The strategic implementation of Free Software in business

This course was developed by:
Shane Martin Coughlan and Ciarán O'Riordan from Free Software Foundation Europe

Shane can be contacted at <coughlan@fsfeurope.org>
Ciarán can be contacted at <ciaran@fsfe.org>

Copyright © 2008 Free Software Foundation Europe except where stated otherwise. Verbatim copying and distribution of this entire document is permitted in any medium, provided this notice is preserved.

Version 5

Table of Contents

Disclaimer

This document and the associated course do not constitute legal advice. None of the information given is intended to be interpreted as legal guidance and/or legally binding.

If you require legal advice you should consult a qualified legal expert.

Free Software Foundation Europe does not give legal advice and does not provide legal services to individuals, projects or businesses. However, we do maintain an informal legal network and we might be able to suggest qualified legal advisers in your country of origin and business.

Introduction

The course we are about to begin is intended to run through most of the topical areas of Free Software licensing and business concerns. While we cannot promise to address all of your questions inside the structure of this course itself, please don't hesitate to ask questions and introduce new topics of conversation.

If you look through this booklet you will notice that all of the primary topic areas are awaiting your input. We are going to foster conversation around the issues presented and work to ensure that everyone comes away with a clear idea of what Free Software can mean in the context of their business, project or area of interest.

Free Software is not a difficult field. It is simply one that requires a little immersion to facilitate full understanding.

What will you get from this course?

This course will explain the types of licence used in the Free Software arena. It will examine the legal questions involved in projects with multiple participants, describing what possibilities and dangers exist in mixing code from different sources. We will cover dual-licensing models, where code is released both freely for community use and in a closed propriety form for specific markets, and best-practice models for developers and projects that adopt Free Software. In addition to this, we will examine how Free Software licensing can be used to produce commercial products within the context of existing business models.

What is the target group of this course?

The course is of benefit to developers, technicians, project managers and coordinators who wish to become more knowledgeable regarding code distribution options.

What previous knowledge is required?

Some knowledge of software development would be useful but is not required. We will introduce and explain the concepts used at each stage.

Intro: Perspectives

What is Free Software?

Free Software is a pretty simple concept. Free Software is software that you can:

- Use
- Study
- Share
- Improve

Software that does not give users these permissions is called “Non-Free Software” or “proprietary software”. The latter name is a reference to the software existing to benefit the proprietor.

How is Free Software different to proprietary software?

You might be able to use proprietary software, but it is fairly certain you will not be able to modify, share and improve it. The licence applying to proprietary software does not provide that type of grant.

Can businesses use Free Software?

Yes. Businesses can use Free Software according to the terms of its licence.

Can products be created using Free Software?

Yes. You can sell software products that contain Free Software. You just have to ensure you follow the terms of the licence.

Case studies: applications

Some well known Free Software:

- GNU/Linux operating system
- Mozilla Firefox
- OpenOffice.org
- Apache web server

Free Software is used in quite a lot of ICT services. Indeed, a lot of the software infrastructure of the Internet is Free Software.

Case studies: embedded devices

Embedded devices are everywhere. Phones, wireless routers, cars.

A lot of embedded devices use Free Software. For example, many use the GNU/Linux operating system. GNU/Linux is very powerful software that you can also find on massive mainframes, super computers and desktops.

Terminology

The operating system shipped by Debian, Red Hat, Ubuntu, Mandriva, etc. are a combination of the GNU operating system, the Linux kernel, and other third party software. We call these systems "*GNU/Linux*". Some people call them "*Linux*", but this misrepresents the operating system. Whichever term you use, the important issue is that they refer to the same operating system.

Similarly, while "*Free Software*" is the original, and in out opinion best, name for the topic of this course, some people call it "*open-source software*". We ask people to prefer the original term, but again, contrary to the controversies that some journalists publish, these are two names for practically the same thing.

Richard Stallman, founder of Free Software Foundation:

> "*Nearly all open source software is free software; the two terms describe almost the same category of software*."

Bruce Perens, co-founder of Open Source Initiative:

> "*My intent has always been for Open Source to simply be another way of talking about Free Software*"

Status of Free Software usage

Who uses Free Software?

The New York Stock Exchange run their mission critical system on GNU/Linux.

NASA have been sending Free Software into space since 1990.

Google – their data centres and web servers.

Wikipedia – the world's 8th most visited website uses 100% Free Software.

The City of Munich's 14,000 desktop computers.

The French Parliament, the French Police.

You?

Who has Free Software based revenue models?

Red Hat

IBM

Sun Microsystems

MySQL

Trolltech

Computer distributors such as Dell, HP, Lenovo, Asus.

Copyleft – the share-alike licences

What types of licence exist?

The common ground of all Free Software licences is that they allow the recipient to use, share, modify and improve the software. The biggest differentiator between Free Software licences is whether or not they have a share-alike clause, called “Copyleft”.

How does Copyleft work?

No Free Software licence requires you to share your software. If you want to keep your software to yourself, that's fine. However, *if you chose to share your software*, Copyleft licences require you to give the recipients all the rights that you got when you received the software.

What licences use Copyleft?

Quite a few licences use a Copyleft provision including the most widely used Free Software licence, the GNU GPL.

How can Copyleft be applied to projects?

A project simply has to adopt a Copyleft Free Software licence and distribute code under its terms.

Can that be compatible with business?

Copyleft projects attract significantly more commercial contribution and support than non-Copyleft projects. This benefits businesses as well as other stakeholders.

Why is that?

The key value proposition in the ICT field lies in the combination of technology, brand recognition and support provided by a product vendor. Copyleft does not compromise this proposition.

With non-Copyleft Free Software project, each contributing company can keep their improvement proprietary. Thinking on in an individual way, it doesn't make sense for any of them to.

Case study: GNU/Linux and the BSD derivatives

FreeBSD is an operating system that is similar to GNU/Linux. The technical aspects of the two systems are roughly equal. One difference is that FreeBSD does not use a Copyleft licence. Their licence gives you the code and puts no requirements on you to publish your changes. Thus, their licence allows companies to apply the business models of the 80s and 90s i.e. they can add features, and they will be the only company with that feature, and they can charge per copy.

From time to time, companies have invested in FreeBSD, but each company keeps all or most of their improvements proprietary. The FreeBSD operating system does not benefit from all the improvements. Companies don't want to put their code under the permissive licence because they know that their competitors could add a further feature and not contribute that back, and then they would be competing against their own work.

With Copyleft licences, everyone has to play fair. No company can exploit the code in a way that others cannot equally exploit it. Thus companies feel secure in contributing their code and the software benefits from each contributor.

For users, this is a big win because instead of having multiple options of exclusive feature sets, they get to use the combination of all added features.

Case study: Licence switching

Not only did GNU GPL'd projects flourish, but we have also seen projects which originally wrote their own licences but which are later switching to the GNU GPL. The Mozilla web-browser suite and the Qt toolkit are two examples.

Business adoption

How can Free Software licensing fit into your business model?

Free Software can help your project or company benefit from an already substantial and continually growing ICT code base. As long as your business model is adapted to take into account licensing expectations and a small amount of retraining for key personnel, Free Software is likely to boost your overall value proposition.

How can products be differentiated from their code?

By the realisation that products are a package consisting of features, support and the trust associated with the brand of distribution.

Does Free Software mean free products?

