 Aug 2004

use, study, copy, modify, redistribute computer programs,²² the rights which are largely disguised, if not ignored, in traditional proprietary licensing models.

What is FOSS? How is FOSS different from proprietary software?

The development of Free/Open Source Software might be taken as a reaction to the existing legal instrument on software copyright from the software developers' community. Both take the access to source code as a prerequisite, and go further to other rights bundled in copyright, such as the right to make copies of a work, the right to distribute the copies of a work, and the right to prepare derivative works from a work.

Free Software

Background: transition in information technology industry and legal institution

In 1970s, both the transformation in legal institution and the information industry (IT) industry contributed to the formation of the free software. The US revised their copyright law, and IT companies also kept claiming the software as a copyrightable work.²³ On the other hand, while earlier software was bundled with hardware in the hardware market, the IT industry started to consider the software as a distinguishable market.²⁴ They started to recruit more developers from labs in research institute, and asked them to sign confidential agreements.

Richard Stallman on a stark moral decision

Since prior to that, the common practices in labs were largely about sharing sources and copies, for Richard Stallman (RMS) who worked in the Massachusetts Information Technology (MIT) lab at that time, such a transformation was accompanied by the diminishing of the community which honors sharing and the ethic of "helping your neighbors", a community he was proud to belong to. Therefore, as a talented programmer who could easily sign a contract and a confidential agreement with a proprietary company in exchange for a well-paid salary, he said he was on a "stark moral decision". He was put to decide whether to join the software industry, or to struggle for the survival and sustainability for the community he belongs. It was the latter one he chose, and developed the necessary infrastructure for the maintainability of the community, i.e. the Free Software.²⁵

²² Available from <http://www.fsf.org/fsf/fsf.htm>, Internet, Accessed 04 Aug 2004

²³ Richard, Jones, "Copyright Protection for Computer Software in the United States", 2002, Available from <http://www.ladas.com/Patents/Computer/SoftwareAndCopyright/Softwa04.html>, Internet, Accessed on 28 June 2004

²⁴ Campbell-Kelly, Martin, "Development and Structure of the International Software Industry, 1950-1990", Available from <http://www.dcs.warwick.ac.uk/~mck/Personal/SoftIndy.pdf>, Internet, Accessed on 1 July, 2004

²⁵ Stallman, Richard, "The GNU Operating System and the Free Software Movement", pp.53-56, CA: O'Reilly & Associates, Inc., 1999.

Free Software Definition

Free Software is about granting the users' freedom to run, copy, distribute, study, change and improve the software. RMS developed the definition of Free Software as followed,

- The freedom to run the program, for any purpose (freedom 0).
- The freedom to study how the program works, and adapt it to your needs (freedom 1). Access to the source code is a precondition for this.
- The freedom to redistribute copies so you can help your neighbor (freedom 2).
- The freedom to improve the program, and release your improvements to the public, so that the whole community benefits (freedom 3). Access to the source code is a precondition for this.²⁶

We could see from the 4 freedoms that other than emphasizing the access to source code, the Free Software Definition also pointed out the right of the user to copy, to distribute the copy, to modify the software, and to distribute the derivative work of a copyrighted work. (The previous 3 kinds of rights are granted exclusively to its copyright holder by the existing copyright law.)

Creating a Free Software Environment

- The GNU project and the Free Software Foundation

The Free Software Definition claims strongly the rights of non copyright holders. But without the environment that allows such an ideal, the above freedoms would be merely unrealistic slogan. The GNU project was launched in 1984 to develop and complete UNIX style operating system which is free software, the GNU system, and the project went further to include other application software. In 1985, the Free Software Foundation (FSF) was established to promote the idea of free software. They promote the development and use of free software not only by distributing the software which happens to be available, but also by concentrating on the development of free software, aiming at creating a coherent system, the GNU operating system, and provide alternative solutions to proprietary software.

<http://www.fsf.org/gnu/thegnuproject.html>

<http://www.fsf.org/fsf/fsf.html>

- GNU General Public License

Under existing legal structure, the copyright protection is granted to the exclusive disposal of the copyright holder once the work is created. Without an explicit expression, it is assumed that the

²⁶ Available from <http://www.fsf.org/philosophy/free-sw.html>, Accessed on 31 May 2003

copyright holder claims all the rights granted to him, and that any dissimilar subject opinion must be ignored. That is to say, the law burdens the copyright holders with the explicit expression not to claim some or all rights granted to him.

While sometimes people might not know how to make such explicit expression, the creation of GNU General Public License (GNU GPL or GPL) serves as a legal tool to bring out the freedoms stated in the Free Software Definition and to maintain the environment that supports free software. The GPL is a license, but different from proprietary licenses, grants the users the rights that were taken as exclusively by the copyright holder by law and by business practices, including the right to access to the source code, the right to run the program, the right to make copies and redistribute the copies, and the right to modify the program and distribute the modified program. On the other hand, although the GPL grants the users many rights and freedoms to use the software, it also sets certain limitations in order to make sure the free software and its derivations could remain as free as it is.²⁷

When a work is licensed under GPL, it means the author of the software still claims the copyright of the work, but adopts a different license as a way of explicit expression to allow the public to have greater freedom to use his work than the copyright law assumes.

(The characteristics of GPL will be further explained in a separate section below.)

Open Source Software

While advocates free software take the freedom of software as a moral issue, some turned to promote the idea of “Open Source Software” that focus more on the technical values of FOSS and is more business friendly.²⁸ The Open Source Initiative (OSI) operates as an organization to promote the open source campaign by managing and promoting the Open Source Definition and its certification mark for open source licenses and products.

Open Source Definition

The Open Source Definition is a revision from a policy document of the Debian GNU/Linux Distribution, which served to clarify which licenses are free licenses.²⁹ The OSI explains the basic idea of Open Source:

²⁷ See GPL Preamble, Available from <http://www.fsf.org/licenses/gpl.txt>, Internet, Accessed on 31 May 2004

²⁸ Wong, Kenneth and Phet Sayo, “Free/Open Source Software, A General Introduction”, pp. 6-7, 2004, Available from <http://www.iosn.net/index.php?module=ContentExpress&func=display&ceid=95>, Internet Accessed on 31 May 2004

²⁹ Perens, Bruce, “The Open Source Definition”, in Open Sources, Voice From the Open Source Revolution, p173, CA: O'Reilly & Associates, Inc., 1999.

The basic idea behind open source is very simple: When programmers can read, redistribute, and modify the source code for a piece of software, the software evolves. People improve it, people adapt it, people fix bugs.

From above explanations, we can see that the Open Source people also emphasize the same rights stated in the Free Software Definition, including the users' access to source code, users' right to copy the work, distribute the copies, modify the work and distribute the derivative works.

Comparing to the Free Software Definition, the Open Source Definition, which consists of 10 clauses, is relatively lengthy. Besides the clauses regarding to the accessibility of source code (sec 2), users' right to copy and distribute the original work (sec 1), users' right to modify the work and distribute the derivative work (sec 3), the Open Source Definition also has several non-discrimination clauses (sec 5, 6, 8, 9, 10). Though not stated in the same way, such non-discriminatory ideas could also be found in the Free Software Definition. Sec 7 of the Open Source Definition is to prevent the source from being closed again, thus is also central to free software. The emphasis on the integrity of the author's source code and the requirements on the distributing of modified works (sec 4) might be the only thing we don't find explicitly in the Free Software Definition.

