

Project Harmony:

Inbound transfer of rights in FOSS Projects

Amanda Brock^a

(a) General Counsel, Canonical Group

DOI: [10.5033/ifosslr.v2i2.49](https://doi.org/10.5033/ifosslr.v2i2.49)

Abstract

Software engineers participating in open source software projects are frequently asked to sign Copyright Contribution Agreements ("CCAs") with respect to the contributions they make to those projects.

As lawyers we are happy to read and understand copyright agreements and licences. As lawyers, few of us can code. Software engineers (at least in Canonical) seem to understand more law than might be expected, but their expertise is software, not law. The proliferation of forms and wording in the CCAs mean that they are routinely asked to sign up to a diverse selection of CCAs differing in meanings and intentions. Valuable coding time is wasted as they wade through legal wording.

This is not only an issue both from a productivity and efficiency perspective, but is also problematic in ensuring that developers understand what rights they grant in their work..

Project Harmony came into being in the Summer of 2010. Its birth followed a long incubation and much discussion within Canonical and with parties such as SFLC and lawyers specialising in FOSS. It is intended in the first instance to tackle CCA Proliferation.

Keywords

Law; information technology; Free and Open Source Software; Copyright; CCAs; Copyright CCAs; Copyright Licence Agreement; Project Harmony

Info

This item is part of the [Articles](#) section of IFOSS L. Rev. For more information, please consult the relevant section policies statement.

This article has been independently peer-reviewed.

Project Harmony – a response to the proliferation of contributions

I joined Canonical in February 2008. Sitting between the developer community and commercialisation of FOSS (Free and Open Source Software), Canonical (through the Linux distribution it represents, Ubuntu) is at the forefront of change and progress in Open Source.

It was a culture that shocked the life back into me after time away from the IT sector. How anyone can fail to love hurtling forward on adrenalin at 100 miles an hour is beyond me. The collaborative nature of the developers has rubbed off on the lawyers involved in FOSS. It is a wonderful experience for a lawyer to find free and open discussion and a collegiate approach between colleagues across companies, countries and firms.

Within a very short time of joining Canonical, with its large developer staff, the piles of paper landing on my desk included CCAs –a consequence of the developers saying: “we have a lawyer now!”

I am not a developer. I had not worked on any other open source project before joining Canonical. I was slightly confused...

An open source software project, whether run by a commercial entity, a not-for-profit or otherwise, frequently establishes a "custodian" of the software. I use the term “Project” to cover both the Custodian and the software.

The custodian is responsible for the wellbeing of the code in the Project, including co-ordinating [release management, feature adoption] maintenance and bug fixes. However, what legal rights that custodian holds in the code and how the custodian manages and administers the legal rights and IP varies between Projects. In most cases, the Project receives contributions from individuals and those individuals form part of a community. The formation of a community is not obligatory and is by no means universal: there are projects which involve the commercial development of FOSS code (which, in development terms, are frequently indistinguishable from proprietary developments), which are internally sponsored by a commercial entity and which do not develop an external community. However, the establishment of an enthusiastic community is a defining characteristic of successful FOSS Projects.

The CCAs I was seeing as Canonical's first lawyer were from many different Projects, some commercial, some not. There were all sorts of CCA documents - copyright assignment agreements (including some misleadingly entitled copyright licence agreements); agreements containing tough obligations on contributors providing free code without payment for their work; agreements lacking clarity as to who contributed as an individual and where a corporate entity should sign for the contribution, and all sorts of wording which had clearly been edited by well-intentioned non-lawyers (NALs). The lack of clarity can even cover issues as fundamental as what work is being contributed, what the CCAs are meant to cover and who the licence or assignment is being given to (for example, where no formal legal entity has been created for a Project).

The outbound licences on which FOSS Projects distribute their software to Projects downstream and other users, are, by comparison, very neat and structured. A couple of decades of active licence creation, followed by consolidation in many projects as a result of the efforts of OSI and others means that, in effect, the licences have become regulated. Of course, this regulation is not by any appointed authority, but is an emergent characteristic of industry and community engagement. The organisations that have emerged out of the community provide boundaries and guidelines as well as, in some cases, approvals for these licences. As a new FOSS lawyer, I was learning fast about the work done by OSI and others in reducing licence proliferation in outbound FOSS distribution.

Many involved with this Journal are much more qualified to write about these licences than I and I won't go into more detail here. But, it struck me as odd that where industry activity had led to FOSS licences becoming clearer and more consistent, in contrast, there was no similar activity in relation to this mass of confusing, potentially onerous and sometimes incorrect CCA documentation that I was reading.

As lawyers we know that there are very few judicial decisions on software licensing in general and, more specifically, free and open source software. We realise that we are operating internationally, without precedent and in many jurisdictions. The paths we tread with respect to these licences are carefully planned and whilst we may not have legal authority to rely on and use as a basis in making our decisions, one thing our industry is not short of is opinion.

Whether made by individual lawyers or by developers (in which case it is likely to be prefaced by "IANAL" - I am not a lawyer - and which is inevitably followed with "but...."), we have the thoughts of the great and the good of the FOSS development community and its lawyers on our outbound licences available to us, through articles, books and blogs. We may not be able to guarantee how a court will interpret a licence provision or its enforceability from one jurisdiction to the next, but we have an idea of why that licence came into being what, the intent of its creator in drafting its provisions are and why someone might choose to use it as opposed to another licence. The same is not true of CCAs.