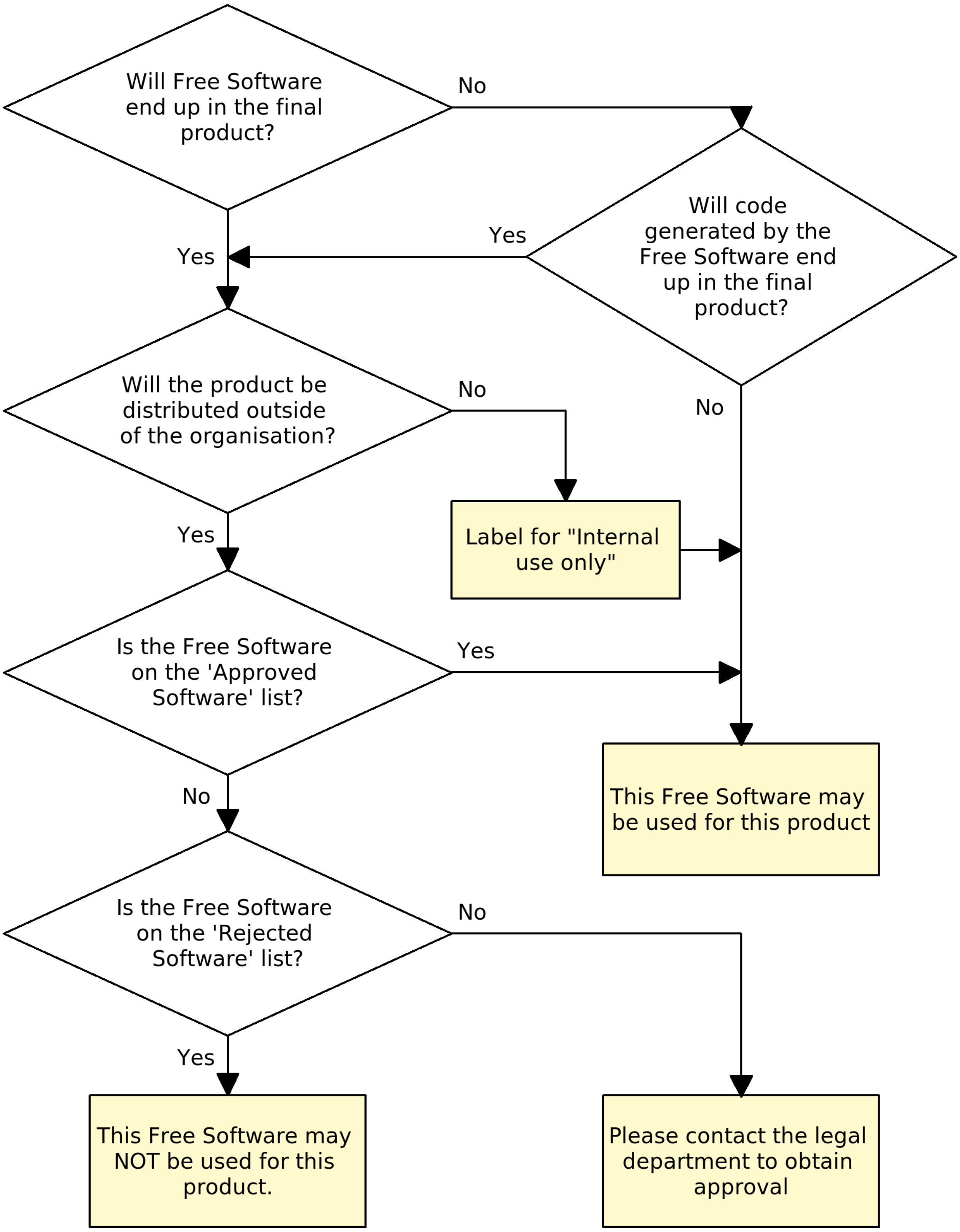
No. You can sell products made with Free Software.

How can Free Software be sold?

The terms of Free Software licences apply to the distribution of Free Software products. These terms do not include any provisions that insist on distribution for zero cost. Many companies sell Free Software. Red Hat is a good example.

Approval flow diagram

Free Software Approval Flowchart



Will Free Software end up in the final product?
No
Will code generated by the Free Software end up in the final product?
Yes
Yes
Will the product be distributed outside of the organisation?
No
No
Label for "Internal use only"
Yes
Is the Free Software on the 'Approved Software' list?
Yes
This Free Software may be used for this product
No
Is the Free Software on the 'Rejected Software' list?
No
Yes
This Free Software may NOT be used for this product.
Please contact the legal department to obtain approval

With thanks to Royal Philips Electronics

Project management

How are Free Software projects different to proprietary projects?

Free Software projects are build on the development and QA work of multiple parties. Proprietary products build on closed platforms are developed by one or few vendors.

How can communities be combined with project coherency?

Free Software is frequently developed by 'communities' of programmers, users, designers and debugging volunteers. The question often arises about how to ensure that these communities do not lead to commercial product fragmentation.

However, a community helps to develop a product. It does not mean that the product itself or its development and release roadmap is fragmented.

How can a project avoid legal problems?

By following 'best practice' processes developed by leading Free Software projects like KDE, GNOME, Debian and Red Hat's Fedora.

Licensing choices

How can you choose the right Free Software licence?

You need to define precisely what your project or product goals are, what your project or company expectations are, and to take into consideration the current and emerging market norms.

How can you apply a Free Software licence to your project?

If you are starting a project from scratch, you can simply select any licence you wish and apply it to your new code. The licence itself will explain how to do this.

If you are using someone else's code as the basis for your project, you need to ensure that you do not contravene the terms of their licence by attempting to apply incompatible terms to your finished project.

Dual licensing

What is dual-licensing?

Dual-licensing refers to a model where code is distributed under two different (and potentially incompatible) sets of conditions.

How can dual-licensing benefit a project?

Dual-licensing can allow a project to benefit from two different types of value proposition at the same time.

What problems are associated with dual-licensing?

Dual-licensing is frequently frowned on by Free Software developers, and it also introduces the potential for confusion on the part of customers wishing to use a product.

When should you choose dual-licensing?

This depends entirely on your project, product and expectations.

Case study: MySQL

MySQL is both the name of a company and of the database technology the MySQL company delivers to the marketplace. MySQL database products are available under the terms of the GNU GPL licence or a proprietary licence.

The advantage of this model is that MySQL benefits from some external development and QA potential, which the disadvantage is that many potential developers are alienated by the thought of the MySQL company distributing code under proprietary terms.

Case study: QT

QT is a windowing and development framework for graphical applications. It forms the basis of the popular KDE Free Software desktop. This framework is available under the terms of the GNU GPL and under proprietary licensing terms.

The advantages and disadvantages faced by Trolltech, the originators of the QT platform, are similar to MySQL.

Consolidation / copyright assignment

One thing that can help with legal maintainability is something called copyright assignment. Because Free Software is often developed by large groups of people ranging from individuals to organisations, the copyright holders of a code base can become fragmented, making it hard to maintain the code base in legal terms.

Copyright consolidation helps resolve this. Even the issue of different legal systems in different countries (like common and civil law) can be overcome with suitably drafted assignment documents.

How can you manage copyright assignment?

As with all issues concerning rights and the transfer of rights, you would be well advised to seek legal counsel.

Copyright assignment documents are available relatively freely on the Internet. For instance, FSFE has a copyright assignment document called the 'Fiduciary Licence Agreement' or FLA. This is a copyright assignment designed to explicitly work in common and civil legal systems.

How can a project ensure legal maintainability?

Having many participants in a project potentially increases the chance of legal maintainability problems. Projects can reduce these risks by maintaining clear copyright attribution and having clear policies for contributions, trademarks and associated areas of development process and branding. It is also worth considering copyright consolidation in certain situations.