<http://opensource.org/docs/definition.php>

OSI approved Licenses

With the guideline stated in the Open Source Definition, OSI certifies and recognized licenses as Open Source Licenses through certain procedure. The certification procedure is upon request, and if a newly approved open source license will be added to a list open source licenses maintained by OSI.

http://opensource.org/docs/certification_mark.php

The number of OSI approved license also grows with the FOSS development in different area. We can find those licenses derived from FOSS community, e.g. GPL, Lesser General Public License (LGPL), PHP License, Nethack General Public License, those from academic/research institute, e.g. NASA Open Source Agreement, MIT License, University of Illinois/NCSA Open Source License, and those from proprietary companies who have adopted FOSS as part of their strategies, e.g. Apple Public License, Eclipse Public License, Qt Public License, Mozilla Public License. Actually a large proportion of the OSI approved licenses are those from proprietary companies.

Free or Restrictive?

Although Free Software Definition and Open Source Definition don't seem to have too many

differences from each other, they do differ in some rhetoric which might also again reflect their differences in philosophy.

<http://www.fsf.org/philosophy/categories.html>

For example, some people might refer to the classic free software licenses such as GPL and LGPL as “highly restrictive” and “restrictive”, for they have set up many restrictions to make sure the free software and their derivative works stay free. But for the FSF, these restrictions are prerequisites for a healthy environment for free software, thus the free software licenses are not supposed to be called as “highly restrictive” or “restrictive licenses”.

For example, the FSF also maintains a list of free software license and non-free software licenses. Although sometimes FSF might refer to those relatively simple licenses as “permissive”, they never refer to the more complicated free licenses as “restrictive”.

However, although with such philosophical differences, the FSF and OSI agree with each other in most cases on which license could be classified as FOSS license.

Among the 26 OSI approved licenses analyzed by FSF, only 2 of them, the Original Artistic License and the Reciprocal Public License, were regarded as non-free licenses (see table in the annex).

How to make the source free/open?

As explained above, under current legal institution, the software is protected by the copyright law as other literary works. The FOSS movement thus makes use of the legal institution and has the authors of the software grant the users some of the rights which are granted exclusively by copyright law. The creation of FOSS licenses helps authors to easily make such an announcement and also forms as agreements of FOSS developers’ communities.

There are many FOSS licenses, and they differ in characteristics. Later on we will focus on the three major kinds of licenses, GNU GPL, the LGPL and the BSD style License, for they not only represent the three very different styles of FOSS licensing, but are also the most pervasively adopted licenses.³⁰ But first by looking at the following table offered by the Open source software foundry is Seeking for Software Freedom (OSSF), Institute of Information Science, Academia Sinica, might help us to get a quick and general overview (the table below is a simplified version).³¹

³⁰ If we look at the SourceForge.net, the largest FOSS development website, we can see GPL, LGPL and BSD are the 3 most adopted licenses. Among all 53026 projects that is licensed under OSI-approved licenses, 36962 projects are licensed under GPL, 5817 projects are under LGPL and 3813 projects are licensed under BSD. Available from http://sourceforge.net/softwaremap/trove_list.php?form_cat=14, Internet, Accessed 1 Aug 2004

³¹ Open Source Software Foundry is Seeking Software Freedom, A Comparison of FOSS Licenses, Available from

	Original Work				Derivative Work			
	As a principle, redistributions should provide source code	When redistribute WITHOUT source code, could the distribution of source code alone charge a fee higher than the physically transferring cost?	When redistribute WITH source code, could the distribution charge a fee higher than the physically transferring cost?	Sub-licensable?	Derivative works should adopt the same license as adopted by the original work	Source code is required to be open	The copyright notice of the original work should be attached	Documentation are required to provide
GPL	Yes	No	Yes	No	Yes	Yes	Yes	Yes
LGPL	Yes	No	Yes	No	Work based on the library			
					Yes	Yes	Yes	Yes
					The executable that links a "work that uses the Library" with the library			
					No	No.	Yes	Yes

	Original Work				Derivative Work			
	As a principle, redistributions should provide source code	When redistribute WITHOUT source code, could the distribution of source code alone charge a fee higher than the physically transferring cost?	When redistribute WITH source code, could the distribution charge a fee higher than the physically transferring cost?	Sub-licensable?	Derivative works should adopt the same license as adopted by the original work	Source code is required to be open	The copyright notice of the original work should be attached	Documentation are required to provide
BSD License	No	Yes	Yes	No	No	No	Yes	No
Artistic License	Yes	(Source code is always redistributed)	No	No	No	No	No	Yes
MIT License	No	Yes	Yes	Yes	No	No	Yes	No
Apache License v1.1	No	Yes	Yes	No	No	No	Yes	No

	Original Work				Derivative Work			
	As a principle, redistributions should provide source code	When redistribute WITHOUT source code, could the distribution of source code alone charge a fee higher than the physically transferring cost?	When redistribute WITH source code, could the distribution charge a fee higher than the physically transferring cost?	Sub-licensable?	Derivative works should adopt the same license as adopted by the original work	Source code is required to be open	The copyright notice of the original work should be attached	Documentation are required to provide
Apache License v2.0	No	Yes	Yes	No	No	No	Yes	Yes
Mozilla Public License v1.1	Yes	(Source code is always redistributed)	Yes	Yes	Yes, the additional rights described in MPL may be included in an additional document	Yes	Yes	Yes

	Original Work				Derivative Work			
	As a principle, redistributions should provide source code	When redistribute WITHOUT source code, could the distribution of source code alone charge a fee higher than the physically transferring cost?	When redistribute WITH source code, could the distribution charge a fee higher than the physically transferring cost?	Sub-licensable?	Derivative works should adopt the same license as adopted by the original work	Source code is required to be open	The copyright notice of the original work should be attached	Documentation are required to provide
Zlib/libpng License	No	Yes	Yes	No	No	No	No	Yes
Qt Public License v1.0	Yes	No	Yes	No	QPL requires all modifications must exist in a form separable from the original work, e.g. a Patch, (does not allow people modify the original work directly), and regulates patches with clauses that are similar to the clauses other licenses regulate the derivative works			
					No	Yes	Yes	No
Common Public License v.1	Yes	No	Yes	Yes	No	Yes	Yes	No

	Original Work				Derivative Work			
	As a principle, redistributions should provide source code	When redistribute WITHOUT source code, could the distribution of source code alone charge a fee higher than the physically transferring cost?	When redistribute WITH source code, could the distribution charge a fee higher than the physically transferring cost?	Sub-licensable?	Derivative works should adopt the same license as adopted by the original work	Source code is required to be open	The copyright notice of the original work should be attached	Documentationns are required to provide
	Original Work				Derivative Work			
	As a principle, redistributions should provide source code	When redistribute WITHOUT source code, could the distribution of source code alone charge a fee higher than the physically transferring cost?	When redistribute WITH source code, could the distribution charge a fee higher than the physically transferring cost?	Sub-licensable?	Derivative works should adopt the same license as adopted by the original work	Source code is required to be open	The copyright notice of the original work should be attached	Documentationns are required to provide

