Copyright in Software and Open Source licensing

OUCS

Rowan Wilson (OSS Watch)

February 2008

Copyright in Software

CAUTION!

- **I am not a lawyer**
- **I am not a management consultant**
- **A lot of material – some of it tedious**

Copyright in Software

- Who are you?
- What is copyright and what does it have to do with software?
- Good practice in managing copyright while developing software
- An Introduction to Free and Open Source Licensing
- Some common FOSS licences
- Exploitation strategies for FOSS-licensed software

Copyright in Software

Copyright in Software

- "**OSS Watch** provides unbiased advice and guidance on the use of free and open source software and licences. OSS Watch is funded by the JISC and its services are available free-of-charge to UK higher and further education. If you want to find out more about open source software, we're the people to ask."
- OSS Watch is based within Oxford University Computing Services
- We are **not** an advocacy group
- We are not lawyers
- http://www.oss-watch.ac.uk

Copyright in Software

What is copyright and what does it have to do with software?

Copyright in Software

Copyright...

- is a form of 'intellectual property'
- is an unregistered right – it comes into existence at the same time that the work is 'fixed'
- protects the 'fixed' form of an idea, not the idea itself
- protects literary and artistic material, music, films, sound recordings and broadcasts, including software and multimedia
- generally does not protect works that are 'insubstantial' – thus names and titles are not protected (although a 'passing off' action may be a possibility)
- gives the author exclusive economic and moral rights over the copyrighted material

Copyright in Software

What exclusive **economic rights** do copyright owners have?

- Making copies
- Issuing copies to the public (publication, performing, broadcasting, online distribution)
- Renting or lending copies
- Adapting the work

Copyright in Software

What exclusive **moral rights** do copyright owners have?

→In the case of software, none. Unlike other creators of literary works, software authors have no statutory protection against derogatory treatment of their code or automatic right to be identified as the author of their code

Copyright in Software

What happens if I'm employed or contracted to write software?

→Your employment contract will govern who owns the copyright, although the default position will be that your employer does

→For University of Oxford employees the position is spelled out in the University Statutes XVI part b

→Even works created outside working hours may be owned by your employer if they are of the same general type as you are employed to create

→Beware! If you bring in contractors or consultants they will by default own the copyright in their work unless the contract you arrange says otherwise

Copyright in Software

When does copyright in software expire under UK law?

→For literary works including software:

• 70 years after the death of the author

→ Calculating copyright expiry is made more complex by the fact that the duration has changed over the last 20 years. Luckily in the case of software its novelty and relatively short shelf-life mitigate this.

Copyright in Software

What can I do with my copyright material?

→Sell it (assign it) – transfer ownership of your rights

→License it – grant use of your rights, possibly for a limited period or within a limited geographical area.

Copyright in Software

A word about patents

→Not at all the same thing

→Generally OSS licensing of code is incompatible with the exploitation of software patents embodied in the code in question

→European Patent Convention 1973 Article 52:

"(1) European patents shall be granted for any inventions which are susceptible of industrial application, which are new and which involve an inventive step.

(2) The following in particular shall not be regarded as inventions within the meaning of paragraph 1:

...

(c) schemes, rules and methods for performing mental acts, playing games or doing business, and ***programs for computers;"***

Copyright in Software

A word about patents

- In fact, over the last 20 years this exclusion has been rendered moot by repeated approval of patents by the EPO and national patent-granting bodies which are, in effect, for software.
- Despite this, there seems to be a general reluctance to litigate in support of these patents in Europe.

Copyright in Software

Good practice in managing copyright while developing software

Copyright in Software

Making sure your code is releasable

- Strongly consider obtaining contributor licence agreements where necessary
- Keep track of your inbound licences and what they oblige you to do (licence compatibility)
- Keep track of the employment/consultancy agreements of contributors, including all institutional regulations that they import
- Keep track of funding conditions associated with contributors
- Your versioning system can be used as a basis for this record-keeping
- Establish what (if any) patents might be obtainable in relation to the work, and plan your code accordingly
- Assess your competition and your risk

Copyright in Software

An Introduction to Free and Open Source Licensing

Copyright in Software

Free and Open Source Software – What Is It?