Trademarks and commercialisation

How can value be built using a brand?

The value proposition of modern ICT products lies in a combination of the technology, brand and associated support services. A good brand encompasses the user interaction with a technological solution.

Case study: Red Hat Enterprise Linux

The Red Hat value proposition lies in providing enterprise operating systems and full-stack solutions with high quality support services.

People do not buy Red Hat solutions because of the functionality. They buy Red Hat solutions because the functionality delivered comes with enterprise quality support.

Case study: Mozilla Firefox

Mozilla Firefox is a serious competitor to Microsoft's Internet Explorer. Mozilla deliver their product for free (as in price) to end users, while deriving profit from lucrative search content delivery deals with Google. Thus Mozilla leverage their brand as a quality product with zero cost to users while they leverage their user-base as a value proposition for Google service delivery.

How can trademarks be used to ensure coherent branding?

Brand value is a key element of the modern ICT business market. Trademarks allow for visible branding on ICT products. By ensuring that you maintain a clear separation between your brand (protected by trademark) and your Free Software code (protected and shared by copyright), you can ensure that you benefit from community development while maintaining a unique value proposition.

How can Free Software licences and trademarks work together?

They do not have to overlap or create a conflict. Free Software licences are copyright law documents. A trademark is a mark consisting of words and/or symbols that represent a business or single product. Products like Mozilla Firefox and Red Hat Enterprise Server use Free Software code to deliver features, and trademark-protected brands to deliver unique identification. A good trademark policy can ensure that all interested parties feel comfortable with this.

Adoption and transition

What are the advantages of Free Software licensing?

Free Software allows projects and businesses to benefit from the widest possible amount of technologies and advancements in a fair and non-restrictive way.

What are the problems associated with Free Software licensing?

Sometimes people assume that Free Software code can be used as if it were in the public domain. However, Free Software is licensed and it is important to ensure that your project meets the terms of the licences under which your adopted code is distributed.

Case studies: Projects like the Linux kernel

The Linux kernel started in 1991 as a project by a Finnish university student. It has become one of the leading operating system kernels in the server marketplace, and is increasingly used on workstations and desktops as the basis of the GNU/Linux operating system.

There have been some licensing maintainability issues with Linux due to a combination of its massively distributed development and some licensing decisions taken early in the project life-cycle.

Case studies: Projects like Java

Java is a programming language created by Sun Microsystems. It is exceptionally popular and powers virtually every mobile device on the market as well as a substantial proportion of web services.

Sun Microsystems has decided to release the Java platform as Free Software under the terms of the GNU GPL licence. This is a good example of a market-leading product transitioning from proprietary to Free Software.

FSFE and FTF

Where can you go for more information?

There is a great deal of information about Free Software available on the Internet.

For general information about Free Software you might want to consider visiting the Free Software Foundation Europe website. It's available in many European languages:
http://www.fsfeurope.org

For licensing specific information about Free Software you might want to consider visiting the Freedom Task Force website:
http://www.fsfeurope.org/ftf

You can also find useful information at the following websites:
http://www.fsf.org
http://www.fsf.org/licensing

Where can you go for help?

The Freedom Task Force can help. We have an informal network of legal experts throughout Europe that currently covers most EU member states plus many East Asian countries, the USA and Australia.

You can contact the FTF through our website or email:
http://www.fsfeurope.org/ftf
ftf@fsfeurope.org

Background reading

Free Software is a big field. We are not going to present you with a huge amount of information in the background reading section of this guide, but we do want to ensure that you have access to some key ideas and documents.

In the following pages you will find detailed explanations of what Free Software is, how Free Software fits into the European marketplace, and you will be introduced to copies of the most famous and most widely used Free Software licences.

An overview of Free Software licensing and the Freedom Task Force

By Shane Martin Coughlan, FTF Coordinator, FSFE

Introduction

Free Software licensing appears to be a formidable subject, encompassing as it does the specialised fields of both technology and law; an initial glance would suggest that two forms of expertise are required to understand it. However, Free Software licensing is less formidable than it appears.

This paper will introduce the reader to software, licensing, Free Software and the emerging Free Software support networks from first principles. By necessity it will be brief. By the conclusion of this paper the reader should have a clearer understanding of Free Software licensing as a concept and why Free Software Foundation Europe (FSFE) established the Freedom Task Force to support individuals, projects and businesses in Europe.

Defining software

The term software was introduced in 1958 by a researcher called John Tukey to describe programs that run on computers.[1] Software is a broad term, encompassing the programs that allow a computer to function, programs that provides system services, and programs that the end-user interacts with. These myriad programs are created through what is called programming – the act of using a computer language to describe mathematical actions – and compiling the resulting code into machine readable programs.

Software is written by one or more programmers and is then distributed to end-users through various channels. The distribution of software is governed by the creator of the code through a licence which the end-user accepts. Software licences tell people how the rights holder of a certain piece of software wishes that software to be used. Such licences can either be restrictive or permissive, giving end-users more or less freedom to do things with the software.

Defining Free Software

One increasingly prominent type of software is called Free Software. Free Software describes all types of computer programs that adhere to a certain doctrine of how end-users should be able to make use of software, namely that users should be able to run, copy, distribute, study, change and improve software.[2] These freedoms are encapsulated

[1] Britannica Online, 2007, '*Computer software*', http://www.britannica.com/eb/article-235920/computer [12th April 2007]

[2] Free Software Foundation, 2007, '*The Free Software Definition*', http://www.fsf.org/licensing/essays/free-sw.html [12th April 2007]

in four defined rights for end-users:

1. The freedom to run the program, for any purpose (freedom 0).
2. The freedom to study how the program works, and adapt it to your needs (freedom 1).
3. The freedom to redistribute copies so you can help your neighbour (freedom 2).
4. The freedom to improve the program, and release your improvements to the public, so that the whole community benefits (freedom 3).[3]

These freedoms are given to end-users through Free Software licences, a type of software licence designed to ensure that the rights holder of the software gives certain permissions to the end-user. There are quite a few Free Software licences, some of which are compatible with each other and others which are not.[i] The Free Software Foundation lists sixty seven software licences that qualify as Free Software licences.[4] Thirty of these licences are compatible with the most widely used Free Software licence of all, the GNU General Public Licence (GNU GPL). According to some sources, the GNU family of licences (GNU GPL and GNU LGPL) are used in between 65.35%[5] and 66.22%[6] of all Free Software projects.

The GNU GPL is a Free Software licence that uses something called Copyleft, a distribution method requiring that software with the four freedoms maintains those freedoms in further distribution of either the program itself or derivative works.[7] This idea facilitates what might be termed sustainable freedom. Not all Free Software licences ensure freedom is sustained through subsequent distributions of the program in question.

The GNU GPL has other features like a limited warranty and disclaimer of liability clause, though such features are also common in other areas of software licensing.[8] Less common is the preamble of licence, which contains a clear rationale for the use of the licence as a tool in providing freedoms to end-users. The GNU GPL and its sister licence, the GNU LGPL, are tools to enable the transfer of the four freedoms of software to each user without additional limitation or restriction.

Free Software in the real world

Free Software is not just as abstract concept or a distant ideal. Since the launch of the GNU project in 1984, Free Software has grown continually and is now positioned as a

[3] Ibid.

[4] Ibid.