	Original Work				Derivative Work			
	As a principle, redistributions should provide source code	When redistribute WITHOUT source code, could the distribution of source code alone charge a fee higher than the physically transferring cost?	When redistribute WITH source code, could the distribution charge a fee higher than the physically transferring cost?	Sub-licensable?	Derivative works should adopt the same license as adopted by the original work	Source code is required to be open	The copyright notice of the original work should be attached	Documentation are required to provide
Academic Free License v. 2.1	No	Yes	Yes	Yes	No	No	Yes	Yes
PHP License v. 3.0	No	Yes	Yes	No	No	No	Yes	No
Open Software License v. 2.1	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes

	Original Work				Derivative Work			
	As a principle, redistributions should provide source code	When redistribute WITHOUT source code, could the distribution of source code alone charge a fee higher than the physically transferring cost?	When redistribute WITH source code, could the distribution charge a fee higher than the physically transferring cost?	Sub-licensable?	Derivative works should adopt the same license as adopted by the original work	Source code is required to be open	The copyright notice of the original work should be attached	Documentation are required to provide
Zope Public License v. 2.0	No	Yes	Yes	No	No	No	Yes	Yes
Python Software Foundation License v. 2.1.1	No	Yes	Yes	No	No	No	Yes	Yes

All the FOSS licenses discussed in the above table, by definition, have the following common features,

1. Source code of the original work should be open;
2. It is allowed to make copies of the original work;
3. Distribution of the original work is allowed, and copyright notice should be attached in all distributions;
4. The license grant is non-exclusive, global, royalty-free, and for all purposes;
5. Warranty is disclaimed.

Even though certain rights have to be granted to the users, these FOSS licenses might differ from each other on how these rights could be exercised.

For example, though offering the access to the source code from the author is essential for FOSS licenses, whether redistributors are also required to do the same varies between different licenses. E.g., when redistributing a BSD-ed program, one is not required to provide the source code. For those who received the program directly from the author, the source code of the original will still be accessible since it is offered from the author.

And even when redistributors are also obliged to provide source code, there're also different regulations on the distribution fee the redistributors could collect. GPL and LGPL are particularly detailed in when a fee higher than the physical transmission fee could be collected. This is because GPL and LGPL provide redistributors various ways to distribute the program, whether with the source code or without the source code. But despite such freedom, GPL and LGPL also wish to keep the acts of redistribution in line with the spirit of free software. One can sell free software at any sum of money as he wishes, since the market would help to keep it at a reasonable range, but if the package sold is without source code, then the fee collected for the distribution of source code itself could not exceed the physical transferring cost.

Clauses on derivative works vary to a greater extent. Although the source code of original works is required by definition, source code of derivative works is not. Even if a FOSS license requires the source code of derivative works to be open, it might not have to be licensed under exactly the same license as the original work is. For example, although a derivative work of a GPL-ed program also has to be licensed under GPL, a derivative work of a BSD-ed program does not have to be licensed under BSD. As a matter of fact, a derivative work of a BSD-ed program doesn't even have to open its source!

It is also on this aspect, FOSS licenses differ with each other on the possibility to allow a FOSS

program to be combined with proprietary programs. When combining different programs into a larger project, it is quite inevitable that the larger project, while embracing the all or part of the source code of the programs it combined, becomes the derivative work of all the combined programs. For example, project ABC is combined with a GPL-ed program A, a BSD program B and a proprietary program C, and has source code from all three programs. Although when being a derivative work of program B, project ABC is not required to either be licensed under BSD License or even open its source, it is, however, as a derivative work of program A, required to be licensed under GPL. Thus the developer would have no choice other than to license the project ABC under GPL, or find an alternative solution of program A, especially if he wishes to make it a proprietary software project.

It is in this sense, that GPL is considered to have so called a “viral” effect, and is unfriendly to proprietary software developments. It is also in this sense, that the GNU Lesser General Public License (LGPL) is designed to encourage more uses of free libraries.

The three typical FOSS licenses, GPL, LGPL and BSD are explained further below.

GNU General Public License (GNU GPL or GPL)

GPL is the classic free software license. It is also the most well-known and most widely adopted one upon all FOSS licenses. GPL is an invention brought out to fulfill the freedoms defined by the Free Software. It is not only a license, but also a document to manifest the basic idea of Free Software.

Copyleft

The way GPL guarantees the freedom is also called “copyleft”. The name of such method “Copyleft” came from an imaginative contrary to what we might have all learned by heart. While traditional proprietors say “copyright, all rights reserved”, it was redressed as “Copyleft, all rights reversed”.

Copyleft is a way to prevent free software from being turned into proprietary software again. It uses copyright law, but serves the opposite of its usual purpose. Instead of a means of privatizing software, copyleft uses the rights granted to authors to keep the software free.³²

Instead of being a work in the public domain that everyone is free to make use of, a GPL-ed work or a copyleft-ed work is still a copyrighted work. The author of the GPL-ed work does not give up

³² Stallman, Richard, “The GNU Operating System and the Free Software Movement”, p.59, CA: O’Reilly & Associates, Inc., 1999.

his rights as a copyright holder, but exercises his rights in a different way than traditional proprietors.

Authors who want to make their software free cannot just disclaim his rights as the copyright holder and release the work into the public domain, for that would be exposing the work to the danger of being privatized again. Instead, the author has to claim his rights, and with the exclusive rights, he is able to regulate the ways people make use of his work. By licensing his work under GPL, the author would be allowing the users to have the rights pursuant to the free software movement, and also asking the users to take on some responsibilities to keep the software and its derivative works as free as the way he has made his work.

Major terms and conditions of GPL

User's freedoms

When a program is licensed under GPL, besides the access to source code, users are free to

1. Run the program; (sec 0)
2. Make copies of the program; (sec 1)
3. Redistribute the program, even for commercial purposes, provided an appropriate copyright notice and disclaimer of warranty are retained (sec 1). Redistribution in the object code or executable form is also possible, so long as source code is available for all recipients (sec 3); and
4. Prepare and distribute derivative works of the program, provided the derivative works are also licensed to all third parties under GPL. (sec 2)

No Warranty

Though the distribution of the work could be commercial, the work itself is licensed free of charge. Therefore, there is no warranty for the GPL-ed software (Sec 11, 12). The distributor could choose to offer warranty protection in exchange for a fee (sec 1).

License Issued Directly from the Author

The work is not sub-licensable. When a program is redistributed, the recipients still receive the license from the original licensor, and the redistributors may not impose any further restrictions on the recipients' exercise of the rights granted in GPL (sec 6).

Acceptance and Termination

By the act of modifying or distributing the GPL-ed program, a person indicates his acceptance of

the license (sec 5). The license grant is irrevocable, but when the licensee violates the license, the rights granted will be terminated automatically. But those parties who received the program from him, since they all received the license from the original licensor, their rights will not be affected so long as they remain in full compliance with the license (sec 4).

Co-exist with other legal obligations?

The GPL does not concede to any contradictory conditions that are imposed on the recipients. The compliance with the license, if it is not possible as a consequence of a court judgment, allegation of patent infringement or for any other reason, the recipient may not redistribute the program at all (sec 7). A GPL-ed program could not be incorporated into a proprietary program, nor could it be linked with a proprietary library.