- Software that the user has the right to adapt and distribute
- Adaptation is achieved by giving users access to the software's source code
- These rights are transmitted via copyright licensing
- It is often available at minimal or no cost
- It is often maintained and developed by a community of interested parties who may or may not be salaried for their work
- It has an increasingly high public profile and market share (linux, apache httpd, firefox, open office, xensource)

Copyright in Software

Some History

- Until the late 1970s most software thought to have little intrinsic value
- Exchange of software and its source code the norm (permissive BSD-style licences)
- Advent of personal computers in 1980s changed the perception of software's value
- Software became productized, source access closed off
- Many developers, particularly within academic communities, felt that this was detrimental to software quality and the teaching of software development best practice

Copyright in Software

Some More History

- As a result of the 'closure' of the source code to Emacs in 1985, software developer Richard Stallman rewrote it and made his version available under a new kind of licence that he composed himself
- His licence prevented re-licensing under variant terms and mandated that ‘derivative works’ must carry the same licence
- Stallman founds the Free Software Foundation at the same time, committed to maintaining software 'Freedom' as both a pragmatic and political aim
- Due to an unfortunate semantic collision in English, the use of 'Free' is widely and incorrectly thought to refer to price, not liberty (beer vs speech). Some prefer to adopt the French ‘Libre’.

Copyright in Software

The FSF's Four Freedoms

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

Copyright in Software

Yet More History

- In late 1997 Eric Raymond gave a paper at the O'Reilly Perl Conference called 'The Cathedral and The Bazaar'
- In early 1998, partly as a result of the success of Raymond's paper, Netscape decides to release the source code of its struggling web browser to the world
- Some within the Free Software community decide that Raymond's apolitical, business-oriented explanation of the virtues of the Free Software and permissive licences ought to have an advocacy group
- In February 1998 the Open Source Initiative is founded, with Raymond as its first president. The term 'Open Source' begins to be widely used.

Copyright in Software

Open Source Definition

- Freely Redistributable
- Source Code Included
- Derived Works Permitted
- Integrity of Author's Source Code
- No Discrimination Against Persons or Groups
- No Discrimination Against Fields of Endeavour
- Distribution of Licence (Rights)
- Licence Must Not Be Specific to a Product
- Licence Must Not Restrict Other Software
- Licence Must Be Technology-Neutral (no 'click wrap')

Copyright in Software

Open Source Initiative

- Fifty eight licences are accredited by the OSI as meeting these criteria
- The most commonly used are the BSD (permissive) and the GPL (copyleft)
- The sheer number of OSI-approved licences is officially considered a problem, and the OSI is working to reduce this number through retiring some licences which duplicate the functionality of others. Recently the OSI has recategorised their licences with a result that just nine achieve the description of 'Licenses that are popular and widely used or with strong communities"
- For practical purposes OSS Watch defines its remit with reference to the OSI approved licence list

Copyright in Software

Some Common Free and/or Open Source licences

Copyright in Software

Some Common FOSS Licences

- **GNU General Public License v2, v3 and AGPL**
- **GNU Lesser General Public License v2.1 &v3**
- **Modified BSD (Berkeley Software Distribution) License**
- **Apache License v2**
- **Mozilla Public License v1.1**

Copyright in Software

Some Common FOSS Licences

GNU General Public License v2

- **Significant Features**
- All modified versions of GPL-licensed software must also be distributed under the GPL (if they are distributed at all) (section 2)
- All modified versions must advertise prominently what has been modified, who modified it, and when it was modified.
- Source code must be provided with all GPL-licensed software, either directly or via a request to the licensor (section 3)

Copyright in Software

Some Common FOSS Licences

GNU General Public License v2

Significant Features

- All licensees of the software gain their licence directly from the original licensor (section 6). This preserves the licensors standing to take action against all licensees.
- No redistributing licensee may impose further restrictions on recipients (section 6)
- Additional restrictions placed on a licensee by a court mean that the licensee cannot distribute the software at all (section 7).

Copyright in Software

Some Common FOSS Licences

GNU General Public License v2

Notes

- Section 2 embodies the 'copyleft' or 'viral' aspect of the GPL. Where GPL'd code is used to produce a 'derivative work' (US term) the resulting work **must** also be licensed under the GPL if it is distributed
- The intention of this section is to prevent code that has been released to the community under an open source licence being 'closed' again by licensee who wishes to redistribute a work based on GPL'd code without also providing the source code to those who receive it. This usually happens when someone wants to make a closed-source commercial product using GPL'd code.