[5] Wheeler, D., 2002, '*More than a gigabuck: Estimating gnu/linux's size*', http://www.dwheeler.com/sloc/redhat71-v1/redhat71sloc.html, [12th April 2007]

[6] Open Source Technology Group, 2007, *SourceForge Software Map*, http://sourceforge.net/softwaremap/ [12th April 2007]

[7] Free Software Foundation, 2007, '*What is Copyleft?*', http://www.fsf.org/licensing/essays/copyleft.html [12th April 2007]

[8] L. Guibault and O. van Daalen, 2006, '*Unravelling the Myth around Open Source Licences*', p.83, T.M.C. Asser Press, The Netherlands.

critical part of academic, public and commercial ICT infrastructure. Major companies have invested heavily in Free Software, including IBM, Oracle, Google and Sun Microsystems.

There are numerous reasons for this growing interest. Free Software allows developers to build on mature platforms and removes the need to reinvent the wheel when creating a new product or solution, thus fostering rapid evolution. It allows everyone to benefit from the power or utility of software, and improvements to the code are returned to the commons, sustaining benefits in the long-term. Free Software also appears to inherently encourage high reliability; a study in 1995 underscored this by confirming that Free Software was less likely to crash than non-Free software when fed random strings of information.[9]

Growing awareness of the advantages of Free Software in the general ICT marketplace has grown to such a level that non-Free software vendors have resorted to media campaigns in an attempt to discredit Free Software projects.[ii] Such actions would appear to be ineffective; according to Netcraft the Free Software web server Apache constituted 58.86% of the total market in April 2007.[10]

Adoption of Free Software has been particularly noticeable in the public sector. In Germany the city of Munich instituted a programme called LiMux to ensure Free Software could be deployed on their systems in the future.[iii] In the UK, Bristol City Council migrated to an office suite based on Free Software,[11] while Birmingham City Council deployed Free Software in libraries with encouraging results.[12]

In the United States, Free Software and the concept of access and sharing it champions is extremely prevalent. When the NSA's Information Assurance Research Group wanted a system to provide security in a wide range of computing environments they decided to extend the GNU/Linux operating system, thus creating SELinux.[13] Likewise, when the State of Massachusetts
searched for a document format that would ensure access without restrictions in the long-term they selected OpenDocument format (ODF), a documentation format born out of the Free Software OpenOffice.org office suite.[14]

[9] Miller, B. P., Koski, D., Lee, C. P., Maganty, V., Murthy, R., Natarajan, A. & Steidl, J., 1995, '*Fuzz revisited: A re-examination of the reliability of unix utilities and services*', http://www.suffritti.it/informatica/tco/fuzz-revisited.pdf [12th April 2007]

[10] Netcraft, 2007, '*April 2007 Web Server Survey*', http://news.netcraft.com/archives/2007/04/02/april_2007_web_server_survey.html [12th April 2007]

[11] Jono Bacon, 2006, '*Bristol Switches to StarOffice*', http://www.linuxdevcenter.com/pub/a/linux/2006/02/23/bristol_migration.html, [12th April 2007]

[12] Birmingham City Council, 2006, '*Opening the door to open source?*', http://www.opensourceacademy.gov.uk/solutions/casestudies/birminham-city-council/file [12th April 2007]

[13] National Security Agency, 2007, '*Security-Enhanced Linux*', http://www.nsa.gov/selinux/ [12th April 2007]

[14] Peter Galli, 2006, '*Mass. Back on Track for ODF Implementation*',

Organisations supporting Free Software licences

Free Software is an important and growing part of the overarching ICT arena. Given that Free Software depends on a specific type of licence for its success, it's worth asking what infrastructure is in place to help support individuals, projects and companies with licence questions.

Just as there are several organisations dedicated to promoting Free Software, there are several projects dedicated to dealing with Free Software licensing issues.

In the United States, Free Software Foundation's Compliance Lab offers licence education and works to ensure that people comply with the terms of the GNU licences. Originally started as an informal activity in 1992, the Compliance Lab was formally established in December 2001 under the management of David Turner. The current FSF Compliance Engineer is Brett Smith.[15] The Software Freedom Law Centre (SFLC) in New York also provides legal representation and other services for Free Software projects.[16]

In Europe, the GPL-Violations.org project, started by Linux kernel developer Harald Welte in January 2004, has engaged with over 100 successful violation resolutions, and continues to promote fair use of Free Software licences.[17] Today the project continues with Mr. Welte acting as project leader, Dr. Till Jaeger representing the project in court, and sviolation detection and analysis being undertaken by Armijn Hemel

However, while GPL-Violations.org continues to resolve compliance issues throughout Europe, there was until recently no organisation dedicated to dealing with licence education and broadening the overall legal infrastructure. For this reason Free Software Foundation Europe launched the Freedom Task Force (FTF) in November 2006.[18]

FSFE's Freedom Task Force

FSFE's Freedom Task Force (FTF) was established to ensure that the legal infrastructure for Free Software in Europe matures coherently and inclusively.[19] It chairs Special Interest Groups, organises discussions and engages with as many parties as possible to ensure mutual understanding.

The FTF wishes to foster dialogue between all parties involved in making use of Free Software. To accomplish this the FTF is building sustainable networks of legal and

http://www.eweek.com/article2/0,1895,2008246,00.asp [12th April 2007]

[15] http://www.fsf.org/licensing/

[16] http://www.softwarefreedom.org/

[17] http://gpl-violations.org/about.html#history

[18] http://www.fsfeurope.org/projects/ftf/

[19] Free Software Foundation Europe, 2007, '*FSFE Launching Freedom Task Force, Co-operating with gpl-violations.org*', http://mail.fsfeurope.org/pipermail/press-release/2006q4/000159.html

technical experts across the continent, and engaging with questions raised by all types of Free Software users.[20] A key aspect of the FTF's work is centred around providing education in Free Software licensing. These activities include answering questions, providing training and helping projects and businesses implement good licensing practices.[21]

The FTF can act as the legal guardian of Free Software projects, consolidating and managing legal affairs while the project itself focuses on developing code. This is done through a copyright assignment known as the Fiduciary Licence Agreement (FLA). The FLA is also available under permissive documentation licences so that projects can customise it and consolidate copyright under their own administrative umbrella.[22]

The FTF works in partnership with the GPL-Violations.org project to resolve Free Software licensing issues. Both projects wish to ensure that issues with code distribution in Europe are resolved for the long-term benefit of the community at large, rather than treated as an isolated incident. This feeds back into the primary goal of the FTF; educating people in the benefits and proper use of Free Software.[23]

In its first six months of operation, the FTF processed over one hundred requests from individuals, projects and companies. It established a list of recommended European Free Software lawyers and built internal networks of legal and technical experts. The fiduciary programme has grown to include Bacula[24] and pyswarm,[25] with discussions under way with several other projects.

The FTF has held talks with embedded device vendors, resulting in the launching of a Far Eastern compliance effort to reduce violations at source. This positive long-term development has occurred in tandem with a mid-term objective; two day licensing courses are now being offered with Internet Academy in Sweden and Digicomp in Switzerland.

Conclusion

Free Software benefits society by allowing people to make the most of technology. The organisations that support Free Software offer a valuable service, not least by working to demystify the emerging digital society. Helping people understand Free Software licences is an important part of this. Free Software licensing need not be formidable. The principles underlying it are straightforward and there is an established body of literature discussing its details.