The Full GPL text could be found at <http://www.fsf.org/licenses/gpl.txt>

FSF also maintains a very thorough FAQ on GPL which could be accessed at <http://www.fsf.org/licenses/gpl-faq.html>

GNU Lesser General Public License (GNU LGPL or LGPL)

Aside from GPL, GNU project offers a special kind of copyleft license for libraries. The GNU Lesser General Public License (LGPL) permits the LGPL-ed libraries to be linked with proprietary software.

Such an exception may be found in different situations. It might a strategic decision in order to encourage proprietary applications on the GNU system.³³ And for a free library whose features could be largely replaced by other proprietary, releasing under LGPL rather than GPL is to encourage its wider use,³⁴ thus make more improvement for it, and with a larger body of free software users, there would be wider support for free software.³⁵

However, free software advocates still encourage people to use GPL for their libraries rather than LGPL, especially for those libraries that have significant unique capabilities. For people who are interested in utilizing such GPL-ed libraries will have to release their modules as free software too,

³³ Stallman, Richard, "The GNU Operating System and the Free Software Movement", p.63, CA: O'Reilly & Associates, Inc., 1999.

³⁴ Stallman, Richard, "Why you shouldn't use the Library GPL for Your Next Library". Feb 1999. Available from <http://www.fsf.org/licenses/why-not-lgpl.html>, Internet, Accessed 29 May 2004

³⁵ Preamble, the "GNU Lesser General Public License" Available from <http://www.fsf.org/licenses/lgpl.txt>, Internet, Accessed 29 May 2004

and thus resulting more useful modules and program in the free software environment.³⁶

Major terms and conditions of LGPL

The LGPL is identical to GPL in many ways, including the clause that disclaims warranty, that says the license is issued directly from the author, that when the license is applied and terminated, and that its relationship with other legal obligation applied upon users. ,

But on users' rights, LGPL distinguishes two different kinds of situation when one uses a library, a "work based on the Library" means either the Library itself or any derivative work under copyright law (sec 0), while a "work that uses the Library" means a program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it (sec 5).

Works based on the Library

In the case of a "work that uses the Library", i.e. the Library itself and its derivative works, the terms are very similar to those in the ordinary GPL.

User's Freedoms

1. Run the program; (sec 0)
2. Make copies of the program; (sec 1)
3. Redistribute the program, even for commercial purposes, provided an appropriate copyright notice and disclaimer of warranty are retained (sec 1). Redistribution in the object code or executable form is also possible, so long as source code is available for all recipients (sec 4); and
4. Prepare and distribute derivative works of the program, provided the derivative works are also licensed to all third parties under LGPL (sec 2c).

In addition, one may also opt to apply the terms of the ordinary GPL instead of LGPL to a given copy of the LGPL-ed library, especially when one is incorporating part of the code into a program that is not a library (Sec 3).

Works that uses the Library

In the case of a "work that uses the Library", the work itself is NOT subject to LGPL. But when linking the "work that uses the Library" with the Library, an executable which is a derivative work of the Library would be created, and such an executable is covered by LGPL (sec 5).

Such executables, although LGPL allows the author to distribute the object code (sec 5) and license it under terms of his choice, but it is also required that those terms permit modification of the work

³⁶ Stallman, Richard, "Why you shouldn't use the Library GPL for Your Next Library". Feb 1999. Available from <http://www.fsf.org/licenses/why-not-lgpl.html>, Internet, Accessed 29 May 2004

for the customer's own use and reverse engineering. When distributing the executable in its object code, the author has a choice either to distribute the Library together, provided the source code of the Library is made available in those ways similar to the distribution of GPL-ed programs, or not to distribute the Library together but only use a suitable shared library mechanism to link with the Library (sec 6).

By creating such category, the LGPL opens a way for LGPL-ed libraries to be used in proprietary programs.

The Full LGPL text could be found at <http://www.fsf.org/licenses/lgpl.txt>

BSD Style Licenses

Berkeley Software Distribution (BSD) License was first used for Berkeley Software Distribution, a version of Unix developed at the University of California at Berkeley.³⁷ It is easy to follow the BSD license template to create one's own license by changing the values of owner, organization and year appeared in the copyright notice and the license. Unlike the copyleft licenses, the BSD style Licenses are relatively simple and have only limited restrictions on the use of the software.

Users' Freedom

1. Make copies of the program and redistribute the program, either its source code or its binary code. The distributor is not obliged to provide source code.
2. Prepare derivative works and distribute the derivative works, either their source code or binary code. The author is free to choose either FOSS or Proprietary licenses for the derivative works.
3. Incorporate the program into proprietary software.
- 4.

The original BSD License (4-clause BSD) has an advertising clause which is now rescinded. The revised BSD License (3-clause BSD) is very similar to the MIT License, but the MIT License does not have the "no endorsement for derivative works" clause. There is also the 2-clause BSD, which has taken away the endorsement clause and is most similar to the MIT License.³⁸

Multiple Licensing

It is important to note that a work could be licensed under more than one license. A license is a choice of the copyright holder regarding what he thinks of the relationship between his work and possible users. But there might be more than one kind of user, and more than one possible

³⁷ "MIT License Definition", June 2004, Available from <http://www.bellevuelinux.org/mitlicense.html>, Internet, Accessed 1 July 2004

³⁸ "WikiReader, Free Software and Free Content", June 2005, Available from http://en.wikipedia.org/upload/a/a9/WikiReader_Free_Software_and_Free_Contents.pdf, Internet, Accessed 8 July 2004

relationship, the copyright holder is entitled to choose different kinds of licenses for different situations.

Take OpenOffice as an example, OpenOffice is dual-licensed under GPL and Sun Industrial Standards Source License (SISSL). Although OpenOffice states clearly that they encourage the users to use the GPL in order to participate fully in the OpenOffice.org community, the SISSL is provided as an alternative for those developers and companies who are not able to use the GPL.³⁹

Now, source code is open, how about documentations?

GNU Free Documentation License (GNU FDL)

Good documentations and manuals are extremely important for FOSS, but when they are not licensed as free/open as the way FOSS programs are, it is hard for people to make good use of the FOSS programs which they are supposed to illustrate.

Although as a license mainly for software, the GNU GPL itself could also be used on works which are not software, so long as it is defined clearly what the “source code” means when adopting the license/.⁴⁰ The FSF also provides a license specially designed for documentations. The GNU Free Documentation License (GNU FDL or FDL) is a form of copyleft licenses for use on a manual, textbook or other document to assure everyone the effective freedom to copy and redistribute it, with or without modifications, either commercially or non-commercially.⁴¹

By applying GNU FDL to a document, similar to applying GPL to a program, the author is granting users to right to make verbatim copies of the work. Since it is a copyleft license, it requires the copy and distribution of modification of the FDL-ed work is also licensed under FDL. But unlike GPL for software, FDL has special requirements for making large quantity of verbatim copies.