Copyright in Software

Some Common FOSS Licences

GNU General Public License v3

Significant Features

- Released June 2007
- Very similar in function to the GPLv2, although substantially rewritten
- Differences
  - 'Anti-Tivoisation' – all keys needed to run adaptations must be provided
  - Corrects unintentional incompatibility with some open source licences (most notably Apache 2)
  - Removes US-specific legal terminology and allows inclusion of regionalised exclusions of warranty and limitations of liability
  - Undermines customer-focused software patent non-enforcement covenants as a means of dividing the Free Software community

Copyright in Software

Some Common FOSS Licences

GNU Affero General Public License v3

Significant Features

- Identical to GPLv3 except:
- Adds a provision that obliges any user to preserve any 'source-spewing' functionality
- Designed to allow the community to benefit from Application Service Providers who use and improve AGPL'd software but do not distribute it and therefore would have no responsibility to make source code available under the standard GPL
- When requested over a network, any adaptation of the AGPL'd must send its current source code

Copyright in Software

Some Common FOSS Licences

GNU Lesser General Public License v2.1

Significant Features

Terms are substantially identical to the GPLv2 with the following exceptions:

- A work that is designed to be compiled or linked with the LGPL'd code is, in isolation, not a derivative work of the LGPL'd code and can thus be licensed in any way the author chooses (section 5).
- When distributing such code (perhaps in binary only form), the author can either not include the LGPL'd code at all, or include the LGPL'd code with its source and with copyright statements intact. The author must also make available tools and information that will allow the licensee to debug the interaction between the LGPL'd code and the author's code (section 6).

Copyright in Software

Some Common FOSS Licences

GNU Lesser General Public License v2.1

Significant Features

- Licensees may relicense LGPL'd code under the full GPL if they wish. They do this by changing the accompanying notices that refer to the LGPL so that they refer to the GPL, and including a copy of the GPL itself.

Copyright in Software

Some Common FOSS Licences

GNU Lesser General Public License v2.1

Notes

- (Slightly) less restrictive version of the GPL
- Originally intended to deal with the case of open source libraries.

"A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables."

- The LGPL can be used for any code, not just libraries, however:
- Derivative works of LGPL'd code must be libraries if they are themselves to be licensed under the LGPL. Other derivative works must be converted to the GPL, as detailed in section 3.

Copyright in Software

Some Common FOSS Licences

GNU Lesser General Public License v3

Significant Features

- Terms are substantially identical to the GPLv3, with the same modifications as between GPLv2 - LGPLv2.1

Copyright in Software

Some Common FOSS Licences

Modified BSD (Berkeley Software Distribution) License

Significant Features

- Short
- Unmodified versions of the software must retain the copyright statement, the licence conditions and the disclaimer of warranties.
- Prior permission must be obtained from the licensor before their name can be attached to any modified version.

Copyright in Software

Some Common FOSS Licences

Modified BSD (Berkeley Software Distribution) License

Notes

- The BSD licence does not prevent the code it licenses being absorbed into a closed source derivative.
- It is most appropriate for software which the author wishes to be as widely used as possible, regardless of whether it remains open source – for example code that implements a standard.
- The Modified BSD License is compatible with the GPL – code licensed under it can be combined with GPL'd code and the whole released under the GPL with no problems.

Copyright in Software

Some Common FOSS Licences

Apache License v2

Significant Features

- Unlike the GPL, linking your code to the interfaces of Apache v2 licensed software does not render the linked whole a derivative work (section 1).
- The licence grants patent rights as well as rights under copyright inasfar as those patent rights are necessary to operate the software (section 3).
- Anyone who starts patent litigation against a licensor automatically loses their licensee status (section 3).

Copyright in Software

Some Common FOSS Licences

Apache License v2

Significant Features

- Derivative works may be licensed in any way provided that the new licence's terms accord with those of the Apache v2 licence (section 4).

- Permission to use the licensor's trademarks, trade names, service marks or product names is not granted under the licence (section 6).

Copyright in Software

Some Common FOSS Licences

Apache License v2

Notes

- Section 3 (withdrawal of licence to individuals pursuing patent claims against the licensee) makes this licence incompatible with the GPLv2 (it is an additional restriction).

- GPLv3 *is* compatible with this licence

- This licence does not prevent the 'closed-sourcing' of code licensed under it.

Copyright in Software

Some Common FOSS Licences

Mozilla Public License v1.1

Significant Features

- Modification may be distributed as differential comparisons against the licensed code (patches) (section 1.11)
- Source code must be made available with any distribution of the software or modified versions of the software (section 3.2).
- Executable versions of the code may be distributed under a separate licence provided that the distributor himself is in compliance with the MPL and the source to the executable continues to be available under the terms of the MPL (section 3.6).