While Free Software licensing infrastructure is relatively immature in Europe, organisations like the FTF are pro-actively addressing this issue. Sustainable networks that link

[20] http://www.fsfeurope.org/projects/ftf/about.en.html
[21] http://www.fsfeurope.org/projects/ftf/education.en.html
[22] http://www.fsfeurope.org/projects/fla/index.en.html
[23] http://www.fsfeurope.org/projects/ftf/enforcement.en.html
[24] http://bacula.org/
[25] http://pyswarm.sourceforge.net/

competent professionals are being fostered. Indeed, it is increasingly possible to find experts to answer questions that arise, and to offer guidance where necessary.

The definition of Free Software

(a.k.a. "Libre Software" or "Open Source")

By Georg C. F. Greve <greve@fsfeurope.org>

Free in Free Software exclusively refers to freedom, it never refers to price. This fact warrants highlighting because it is at times obscured by a particular weakness of the English language that is generally not shared by other languages. Primarily used in this definition since the 1980s, Free Software is defined by four fundamental freedoms:

1. **The freedom to run the program, for any purpose.**
   Placing restrictions on the use of Free Software, such as time ("30 days trial period", "licence expires January 1st, 2004") purpose ("permission granted for research and non-commercial use") or geographic area ("must not be used in country X") makes a program non-free.
2. **The freedom to study how the program works, and adapt it to your needs.**
   Placing legal or practical restrictions on the comprehension or modification of a program, such as mandatory purchase of special licences, signing of a Non-Disclosure-Agreement (NDA) or - for programming languages that have multiple forms or representation - making the preferred human way of comprehending and editing a program ("source code") inaccessible also makes it proprietary (non-free). Without the freedom to modify a program, people will remain at the mercy of a single vendor.
3. **The freedom to redistribute copies so you can help your neighbour.**
   Software can be copied/distributed at virtually no cost. If you are not allowed to give a program to a person in need, that makes a program non-free. This can be done for a charge, if you so choose.
4. **The freedom to improve the program, and release your improvements to the public, so that the whole community benefits.**
   Not everyone is an equally good programmer in all fields. Some people don't know how to program at all. This freedom allows those who do not have the time or skills to solve a problem to indirectly access the freedom to modify. This can be done for a charge.

Rights, not obligations

These freedoms are rights, not obligations. Any institution and person can choose to not make use of them, but may also choose to make use of all of them. In particular, it should be understood that Free Software does not exclude commercial use. If a program fails to allow commercial use and commercial distribution, it is not Free Software. Indeed a growing number of companies base their business model completely or partially on Free Software, including some of the largest proprietary software vendors. Free Software makes it legal to provide help and assistance, it does not make it mandatory.

Implemented in Copyright

These freedoms are generally implemented by means of copyright, although not in all cases: public domain software is also Free Software, although a special case. For the vast majority of Free Software it is the copyright licence that determines whether a particular program is indeed Free Software. If a copyright licence grants the freedoms described above, it is a Free Software licence, of which between 50 and 150 are in use today.

This surprisingly low number is caused by the tradition to choose established and well-understood licences for Free Software, rather than writing a new licence for every program. Thus, by examining a handful of comparatively simple licences, it is possible to understand the licensing terms of more than 90% of all Free Software, greatly reducing the overhead of licensing administration and compliance.

Terminology

Free Software is discussed under various headings, with alleged antonyms and synonyms, which frequently cause confusion and doubt and will therefore be explained briefly.

Antonyms

The antonym of Free Software is proprietary software, or non-free software. Commercial software is not an antonym to Free Software, being commercial is unrelated to freedom. Commercial Free Software is just as normal as non-commercial proprietary software, sometimes also referred to as "freeware".

Synonyms

As of 1992, the term "Libre Software" was promoted as a synoym to Free Software in parts of Europe to address the particular confusion of the English language. The term "Open Source" was proposed in 1998 as a marketing term for Free Software by the Open Source Initiative (OSI). The OSI definition of "Open Source" covers an identical body of copyright licences to that of the 1989 Free Software Definition explained above.

From the copyright licensing viewpoint, both "Libre Software" and "Open Source" are Free Software synonyms. Combination of terms, such as "FOSS" and "FLOSS" combine

synonyms, redundantly identifying the same body of software.
Ambiguities

The term "Open Source" was occasionally used in various ways before its 1998 definition, and is used in several meanings today, which are often mutually exclusive with each other, and in particular the Open Source Definition of the OSI. So "Open Source" can refer to Free Software, but it can also refer to software not meeting the criteria above. It is also at times used to describe a particular software development model, although some parts of Free Software are developed in closed development models, and proprietary software is increasingly experimenting with open development approaches. This makes the term "Open Source" highly ambiguous, and indeed difficult for all areas that depend on precision in their language, such as science, law and politics.

Public Policy Considerations

Unlike proprietary software, there is never any single company that has absolute control of a Free Software solution. By choosing Free Software, governments protect their independence from the corporate interests of any single vendor, local or foreign. Maintaining their ability to freely and independently live up to their political mandate is the sovereign right of any government. Preferring or mandating Free Software promotes this goal and is always non-discriminatory. It preserves technological and political neutrality because Free Software belongs to no single vendor or organisation, and any vendor is welcome to supply Free Software of third parties. If vendors seek to do business with governments, they can make the decision to give them independence by releasing their own software under a Free Software licence.

Free Software in Europe

By Georg C. F. Greve <greve@fsfeurope.org>

Introduction

Free Software — especially the GNU/Linux operating system — and the FSFE have recently become more visible on the political agenda. This article will seek to explain some of the larger economic, social and political benefits that Free Software offers the European countries and Europe as a whole. It will also give an insight into the work of the FSFE.

As a concept and paradigm, Free Software addresses some of the most fundamental needs of any society in its development towards the post-industrial information era. The most visible organisation in this field, the Free Software Foundation (FSF), was founded in 1985, a time when people had barely begun grasping the most basic principles of information technology.

With the first formal definition of Free Software and the creation of the GNU General Public License (GPL) and GNU Lesser General Public License (LGPL), the FSF not only created (and still maintains) the two most popular licences for Free Software in use today, but also invented the notion of "Copyleft," referring to Free Software protected against being stripped of its freedom.

Free Software itself is defined by four basic freedoms. The first freedom — sometimes referred to as freedom 0 — is the unlimited use of a program for any purpose. This means that a Free Software licence must allow use for all commercial or non-commercial applications in order to fulfil this criterion.

The second freedom — freedom 1 in the Free Software definition[26] — is the freedom to study a program to learn how it works and to adapt it to your own needs. The remaining two freedoms are the freedoms to redistribute unmodified copies and the freedom to release modified copies that improve the state of the art.

As these are freedoms, people are free to choose to exercise one or several of them, but they may also choose to exercise none.

Licences providing these freedoms are referred to as Free Software licences.[27] A special case of Free Software licence, the so-called "Copyleft" licence, has already been mentioned above. These licences give any user the freedoms described above, but they explicitly forbid a distributor to remove that freedom, which would make recipients of such freedom-deprived software dependent on that specific distributor.

Since access to the source code is a necessity to exercise these freedoms for

[26] http://www.gnu.org/philosophy/free-sw.html

[27] http://www.gnu.org/licenses/license-list.html

programming languages with distinct source code, some people suggested using "Open Source" as a marketing term for Free Software in 1998; nowadays Free Software is sometimes referred to under this marketing term.

The good intention of making Free Software more widely known has unfortunately had the unexpected side effect of weakening the distinction between Free and proprietary/non-free software.[28] Therefore the Free Software Foundation strongly recommends speaking about Free Software or the adequate term in the local language; as will be done in the remainder of this article.

Economic perspectives of Free Software

Despite the attempts of proprietary software vendors — especially those located in the United States holding a monopoly in their respective areas — to make it seem so, Free Software is not an attack directed at specific companies.