Creative Commons Licenses

Inspired by the FOSS development, Creative Commons Cooperation advocates the openness of digital contents and is urging for a more reasonable and flexible layer for copyright in the face of increasingly restrictive default rules.⁴²

³⁹ “Licenses”, Aug 2002, Available from www.openoffice.org, Internet, Accessed 28 June 2004

⁴⁰ Available from www.gnu.org/licenses/gpl-faq.html#GPLOtherThanSoftware, Internet, Accessed 4 Aug 2004

⁴¹ Available from www.gnu.org/licenses/licenses.html#TOCFDL, Internet, Accessed 4 Aug 2004

⁴² Creative Commons, “Some Rights Reserved”, Building a Layer of Reasonable Copyright”, Available from

In 2002, they released the first version of Creative Commons Public Licenses (CC licenses). By identifying major concerns which authors regard to be important, i.e. whether attribution is required (attribution, BY), whether the users are allowed to make commercial uses of the work (non-commercial, NC), whether the users are allowed to make derivative works (no derivative works, ND), and when derivative works are allowed, whether they are required to be licensed under exactly the same license as the original work (share alike, SA), Creative Commons created 11 different CC licenses. Each represents a unique way of combination of above 4 attributions. Authors are free to choose among the 11 licenses and decide which might best suit his need and his work.

In 2004 Creative Commons released the second version of CC licenses. Since the requirement of attribution has been greatly adopted by users of CC licenses, the attribution requirement has become default, and thus there are only 6 CC licenses in the second version. However, the 11 licenses in the first version are not superceded and are still available.⁴³

CC licenses are designed for all kinds of digital contents but software projects, including art works, photographs, music and literary texts. It does not deal the open source issue, since all the sources in such work are transparent and are not compiled into forms which could not be perceived. Some of the CC licenses does not allow modification and might not be regards as not “free”. However, CC licenses are successful in spreading the idea of freedom and openness to the greater public which might not be familiar with software development.

<http://creativecommons.org/learn/aboutus/>, Internet, Accessed on 4 Aug 2004

⁴³ Creative Commons Public Licenses are available from <http://creativecommons.org/licenses/>, Internet, Accessed 5 Aug 2004

	Authors' Credit is Required	Allow Commercial uses	Allow Derivative Works	Derivative Works should be licensed under the same license as the original work is
CC BY	Yes	Yes	Yes	No
CC BY-NC	Yes	No	Yes	No
CC BY-NC-ND	Yes	No	No	
CC BY-NC-SA	Yes	No	Yes	Yes
CC BY-ND	Yes	Yes	No	
CC By-SA	Yes	Yes	Yes	Yes
CC NC*	No	No	Yes	No
CC NC-ND*	No	No	No	
CC NC-SA*	No	No	Yes	Yes
CC ND*	No	Yes	No	No
CC SA*	No	Yes	Yes	Yes
GNU FDL	Yes	Yes	Yes	Yes

BY: Attribution. For any reuse and distribution, it is required that credit is given to the original author.

NC: Non Commercial. The work cannot be used for commercial purposes.

ND: No Derivative Works. The work cannot be altered, transforms; derivative works built upon this work is not permitted.

SA: Share Alike. It is allow to alter and transform the work, and to prepare derivative works upon

this work, so long as the resulting work is licensed to a license identical to this one.

* Starting 2004, CC licenses has made “attribution” requirement as default in the second version, thus only the first 6 CC licenses above remain in the second version. Those without attribution requirement are no more available in the second version.

Scenarios

For different stakeholders, the use of FOSS might be different from each other. A developer's use of a program might be more intense than an end-user's use, thus a developer's developing activities might be subject to more restrictions than end-user's mere running a program. The following section tries to provide some scenario as examples to explain the different legal issues which might arise in different uses of FOSS.

End user (individual/business/government)

Abul is a public high school teacher. His school couldn't afford the expensive license fee of proprietary office applications. Although proprietary software companies provide special programs for these schools, Abul still wants to seek for an alternative solution to lessen the students' dependence on proprietary software. His friend, Nazlee, is a programmer interested in FOSS development and introduced him a FOSS office application. He and his colleagues downloaded FOSS office solutions and teach students both kind of applications. He was satisfied with the performance and introduced the program to his colleagues. Gradually, the school administrative body also uses the FOSS solution for various administrative works.

In this case, neither Abul (an individual) nor his school (a public government body) made any modification to the solution they downloaded from the web. They were simply end users.

The situation for the end-users is relatively simple. The end-user of a software program might be an individual, a government body, or a business entity. These individual persons or legal entities might have different reasons to use FOSS, some might try to find a cheaper solution or an alternative solution that better suits their needs, some might wish to use FOSS for better customization, some might wish to lessen their dependency on proprietary companies.

Legal Issues Involved

The way end-users use FOSS solutions might not be very different from the way they use proprietary solutions. They download a copy of a FOSS solution or purchase the copy (usually in exchange of some support and services), install it in the computer (thus make a copy on the hard disk again), run the program, and have its functions serve his needs.

The rights concerned here are the right to make copies of the program and the right to run it. (The act of running a program might also be taken as an act of making copy, but is stated differently in some FOSS licenses. For example, the GPL has no restrictions on the running of the GPL-ed program but does regulate the act of copy.) Such rights are all granted by all FOSS licenses, and thus less legal disputes are concerned in the case of end-user.

Though there is less legal disputes in this case, end-users might find some other issues are needed to be taken into their considerations,

Other Considerations- Technical Support

Since an end-user might not be a computer whiz, even when there are existing FOSS solutions, an end-user might be concerned with the technical support of the FOSS solution he needs. Thus instead of simply download a copy of the FOSS solution which saves him some money, he might choose to purchase a box of FOSS solution in a shop where a proprietary solution is also available, and sometimes at approximately the price of the proprietary solution! But the difference is that when purchasing, let's say, a Red Hat Linux, in the store, the end-user is not paying for the license fee, but the service and support. When the term of service is expired, the end-user can choose to pay for another term of service, or ask other available service provider for similar services.

Other Considerations- Customization

When existing FOSS solutions do not fit his needs, the end-user might need to ask individual developers or vendors to make customization for him. In such cases, the end-user, either private or public bodies, might want to have written clauses in the contract to have the vendor or developer take on the whole responsibilities for any possible copyright infringement, and compensate for the possible loss caused by the allegation of infringement. The buyer is free to add these clauses in the contract.

Other Considerations- in Government procurement

However, it has to be noted that in cases of government procurement, since FOSS licenses provides different models to the copyright law, the government should be particularly aware when opens a bid for software solutions or signs a contract with vendors, Existing government bid and contract templates might be drafted under the model of traditional copyright law, and have to be examined, even revised if they fail to treat FOSS and proprietary software equally.

Developer (individual, business)

Developers, (both individual developer and business entities) need to be more careful with the terms and conditions of different licenses for their intensive use of FOSS. Developers might not just run and copy the software, but also create derivative works of the software, and distribute these derivative works together with the original program.

Therefore, for developers, in order to contribute to the development of certain FOSS program, the right to run the program, the right to make copies of the program, the right to distribute the

program and the right to prepare derivative work are all very essential.

The above rights are granted by all FOSS licenses, for these essential rights are considered important both in Free Software Definition and Open Source Definition. Nevertheless, different FOSS licenses might have different restrictions regarding to the practices of such rights, especially on creating and developing derivative works. Developers should pay particular attention on this, and consult their lawyers on their specific situation when needed.

Different considerations might be taken when a developer participates in different stages of software development.

When starting a new project

Abul's colleague Jolly is the school librarian. The school library is not that big, but it also opens to the villagers. In order to keep good records of the books, Jolly sought her friend's help to write a program for her.