Copyright in Software

Some Common FOSS Licences

Mozilla Public License v1.1

Significant Features

- Licensees may create a 'Larger Work' – that is a combination of the MPL-licensed code and other code – and distribute the whole. In these circumstances the MPL-licensed code must continue to be distributed under the terms of the MPL, but the other code may be licensed as the author wishes (section 3.7).

Copyright in Software

Some Common FOSS Licences

Mozilla Public License v1.1

Notes

- Resembles a commercial licence more closely than other open source licences, due to its origin in a large corporation.
- The MPL represents a half-way house between the permissiveness of the BSD licence and the strictness of the GPL. Section 3.7 opens the possibility of a licensee taking the code, including the contributions of many other programmers, and adding functionality to this codebase in a fashion that qualifies as 'other code' (meaning code under another licence). In this case, the licensee can sell licences to their own 'other code' without providing the source, and distribute it as a bundle with the MPL-licensed material.

Copyright in Software

Exploitation Strategies for FOSS-licensed Software

Copyright in Software

- First I should say...
- Many commonalities with software-related business in general
- Not an exhaustive list
- Still an evolving area of business practice
- Your aim may just be sustainability or internal cost reduction
- Dissatisfaction with the short-comings of current proprietary models is a strong factor in the current success of some open-source businesses
- Smorgasbord

Copyright in Software

- FOSS Exploitation Models
- *Trademarking*
- Just because your code is open, it does not mean that your brand has to be
- Competitors may fork your code but they cannot use your trademark
- Effective both for direct software provision and service provision
- Possible downside: Trademark law not as internationally consistent as copyright law

Copyright in Software

- FOSS Exploitation Models
- *Service Provision*
- Training and accreditation
- Customisation
- Consultancy
- Support
- Possible beneficial interactions with 'rivals' (Zea consortium model)

Copyright in Software

- FOSS Exploitation Models
- *Red Hat monolithic subscription model*
- Customers pay a subscription to gain access to live support, a knowledge base, intellectual property indemnification, a tested suite of binaries including upgrade facilitation technologies and free distribution upgrades as long as the subscription lasts
- Red Hat runs the Fedora project in parallel with Red Hat Enterprise Linux and migrates tested developments in Fedora to the commercially-supported RHEL
- The source to RHEL's components remains available to anyone

Copyright in Software

- FOSS Exploitation Models
- *Xensource B2B model*
- Specialist consultancy to resellers
- Provide bundles of second-tier support to resellers
- Interact with development community while leaving customer support to downstream resellers

Copyright in Software

- FOSS Exploitation Models
- *Squiz proprietary bolt-on model*
- Core software is available under an open source licence
- Certain plug-in modules for commonly requested functions are available under a non-open licence with additional support
- Possible downside: may have to compete with community-developed open plug-ins

Copyright in Software

- FOSS Exploitation Models
- *MySQL / LAMS / Sleepycat Dual Licensing model*
- Applicable if you own all the IP in the software in question
  - IPR in community-submitted patches must be assigned to the company or licensed very permissively if they are to be included in the 'core' release
  - Software is available to all under a 'copyleft' open source licence (GPL)
  - Software is also available under a non-open licence to those who wish to adapt it and distribute it under something other than a 'copyleft' open source licence

Copyright in Software

- FOSS Exploitation Models
- *Community Creation*
- A body of useful software can become the focus of academic, social or commercial work for people from many different backgrounds
- Work on publicly-tracked and -available software can be a convenient route to recognition and progress within a given problem domain or community, both for individuals and institutions
- Increasingly funding bodies stress Business and Community Engagement as a desired outcome for the projects that they fund. FOSS communities can serve as examples of both.

Copyright in Software

Questions?

Copyright in Software

- Links
- **OSS Watch –** http://www.oss-watch.ac.uk/
- **The Free Software Foundation** - http://www.fsf.org/
- **The FSF Europe** - http://www.fsfeurope.org/
- **Open Source Initiative –** http://www.opensource.org/
- **European Patent Convention 1973** - http://www.epo.org/patents/law/legal-texts/html/epc/1973/e/ma1.html
- **Apache Contributor Licence Agreements** - http://www.apache.org/licenses/#clas
- **UK Government Open Source Policy v2** (PDF) - http://www.govtalk.gov.uk/documents/oss_policy_version2.pdf