Free Software should be understood as a new paradigm, a new model of dealing with software based upon mature concepts. It is a model based upon keeping the markets open and freely accessible; as such it cannot be an attack on specific companies, since any company can participate in this new market.

In a Free Software economy, there will be market leaders, but the possibility of uncontrollable monopolies is much lower.

To current monopolies this may seem threatening. But as one of the most important — maybe even the most important — problems of the European IT industry is its dependence on foreign IT monopolies, weakening these monopolies has become necessary for Europe to prosper.

That current monopolistic situation is a logical consequence of the proprietary software model, which has a strong system-inherent tendency towards proprietary software. The reason being that proprietary software tends to only work properly with itself.

With such proprietary software, communication between two users requires that both use the same software. Given that all people in western countries supposedly know each other over no more than five others, this leads to a kind of "viral" effect, where one user forces the next to use the same software, creating a monopoly.

In theory, open standards would provide a way out of this vendor lock-in, but history has shown that no open standard was ever truly successful unless it was implemented in Free Software.

The possibility to enlarge and lock-in a user base by modification of an open standard — a process euphemistically described as "improving" a standard — that in consequence allows only migrating to a certain piece of software, but not away from it, has proven to be

[28] http://fsfeurope.org/documents/whyfs.html

too much of a temptation for the major players in the field.

As the past has proven, it is ineffective to impose open standards on vendors of proprietary software because of the fast-paced development in this sector in combination with the in-transparency of proprietary software and the comparably slow workings of the political decision process.

That is if the vendors accept such measures and do not exert their monopoly-based clout to stop such actions altogether, as recent anti-trust cases in the United States have shown.

Structure of a Free Software economy

The differences are much smaller than many people would make you believe. The financially most important sector today is software for business activities and most of the revenue is generated through service. This is unlikely to change.

It is true that licence revenue will most likely go down, probably significantly. However this only affects a very small part of the software generated revenue; a part which generates a negative trade balance between Europe and the United States today.

The by orders of magnitude largest source of revenue today is service. This sector will be able to grow significantly in a Free Software economy.

In the current system, dominated by proprietary software, only those companies supported by the monopolies can offer services; usually only a small part of what would be possible. The remainder is either done by the monopolies themselves — generating another stream of revenue flowing out of Europe — or not at all.

Free Software offers greater independence of European businesses, allowing them to offer the full array of services if they wish or cooperate with others if this seems economically more useful.

Also they will be able to provide solutions for those services that are already in demand, or that they can create a demand for, which are currently impossible because businesses lack adequate access and control over the software these services depend on.

In a Free Software economy, the current revenue in the service sector will be redistributed more in favour of the European vendors and the sector as a whole can be expected to grow.

Reducing dependencies

It also must be considered that currently the holders of monopolies have control over the European IT industry as they could drive most companies out of business by denying them access to their monopoly or by making access so difficult that the economics of the situation will possibly drive the company out of business.

To further worsen the situation, software monopolies can effectively be coupled with hardware monopolies. So a piece of monopolistic software will run only on a special kind of hardware and in return the vendor(s) of that hardware will only deliver their machines with this particular software.

The Free Software paradigm does not allow building this kind of coupled monopoly. In fact Free Software encourages platform independence and the Free Software systems (e.g. GNU/Linux and the BSD systems) run on more hardware platforms than any proprietary operating system.

Because the freedom to modify allows adding support for other hardware platforms, Free Software provides a stable foundation for innovative hardware initiatives that might even start on a local or regional level.

That way Free Software not only brings back competition into the software, but also furthers it in the hardware field.

National Economy

Because the largest part of software development is putting together old and well-known principles, these get reimplemented at least once by every company, sometimes even once for every project.

In terms of national economy, proprietary software is waste of highly skilled labour. The proprietary software paradigm keeps software developers busy reinventing the wheel, slowing down innovation.

Free Software allows building upon these old and well-known building blocks, consequently reducing the market-entry barrier for new and innovative companies.

Also, the software industry is only one part of economy as a whole. As software is the glue that ties together a digitally networked economy, all sectors pay the price for the inefficiency of the proprietary software model.

Today, most non-IT companies use proprietary solutions. This makes them relying entirely on their vendors for crucial aspects of their own economic activity such as keeping stocks, writing and paying bills or communication with their customers, suppliers and/or competitors.

Forced updates are one result, the need to sometimes replace a whole IT solution, downtimes and new training of employees included, is another. Solutions based upon Free Software remove this dependency almost entirely.

As the company gains the freedoms described above, updates can be made according to the economic situation of the company. In case of problems with the vendor, the solution will still remain usable and another vendor can be found.

In the latter case, an investment for the new vendor to work itself into the solution is required, but that cost is significantly lower than the cost of an entirely new solution. Also the indirect costs in terms of customer dissatisfaction, training of employees and downtimes usually do not arise.

It can be expected that these effects will help revitalising economy as a whole. In essence, Europe can only win economically by furthering massive deployment of Free Software.

Social issues

Access to software becomes increasingly important to participate in the cultural, social and economic exchange of mankind. For the individual this means that access to software determines ones ability to communicate, to study and to work. Studies from the United States indicate that the average person interacts about 150 times each day with software.

In consequence, software has to be understood as a form of cultural property, a cultural technique. As long as mankind exists, new cultural techniques have risen the question of who is given access to them. Free Software ensures all people retain equal access to the cultural property that software has become.

In terms of data security and protection, another issue arises. As computers are always opaque — it is not possible to tell by mechanical observation what a computer does — it becomes even more important that the software is entirely transparent. Otherwise people lose the ability to determine what their computers do and consequently have no control over their personal or other data.

Free Software is by nature entirely transparent, preserving the maximum of informational self-determination.

2001: The Free Software Foundation Europe

Networks tend to be more stable than single nodes and Europe is one of the leading — if not the leading — regions for Free Software. So in 2001, the Free Software Foundation Europe (FSFE) was founded as a sister organisation of the Free Software Foundation in North America. Legally, financially and personally independent of each other, they are working together on all aspects of Free Software in a spirit of equal cooperation.

The FSFE itself encompasses the vision of a strong Europe united in cooperation and mutual understanding with currently four countries (France, Germany, Italy, Sweden) fully represented, three others associated (UK, Portugal, Austria) and several others involved through regular cooperation.

A main function of the FSFE is providing a European competence centre for Free Software, offering advice to governments, commissions, companies, journalists and others.

In the scope of these activities, the FSFE was invited to provide an expert for the

Commission on Intellectual Property Rights in London[29] and presented Free Software at an OECD workshop in Tokyo on invitation of the German Ministry of Economics and Technology.

Other activities involve regular project work, for instance in AGNULA,[30] a project funded in the scope of the 5th framework programme of the European Commission (IST-2001-34879).

For the 6th framework programme, the FSFE issued a recommendation supported by over 50 parties, in which the advantages of Free Software for Europe are addressed in how they refer to accepted European goals; concrete recommendations on how Europe can capitalise on them are given.[31]

Also the FSFE is doing work to support the legal foundation of Free Software, for instance it helped a local institute for legal issues of Free Software, the ifross, with the amendment of a German copyright law revision and recently issued the Fiduciary Licence Agreement (FLA),[32] which will help upholding the legal maintainability of Free Software.

Capitalising on Free Software

Free Software offers unique opportunities for Europe as a region and the European states. In fact Europe is currently the region with the best position to gain the full advantages of Free Software and go into the information age with a head-start.