Legal Issues Involved- Choose a License of One's Own

Developers: What does this project mean to me and to others? How do I want others to be involved? What do FOSS licenses say? What are the differences between FOSS licenses?

The situation is relatively simple when a developer is starting his new project without using any existing modules, since he would not have to look through the licenses of any existing modules which he might be interested in using.

Starting a new project all over would not be an easy task either, but if that is what a developer chooses to do, then from ground zero, he might be interested in finding a license that matches his needs. The different characteristics of FOSS licenses will have significant influences on the possible development route of the project. The developer is suggested to define what his requirements are before choosing his license.

For example, if the developer is a strong follower of Free Software, he might stick to the GPL or LGPL. If the developer thinks he doesn't need to restrictive people's use of his program, he might think a BSD style license would be enough. Or when the developer thinks it is better if the development could be controlled in a firm and central line, he might not be interested in BSD style licenses. On the contrary, if forking is preferred in the future development, the BSD style license might be a better choice, though there might be the difficulties to merge the forked versions back if they are licensed under incompatible FOSS license.

Developer: Can I change my mind after licensing my project?

Although the copyright owner of a project can always decide to choose another license for his

program when the previous versions have been already licensed under certain FOSS license, the rights of the recipients of the previous versions will not be affected since the license grants are irrevocable. The situation will be more complicated if contributions from the community have been incorporated into the newer version, thus the copyright holder might also include contributors.

Developer: I don't like any of those existing FOSS licenses. Can I start a new one?

Though there are already many FOSS licenses, it is still possible that a developer finds he does not like any of those. A developer is entitled to choose any license for his own project, including a new one he drafted by himself. However, creating a new FOSS license requires particular legal knowledge and skill to avoid vagueness and loopholes. Also, there're already many FOSS licenses and are already causing too much transaction cost in understanding these license. It is not suggested to create a new license unless a developer could not find a suitable one and has strong reasons to do so.

When modifying an existing module

The office application Abul and his school are using is an English one and does not support their language. Although for high school students using English interface might not be too much a problem, it would be difficult, however, when Abul tries to teach the villagers Abul consulted Nazlee, Nazlee has been constantly contributing to the FOSS programs and is also quite familiar with the source code of the office application. She discussed this with a few more friends and as a team they started to localize the application.

Legal Issues Involved: Find out what is the license of the program one is Modifying

When a developer tries to modify an existing module, and when such modification is not solely for his own use but to make further distribution, e.g. when localizing a project, he needs to find out first the license of the module. Thereafter would he know what are the rights granted in the licenses and what is required when exercising those rights.

Developers: Under the license, what are the rights I am granted and what are the restrictions in exercising those rights?

For example, on distribution of a FOSS work, some FOSS licenses (e.g. GPL, LGPL) might require the distributor provide both object code and source code, or at least provide the information on the access of the source code; on modifying a FOSS work, some FOSS licenses (e.g. GPL, LGPL, BSD) may require the modifier to provide documentations on the changes he made; on distributing the derivative work, copyleft licenses require the derivative works to be licensed under the same license as the original work, while other FOSS licenses allows the modifier to choose a

different license (BSD, MIT).⁴⁴ So in Nazlee's case, and let's say they're trying to localize the dual-licensed OpenOffice. If as OpenOffice.org suggest, Nazlee and her friends decide to use the GPL, then as a result, the localized OpenOffice would also be GPL-ed. Some FOSS licenses, e.g. the MIT License, might allow users to sublicense the original work. This means when distributing the verbatim copy of the original work, within the scope granted by the original copyright holder, the distributor may choose a different license and become a licensor himself. In such cases, when a developer creates a derivative work and distributes it together with the original work, he can choose to become a licensor of both works and simplifies the legal relations between the two parties. If sublicense is not allowed, the licensor of the original work would be the original licensor while the licensor of the derivative work would be the developer who prepared the derivative work.

When integrating different FOSS modules into one service

Nazlee works in AA Software Inc. To better oversee the many different projects they're developing, they built a project management system by integrating different FOSS modules. The management system is now only for internal use, but since it is pretty handy, they also plan to make commercial distribution in the future.

This might be a relatively complicated situation and integrating the adopted modules together also creates derivative works of these modules.

Legal Issues Involved: Find out what are the licenses of the programs one is Integrating, and look at the compatibility of these licenses.

In this case, it is essential that AA finds out the licenses of each module. If they happen to be licensed under the same license, e.g. GPL, then it would be relatively simple. Since no matter how these modules are linked with each other, no matter how they might need to modify the modules to integrate them together, the integrated system would still be licensed under GPL. The situation is similar when all modules are licensed under BSD License, but in this case, AA might be able to choose a proprietary license for the modified modules and the integrated system.

But if some of the modules are licensed under GPL, and some of the modules are not, then AA will have to look at the compatibility of different licenses. When two licenses are compatible, the two modules licensed under the two licenses could be combined into a larger work while complying

⁴⁴ The FOSS licenses are many and they might differ with one another in different ways. "A Comparison of Open Source Licenses" provides an analysis of 15 commonly used FOSS license, Available from <http://www.openfoundry.org/en/archives/000388.html>, Accessed 5 July 2004.

with both licenses.⁴⁵ FSF provides a list of GPL-compatible and GPL-incompatible FOSS licenses.⁴⁶

When combining a GPL-ed program and BSD-ed (GPL-compatible) into a larger program, the larger program would be GPL-ed to meet both the requirement of the GPL-ed program and BSD-ed program. When some of the modules are GPL-ed but some of the modules are GPL-incompatible, since integrating them into a larger system might be more than mere aggregation of the modules, the result might be AA must decide which module is more important for them and replace the other one with a module that uses a compatible license.

The licenses used in different modules and the way they are combined together would decide how the integrated system could be licensed and distributed.

Other Considerations: Choice of Law and Choice of Venue Clauses

Finally, for developers who are able to choose licenses for their programs, either that they started their own programs or they are allowed to choose licenses for the derivative works they prepared, they should be aware that many of the OSI-approved licenses are derived from proprietary software companies. Some of these licenses are designed to meet their company policy and strategy, and thus might not be a good choice for the general developers. Some technical issues, e.g. the choice of law and choice of venue clauses (which could be found in Qt Public License, Mozilla Public License, Common Public License, etc.), might be significant in a real lawsuit and need to be taken into consideration.

Vendor/ Producer (business)

Nazlee and her friends have made the localized version of the FOSS office version available. AA Software Inc. is very interested in this application and they have also developed some other small but useful programs for administrative work. They package the localized office together with their own programs (licensed under their proprietary license). The package was a big hit, a few months later, AA also decided to make commercial distribution of the project management system which they integrated from different FOSS modules.

Mere Distribution

⁴⁵ “What does it mean to say that two licenses are compatible?” Available from <http://www.fsf.org/licenses/gpl-faq.html#WhatIsCompatible>, Accessed 7 July 2004; “FAQ on Open Source Licenses”, Available from <http://www.openfoundry.org/en/archives/FAQonOSL.pdf>, Accessed 7 July 2004

⁴⁶ “Various License and Comments About Them”, Available from <http://www.fsf.org/licenses/license-list.html#GPLCompatibleLicenses>, Accessed 7 July 2004

In the former situation, both FOSS programs and proprietary programs are distributed in one package. For the FOSS application, they are merely distributors, and they have to distribute it as its FOSS license requires. For the proprietary programs, AA holds the copyrights and is able to decide the license and the ways of distribution. It is ok to put FOSS applications and proprietary applications into one distribution, e.g. one CD-ROM, if the applications function separately and does not link together to create any derivative work.