Possible advantages include greater independence, increased sustainability, freedom from foreign mono- and oligopolies, alternative hard- and software possibilities, a strengthened domestic market and better protection of civil rights.

For these to become reality, it becomes increasingly important to make clear statements and policies in favor of Free Software, such as the evaluation bonus for Free Software projects defined in the IST work programme or the policy statement by Liikanen in the European Parliament[33] regarding Free Software in public administration.

In fact public administration happens to provide an excellent starting point for the transition towards Free Software for three reasons.

Firstly, a government using proprietary software creates a tendency to force its citizens to use the same software because of the aforementioned "viral" effect of proprietary software. As governments have the ethical obligation to be available to all its citizens, they can make a just case for Free Software based upon the consideration of not wanting to force their citizens into a harmful monopoly.

[29] http://www.iprcommission.org

[30] http://fsfeurope.org/projects/agnula/

[31] http://fsfeurope.org/documents/fp6/recommendation.html

[32] http://fsfeurope.org/projects/fla/

[33] http://www3.europarl.eu.int/omk/omnsapir.so/cre?FILE=20021023r&LANGUE=EN&LEVEL=DOC&NUMINT=3-188&LEG=L5

Secondly, public administration is always short of resources, but the majority of resources spent on IT get squandered by creating a separate solution for each ministry or region, while the problems addressed tend to be similar and massive cooperation would be possible.

And finally, use of Free Software in public administration will provide a role model, encouraging citizens and businesses to get out of unhealthy dependencies, getting accustomed to the new model and becoming economically and socially active in it.

Several European regions already have initiatives to make use of Free Software mandatory for public administration. The commission entrusted with this question for the French speaking part of the region of Brussels came out in favor of such a regulation on February 11th, 2003, for instance.

Public administrations in Europe should at least make sure to prefer Free Software over proprietary and require open standards for which a Free Software reference implementation exists.

Also wherever public money is spent, spending it on Free Software is making sure that it will benefit the public and economy. In the past, such money was usually spent on proprietary software, often benefiting only that proprietary vendor company directly at the cost of society and economy as a whole, or getting lost entirely.

For that migration period towards a more sustainable approach, especially the so-called "Copyleft" licences — the GNU General Public License (GPL) being the most widely known — provide a sound basis for such projects.

These licences will make sure that the results of resources spent will be available for all of economy and society equally, fostering a general increase of economic activity. They will resist having the results procured by any single company or person trying to restore old monopolistic situations.

Information Age aware governance

Like information technology permeates all of economy and society, governance decisions in one area can influence chances in the information age significantly. Given the European goal of becoming an information economy, it becomes necessary to be aware of these issues in all areas of governance.

There are several policies pending or in implementation that are about to inflict serious harm on the European competitiveness. These should be prevented or abolished if seeking to increase the European edge.

One policy endangering proprietary and Free Software alike are software patents. Patents are an entirely unsuitable concept for software as it has very different properties. Experience indicates the United States are already paying dearly for their software patent system with reduced innovation.

To quote Bill Gates from an internal memo: "If people had understood how patents would be granted when most of today's ideas were invented and had taken out patents, the industry would be at a complete standstill today. ... The solution is patenting as much as we can. A future start-up with no patents of its own will be forced to pay whatever price the giants choose to impose. That price might be high. Established companies have an interest in excluding future competitors."[34]

Another extraordinarily harmful law is the European Copyright Directive (EUCD). Its US counterpart, the Digital Millennium Copyright Act (DMCA) is already being used successfully by groups such as Scientology to censor unwelcome web sites.[35] Similar cases can be expected in Europe.

Economically, the EUCD is highly anti-competitive. As it makes it illegal to circumvent whatever is considered a protection measure, the company that created this technical measure is given ultimate control over who may or may not participate in the market based upon it or how these companies should behave.

Example is given by the recent case against the teenager Jon Johansen, in which the question whether buying a DVD in a store will entitle the customer to view that DVD on their computer has become the central issue. The EUCD also provides a serious impediment of the freedoms of speech, communication and choice of profession, giving it a somewhat anti-democratic air.

These two policies are either in the process of adoption or adopted already and should be abolished before they can do further harm to Europe's competitive edge.

The current new initiative to reduce competition in the market further are Palladium and its hardware counterpart proposed by the TCPA. This initiative, which wishes to be known as increasing the trustworthiness of computers, is best described as "Treacherous Computing."[36]

Under the pretence of trying to improve computer security, the TCPA apparently seeks to eliminate concepts and paradigms competing with the monopoly holders of the proprietary software model. Again, Europe would be on the losing side.

Résumé

Free Software as a new paradigm offers a stable, lasting and sustainable approach with higher dynamics and increased efficiency. The first region to understand and adopt it on a larger scale is likely to become a leading force in the information age.

[34] http://swpat.ffii.org/archive/quotes/index.en.html

[35] http://www.theregister.co.uk/content/6/24533.html

[36] http://www.counterpane.com/crypto-gram-0208.html#1

Currently it seems unlikely that Free Software will ever replace proprietary software completely, but by making Free Software the predominant model, Europe could relieve dependences on foreign monopolies, which currently create a highly unstable and unfavourable situation for the European information technologies industry.

Europe is right now in the unique situation of having a large supply of Free Software competence and growing network of smaller companies that are based upon or centred in Free Software. Also more of the old and traditional European IT companies have begun shifting at least partially towards Free Software.

If this is furthered now, Europe has the potential to become global leader in the information age.

The GNU GPL version 2

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion

of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the

Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License

may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF

DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

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

This program is free software; you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation; either version 2 of the License, or
(at your option) any later version.

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

You should have received a copy of the GNU General Public License along
with this program; if not, write to the Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.
```

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program
`Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License.

GPLv3: Simplicity and Length

By Ciarán O'Riordan <ciaran@fsfe.org>

Everyone wishes that free software licences were shorter. The good news from the GPLv3 process is that by changing the LGPL from being a whole licence to being an additional permission that can accompany the GPL, the LGPL has shrunk drastically and the proposed GPL and LGPL texts, combined, are shorter than the current GPL and LGPL combined. But GPLv3 itself will indeed be longer than version 2 is.

Additional words are necessary to protect against additional problems, to ensure it works the same across national boundaries, and to clarify the meaning of the licence. Since FSF started soliciting comments in January 2006 about how to improve the GPL, there have been few or no suggestions as to how it could be significantly shortened.

But longer doesn't have to mean more complex. GPLv3 is simpler in some ways, and I'd like to describe how it was made simpler so that maybe some people can see other parts of the licence that those same principles can be applied to.

- Clarity of wording: When something is not defined, the reader will have to check their local laws to see what happens in undefined cases. For example, "distribution" was undefined in GPLv2, but re-labelled as "conveying" in v3 to avoid clashing with existing definitions of "distribution" in local laws, and it has been defined. This should make it easier to comply with the licence.
- Clarity of intent: Similarly, GPLv3 adds words to clarify what is intended to happen in new circumstances, such as described in the recent piece I wrote on GPLv3 and the EUCD/DMCA.
- Labelling: Each section in v3 has a descriptive title, unlike version 2 where each section only had a number. This just makes it easier to find what you're looking for.
- Layout: All related things should be grouped together, and each clause should be in the section you'd most expect it to be in.

The number of words needed to ensure that software users had the four freedoms in the 1970s was zero. There were no software patents, no DMCA/EUCD laws, software generally came with source code, and there was generally nothing limiting a person from redistributing software.

As software distributors started blocking these freedoms by legal and technical means, it became necessary for software that was intended to come with those freedoms to be accompanied by licences granting those freedoms and requiring others to pass them on when they pass on the software.

GPLv1, written in 1989, had 1,500 words. GPLv2 has 2,984 words. The GPLv3 has 5,680 words. The most important implementation detail is that it has to work in court, and this can't be compromised for the sake of making a shorter text. But if you can see ways to make it simpler, that would be very useful because it's not only technology lawyers that have to read the GPL, it's software developers and judges too.

The GNU GPL version 3

GNU GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program--to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code.

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's

System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

a) The work must carry prominent notices stating that you modified it, and giving a relevant date.

b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".

c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.

d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.

b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the

Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or

b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or

c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or

d) Limiting the use for publicity purposes of names of licensors or authors of the material; or

e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or

f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR

DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

> <one line to give the program's name and a brief idea of what it does.>
> Copyright (C) <year> <name of author>
>
> This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.
>
> This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.
>
> You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

```
    <program>  Copyright (C) <year>  <name of author>
    This program comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
    This is free software, and you are welcome to redistribute it
    under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, your program's commands might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary. For more information on this, and how to apply and follow the GNU GPL, see <http://www.gnu.org/licenses/>.

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read <http://www.gnu.org/philosophy/why-not-lgpl.html>.

The Fiduciary Licence Agreement

(Version 1.2)

Copyright (C) 2002-2007 FSFE, e.V.,
Talstrasse 110,
40217 Dusseldorf,
Germany

Everyone is permitted to copy and distribute verbatim copies of this licence document, but changing it is not allowed.

Preamble

Access to software determines participation in a digital society. To secure equal participation in the information age, the Free Software Foundation Europe (FSFE) pursues and is dedicated to the furthering of Free Software, defined by the freedoms to use, study, modify and copy. Independent of the issue of commercial exploitation, it is proprietary, freedom-diminishing licensing that works against the interests of people and society at large, which is therefore rejected by FSFE.

The purpose of this agreement is to ensure the lasting protection of Free Software by making FSFE the fiduciary of the author's interests. It empowers FSFE - and its sister organisations - to uphold the interests of Free Software authors and protect them in court, if necessary.

FSFE is given the right to relicense the software as necessary for the long-term legal maintainability and protection of the software. The agreement also grants the author an unlimited amount of non-exclusive licences by FSFE, which allow using and distributing the program in other projects and under other licences.

The contracting parties sign the following agreement in full consciousness that by the grant of exclusive licence to the Free Software Foundation Europe e.V. and by the administration of these rights the FSFE becomes trustee of the author's interests for the benefit of Free Software.

Agreement

Between (please cross out unsuitable alternatives) the author
____
____ (Occupation, D.o. Birth)
____ (Street)
____ (ZIP, City, Country)
____ (Pseudonym, Email)

the owner of the exclusive licence,
____ (Company or name)
____ (HQ or addr.)
____ (managing director)

acquired by virtue of a contract date as of contracting party:
acquired as employer in the context of a work and service
- hereinafter referred to as "Beneficiary" -
and the
Free Software Foundation Europe, e.V.,
Talstrasse 110
40217 Dsseldorf
Germany
- hereinafter referred to as "FSFE" -
the following agreement is entered into:

1 - Grant

Subject to the provision of Section 2, Beneficiary assigns to FSFE the Copyright in computer programs and other copyrightable material world-wide, or in countries where such an assignment is not possible, grants an exclusive licence, including, interalia:

- the right to reproduce in original or modified form;
- the right to redistribute in original or modified form;
- the right of making available in data networks, in particular via the Internet, as well as by providing downloads, in original or modified form;
- the right to authorize third parties to make derivative works of the Software, or to work on and commit changes or perform this conduct themselves.

Beneficiary's moral or personal rights remain unaffected by this Agreement.

In some countries, the law may provide that the employer is deemed to be the owner of the rights on materials developed by an employee in the course of his or her employment, unless the parties have agreed otherwise. The Beneficiary is aware of these provisions, and therefore warrants, represents and guarantees that the Subject Matter is free of any of his or her employer's exclusive exploitation rights.

2 - Subject Matter

The rights and licences granted in Section 1 are subject to all ``Software'' and ``Documentation.'' For the purpose of this Agreement, ``Software'' shall mean all computer programs, copyrightable sections of computer programs or modifications of computer programs that have been developed or programmed by Beneficiary and that are specified in this Agreement below or that are listed in Exhibit A attached to this Agreement and dated and signed by the contracting parties. Likewise, ``Documentation'' shall refer to all manuals and documentation written by Beneficiary alongside and usually distributed with the ``Software'' and are similarly specified below or listed in Exhibit A:

____

____

____

Except in countries where such an assignment is not possible, the rights [and licences] granted under this agreement by Beneficiary shall also include future developments, future corrections of errors or faults and other future modifications and derivative works of the software that Beneficiary obtains copyright ownership. Excluded from this provision are modifications that are not derived from the subject matter and that have to be regarded as independent and original software.

Section 3 - FSFE's Rights and Re-Transfer of Non-Exclusive Licence

FSFE shall exercise the granted rights and licences in its own name. Furthermore, FSFE shall be authorized to enjoin third parties from using the software and forbid any unlawful or copyright infringing use of the Software, and shall be entitled to enforce all its rights in its own name in and out of court. FSFE shall also be authorized to permit third parties to exercise FSFE's rights in and out of court.

FSFE grants to Beneficiary a non-exclusive, worldwide, perpetual and unrestricted licence in the Software. This right's [and licence's] scope shall encompass and include all the rights [and licences] specified in {\S}~1. Furthermore, FSFE grants to Beneficiary additional non-exclusive, transferable licence to use, reproduce, redistribute and make available the Software as needed for releases of the Software under other licences. This re-transfer shall not limit the scope of FSFE's exclusive licence in the Software and FSFE's rights pursuant to Section 1.

FSFE shall only exercise the granted rights and licences in accordance with the principles of Free Software as defined by the Free Software Foundations. FSFE guarantees to use the rights and licences transferred in strict accordance with the regulations imposed by Free Software licences, including, but not limited to, the GNU General Public Licence (GPL) or the GNU Lesser General Public Licence (LGPL) respectively. In the event FSFE violate the principles of Free Software, all granted rights and licences shall automatically return to the Beneficiary and the licences granted hereunder shall be terminated and expire.

The transfer of the rights and licences specified in Section 1 shall be unrestricted in territory and thus shall apply world-wide and be temporally unlimited.

Section 4 - Miscellaneous

Regarding the succession of rights in this contractual relationship, German law shall apply, unless this Agreement imposes deviating regulations. In case of the Beneficiary's death, the assignment of exclusive rights shall continue with the heirs. In case of more than one heir, all heirs have to exercise their rights through a common authorized person.

Place of jurisdiction for all legal conflicts arising out of or in connection with this Agreement is Munich, Germany.

____________________, ________________ ____________________, ________________

_______________________________________ _______________________________________

(Beneficiary) (FSFE)

[i] Free Software Foundation, 2007, '*Various Licenses and Comments about Them*', http://www.fsf.org/licensing/licenses/ [12th April 2007]

[ii] Microsoft, 2007, '*Get The Facts*', http://www.microsoft.com/windowsserver/facts/default.mspx [12th April 2007]

[iii] Portal München Betriebs-GmbH & Co. KG, 2007, '*LiMux - The IT-Evolution*', http://www.muenchen.de/Rathaus/dir/limux/english/147197/ [12th April 2007]