Distribution of Integrated Systems

In the case of an integrated system distribution, it is up to the licenses of the different integrated modules and the ways they're combined. As explained above, AA needs to make sure first the licenses of different modules allow AA to combine them together. These licenses will also advice the ways AA could distribute the integrated system.

FOSS is also used in embedded systems. Many devices, e.g. cell phone, handheld, digital camera, DVD players, now uses FOSS embedded systems. The device manufactures use FOSS to lower down their cost when developing new products but not distribute the FOSS itself.

Government Sponsored Projects

FOSS movement and the rapid FOSS development have been getting much attention not just from the FOSS community, but also from academic and policy makers. In some Asian countries, governments work with PC manufactures/vendors to provide affordable PCs bundled with FOSS operating system and office application.⁴⁷⁴⁸ These efforts cut down the prices of owning a PC successfully and made proprietary software giant Microsoft cut down their prices.⁴⁹⁵⁰ Governments also back up FOSS, generate FOSS related projects and promote FOSS as a national technology policy.⁵¹ Some government academic institutes might have started to work on FOSS related projects far before the governments started to notice the potential of FOSS and made clear position on it.

On the FSF-maintained FAQ list about GPL, there are also questions whether US Government

⁴⁷ "Malaysian 'People's PC'- Microsoft experience "Thailand Linux" pain all over again", Mar 2004, Available from http://www.asiaosc.org/article_191.html, Accessed on 7 July 2004.

⁴⁸ Koanantakool, Thaweesak, "A Case for Nation-wide PC Distribution", Nov 2003, Available from <http://www.asia-oss.org/>, Accessed 7 July 2004

⁴⁹ Chai, Winston, "Microsoft cuts prices in Thailand", June 2003, Available from http://news.com.com/2100-1012_3-1019067.html, Accessed on 7 July 2004

⁵⁰ Chai, Winston, "Microsoft Cuts Prices for Malaysia", Mar 2004, Available from http://news.com.com/Microsoft+cuts+prices+for+Malaysia/2100-1016_3-5168029.html?tag=st.rc.targ_mb, Accessed on 7 July 2004

⁵¹ Related links available from <http://uwstudent.org/wiki/OpenSourceInGovernment>, Accessed 8 July 2004; and also available from http://www.asiaosc.org/enwiki/page/Ideas_for_OSS_policy.html, Accessed 8 July 2004

could release a program under GPL or release improvements to a GPL-ed program.⁵² Situations might differ from country to country and from case to case under different government regulations in different countries. Most government regulations on government sponsored projects are drafted under the current structure of national and international intellectual property law and might be more national economically protective and unfamiliar, or even unfriendly to FOSS licensing and developing models.

Following are two cases on government funded FOSS studies. The first one is FOSS related studies made in government research institute without related government policy, while the second one is about the national FOSS project itself.

Government Funded FOSS Projects- Cases from Asia-Pacific

FOSS Project under Government-affiliated Research Institute— Multi-Lingual Editor, Japan⁵³

Emacs is a multilingual text editor first developed by Richard Stallman in MIT. After the GNU project started in 1984, the development of GNU Emacs started and first released in 1985. The GNU Emacs was released under GPL.

The Japanese governmental research institute, Electrotechnical Laboratory (ETL) started to work on the multilingual information processing and integration of GNU Emacs and Mule (multilingual text editor based on Emacs and later merged into GNU Emacs as MULE) in mid 1990s, but there were various issues on copyright. The developer ETL was a governmental research institute, and the licensing model in GPL is very different from the copyright law, so no one was able to decide whether ETL could assign the code to FSF and release the code under GPL. As a result, ETL never officially release the code, but expediently, they released the trial versions. More negotiation between ETL and FSF was done later, and came out with a special agreement. FSF agreed not to require ETL to assign the copyright of the modified code to FSF, and ETL agreed to grant FSF the right to use the code. This was the first time that part of the code in Emacs did not belong to FSF. In 2001 the ETL was reorganized into National Institute of Advanced Industrial Science and Technology (AIST). Although AIST is still a government funded institute, it is an independent organization, and AIST's assets are not national properties. It seemed that AIST would be able to release the code under GPL officially since then, but at first it was still very difficult to make the higher levels of AIST to make a final decision. So it took another year of internal negotiation to

⁵² Available from <http://www.fsf.org/licenses/gpl-faq.html#GPLUSGov> and <http://www.fsf.org/licenses/gpl-faq.html#GPLUSGovAdd>, Accessed on 10 July

⁵³ Handa, Kenichi, "Development of Multi-Lingual Editor", 2003, Available from <http://www.asia-oss.org/nov2003/present/handa/handa.html>, Accessed on 10 July

have them decide AIST is entitled to release their works and choose the licenses of the works. It was not easy to convince people about the advantage of adopting GPL, and according to Dr. Kenichi Handa, a senior researcher in AIST, it is never clear what the main reasons were that have made the high levels to come up with this final decision.

This happened before the Japanese Government formed a clear position on FOSS development. On an open source conference among Asian countries where Dr. Handa was invited to give the development of Emacs, Shuichi Tashiro, the leader of Japanese FOSS project under Ministry of Economic, Trade and Industry, added that the Japanese Government has made necessary regulatory revisions to give the developer of government-funded projects the copyright (and thus the right to choose license), so long as the law is applied from the beginning of the project.

National FOSS Project – Taiwan

Under pressures from the congress, the Taiwanese Government started the planning of national FOSS project in 2002 and started to allocated budget on a 5-year FOSS projects. Ministry of Economics (MoE) was assigned to structure, sponsor and oversee the sub-projects.

Under general government regulations, even the results could be copyrighted by the entity that carried out the government-funded projects, the applications of the results of projects are subject to certain principles. So unless otherwise would be more beneficial for the (national) development of science and technology, the results have to be 1) licensed for a fee, 2) licensed to Taiwanese institutes or firms, 3) used or manufactured within Taiwanese jurisdiction.

Though the exceptions might be made for FOSS projects, while the national FOSS development is assigned to MoE, which bears a more important task in protecting national interests and national economic competency, the more protective/restrictive regulations are applied to national FOSS projects. Under MoE regulations, only the third principle (used and manufactured within Taiwanese jurisdiction) could be made as an exception. Such principles are not in the line with the FOSS development model, and thus made it difficult for the sub-projects under general FOSS project to release their code.

Since the task of FOSS projects would be hindered under such regulation, questions regarding this issue have been raised in the first year of the 5-year FOSS project. Different government bodies related to FOSS projects met up several times to find a solution, but while FOSS licensing model is alien to the models they are used to, the problem have not yet been resolved in the second year (2004), and thus the code developed in the first year is not able to be officially released.

The most recent negotiation was held in May 2004, where different government bodies came out a conclusion. MoE would be submit a case to the Administrative Yuan (highest administration body) on applying the general rules to FOSS projects, seeing for official interpretation from the Government that FOSS meets the exception clause and is exempt from the principles. In the mean while, MoE would be looking to the possibility on the revision of MoE regulations. And while some code-generating FOSS projects are assigned to National Science Council and are applied to the general rules, hopefully the FOSS projects will be able to release the code under FOSS licenses this year.

Online Legal Resources and Materials

- Free Software Foundation, <http://fsf.org>
- Open Source Initiative, <http://opensource.org/>
- Open Source License Law Resource Center,
<http://www.denniskennedy.com/opensourcelaw.htm>
- Open Source Licensing, <http://www.anu.edu.au/people/Roger.Clarke/EC/OSLic.html>
- WikiReader, Free Software and Free Content,
http://en.wikipedia.org/upload/a/a9/WikiReader_Free_Software_and_Free_Contents.pdf
- Groklaw, <http://www.groklaw.net/index.php>
- FLOSS Concept Booklet, http://wikibooks.org/wiki/FLOSS_Concept_Booklet
- Frequently Asked Questions about GNU GPL, <http://www.gnu.org/licenses/gpl-faq.html>
- Quiz to Test Your Knowledge of the GPL and LGPL, <http://www.gnu.org/cgi-bin/license-quiz.cgi>
- Apache License and Distribution FAQ, <http://www.apache.org/foundation/licence-FAQ.html>
- Mozilla Relicensing FAQ, <http://www.mozilla.org/MPL/relicensing-faq.html>
- Netscape Public License FAQ, <http://www.mozilla.org/MPL/FAQ.html>
- Feature, Open Source Families and Facts,
<http://www.unixreview.com/documents/s=8925/ur0312b/>
- FAQ on Open Source Licenses, <http://www.openfoundry.org/en/archives/FAQonOSL.pdf>,
- A Comparison of Open Source Licenses, <http://www.openfoundry.org/en/archives/000388.html>
- Electric Frontier Foundation, <http://www.eff.org/>
- Foundation for a Free Information Infrastructure, <http://www.ffii.org/>
- IP Justice, Campaign for an Open Digital Environment, <http://ipjustice.org/CODE/>
- League for Programming Freedom, <http://lpf.ai.mit.edu/>
- Infochange - Intellectual Property Rights,
http://www.infochangeindia.org/Intellectual_Property_Rights.jsp
- History of Copyright - A Chronology, <http://www.musicjournal.org/01copyright.html>
- Copyrights and Software Protections by Patents and Copyrights,
<http://www.ladas.com/Patents/SoftwareProtectionIndex.html>
- Journal of Information, Law and Technology, <http://elj.warwick.ac.uk/Jilt/>

Glossary

Copyleft

Proposed by free software advocates, copyleft is an alternative idea to the present image of human creative activities defined in copyright law. The copyright law usually confers exclusive rights to copyright holders and thus limits all others' access to the work. Authors might want to "copyleft" his works to grant certain rights to people who are interested in using his works, provided these people would also "copyleft" all the works they created based on these works. Although copyright and copyleft might represent very different ideas of the relationship between authors and their works, copyleft, is not against copyright law. On the contrary, without the rights granted by the copyright law, authors will not have the power to copyleft their works.

Copyright

A bundle of rights regarding to the use of an original expression (including literary works, music composition, movies, paintings, software, etc) granted exclusively to the holder. Copyright is applied to a work upon its creation. Except for the limitation set by copyright law, any use of a work without copyright holder's consent is regarded as illegal infringement. Please note that copyright law protects only the particular way of expression of ideas, not the ideas themselves.

Copyright holder

The individual or legal entity who is entitled to the exclusive rights under copyright law. It is usually said that copyright law is to protect authors of creative works, but most of the rights protected are treated as property rights and could be transferred. Many copyright holders are not authors of the works themselves but their employers.

Derivative work

Copyright law is applied to every work once it is created. When agreed by the copyright holder, one can base on this (original) work to create derivative works. For example, a newer version of a program might contain all or part of the code in its earlier version, thus the newer version is a derivative work of the earlier version. Translation of a document is also regarded as a form of derivative works.

Distribution/ Redistribution

Distribution of the copies of a work is also an exclusive right granted to the copyright holder. In FOSS licenses, all receivers of copies of program are allowed to make further distributions. The term redistribution may be used when emphasizing the distributor has received the program from somewhere and is distributing further.

Fair Use

Copyright law is about a balance between private and public interests. “Fair use” is developed to limit copyright protections and to provide a greater access to the works to the general public.

When a work is used without seeking consent from the copyright holder for purposes of criticism, comments, news reporting, teaching, scholarship or research, it might not be considered as an infringement. The following factors would be considered by the court in deciding whether a case falls in fair use or infringement,

- (1) the purpose and character of the use, including whether such use is of a commercial nature or is for nonprofit educational purposes;
- (2) the nature of the copyrighted work;
- (3) the amount and substantiality of the portion used in relation to the copyrighted work as a whole; and
- (4) the effect of the use upon the potential market for or value of the copyrighted work.

License

Copyright or patent holders usually require users to accept the terms and conditions of a license as a prerequisite to allow their uses of the copyrighted works.

Multiple Licensing

The copyright holder of a work could have various ways to make use this work. The terms and conditions he would want users to accept might differ from case to case. For example, the copyright holder of an Editor might be willing to issue academic license which is cheaper and more affordable for students, while commercial licenses are adopted when selling the program to commercial entities. A copyright holder can also decide to license a work under both FOSS licenses and proprietary licenses to achieve different purposes.

Public Domain

Public domain is a pool that consists of creative works that are not protected by the copyright law and could be used freely. Works fall into the category might be cultural heritage that came to existence before the copyright law, or works that were once protected but copyright law but has expired its terms of protection, or works that its copyright holder decides not to claim copyrights. In the latter case, such disclaimer must be made explicitly.

Source Code

Source Code is written in a special kind of language designed for programming. A program in its source code form might not be easy for lay persons to understand, but it is perceivable for trained

programmers. When the source code is converted to machine readable form, programmers would have difficulties to understand and to modify the program. Therefore, access to source code is a prerequisite for the development of FOSS and a principle embraced in all FOSS licenses.

More explanation of “source code” could be found in the Glossary of the introductory primer, Free/Open Source Software, A General Introduction, which is available from, http://www.iosn.net/downloads/foss_primer_current.pdf

Sublicense

When a copyright holder license his work to someone else, he could also choose to let the licensee sublicense his work, i.e., when the licensee distribute the work, within the scope of rights granted by the licensor, the licensee is not only a (re)distributor, but can also become a licensor of a sublicense between him and the other party (licensee of a sublicense).

However, most FOSS licenses do not grant people to the right to sublicense. For example, A is the copyright holder of X program. B receives a copy of X and distributes more copies. C receives the copy from B. If A does not grant B the right to sublicense, both B and C receive the license directly from A. If A grants the right to sublicense program X, within the scope of the rights granted by A, B may start a new license, and himself become a (sub)licensor of program X.

Warranty Disclaimer

Warranty is a guarantee of the liability of a product. In all FOSS licenses, a warranty disclaimer can always be found. Such clauses are designed to protect the author of FOSS programs, for these programs are licensed without royalty and changes might be added in its development. Though the FOSS programs themselves and royalty-free and disclaim warranty, nevertheless, vendors of FOSS programs could always provide its customers warranty and various kinds of supports with a fee.