For some time we discussed this internally in Canonical with both management and developers (and in many cases the two are the same). With the support of Canonical's senior management, I started to speak to my contemporaries and the bodies representing the FOSS Communities, such as SFLC. From these discussions and our recognition of the need to improve CCAs and to remove proliferation of CCAs, Project Harmony was set up.

Project Harmony Terms of Reference

Terms of reference for Project Harmony were circulated after our US kick-off meeting on 16 June, 2010 and agreed by our follow up London meeting in July of that year. They are:

Project Harmony is made up of a group of industry interested parties, from companies, projects and those with personal interests in FOSS. It is open to all interested parties to join. Its initial goals are to avoid proliferation in CCAs across

FOSS software projects where those organisations chose to work with CCAs. In doing this we hope to assist organisations which use CCAs by providing standardised variable templates with clear and concise explanations; to come to a common understanding on these; and to recognise the relative maturity of FOSS by dealing with its internationalisation. Our goal is to make the process of contributing to FOSS projects easier for developers regardless of who their employers are. We believe that standardised CCAs should make it easier for developers to contribute regardless of who their employers are

We may look to establish this forum for these CCAs into an organisation which will have a long term role in administering this area.

To achieve our initial goals we will be working together to plan and to create:

**ContributionAgreements*:*

*A suite of standard language which works across international borders and which provides standardised wording for FOSS contributor agreements, minimising the need for legal review. We understand and accept that there are varying positions and decisions in different projects based on commercial positions, community preferences and the wording needed to meet this. **Inter-CompanyAgreements*:*

In addition it is recognised that there would be an advantage to put in place "inter-company" agreements for employee contributions to other company or projects' software. The intention of such documents is to remove the need for employee signature of project contributor agreements and to equalise the playing field between organisations.

It has been repeatedly emphasized in our meetings that clear and useful FAQs are an essential output of Project Harmony and a goal of the Project.

We are currently working out whether we will deliver a single modular document encompassing options or if we will follow a model more like Creative Commons, multiple documents. We will certainly learn from the past, and whilst I may be relatively new to this type of activity, other participants were involved with Creative Commons and GPLv3's creation.

Some Thoughts on Copyright

I have set out in the next few pages some thoughts around Copyright and CCAs. These are some general thoughts, trying to look factually at various options and copyright issues and are certainly not intended to recommend any one solution over another. They are also not intended to represent the output that we will provide from Project Harmony, but they do encompass some of the considerations we have in the Project.

To the extent that I have failed in my impartiality, quality of explanation or accuracy, the thoughts

that follow are my own and not those of Harmony. One of the best things about the Project Harmony is that we have managed to pull together an expert and considered group of people, who are not only very learned in their fields but who are, some might say, very unharmonious in their beliefs! Some do not see the need for CCAs. Others have been very vocal in their opposition to assignment or to developer warranties. Some think assignment is the only answer and others prefer licences.

There is room in our Project for all views and as can be seen in our terms of reference, to meet those, we need to create (a) document(s) and supporting FAQs which will achieve industry acceptance, to include all views.

Why have a CCA?

In climbing my personal CCA mountain during my first few months at Canonical, it also took me a while to understand why some Projects felt they did not need a signed CCA at all, whilst other Projects would not accept any contribution without signature of their CCA. Surely it was not just differences in their legal advice that made such vast differences between Projects?

Clearly any Project requiring an assignment must have a document. In many jurisdictions an assignment of copyright or other IP rights is not valid if it is not in writing and signed.

I discovered that Projects often have governance provisions in place to regulate participant's conduct and the contribution of code, such as Canonical's Ubuntu Code of Conduct. Participation is subject to adherence to these rules and some projects have historically considered and may well continue to consider that these are enough, through the development of community norms and extra-legal self-enforcement. The Linux kernel is a good example of this and relies on exceptionally strong Project governance.

CCAs may offer some benefits. Initially people may have contributed or contribute under their nickname or without their full identity being known. Their identity may have been known, but if they later became uninterested and moved on, they might also become difficult to contact. CCAs provide a means for a FOSS project to track who is providing code or making contributions into the project and form part of many projects' governance.

They may avoid problems by showing:

- who contributed what
- documenting their agreement as to how their contribution is used
- confirming that the contributor had the right to make the contribution

In short, CCAs can be used to confirm the pedigree of the contribution and in turn, the pedigree of the Project. To grant its outbound licence in the IP in its software, the Project holds itself out as having the right, to grant that licence. So it must either be the copyright owner or have the right to do this under (an) appropriate licence(s) to it. If it does not have that right then it cannot give a valid licence. Each project is responsible to those it distributes to. Such a licence could be unwritten, but that could not in the case of an assignment and even in a licence this pedigree

requirement may not be met.

A CCA will deal with licensing and assignment of rights from a legal perspective, but its also a political document. The CCA reflects the organisation or project's positioning in the market place and the Project's beliefs. It can be very different to look at a CCA to be used by software in a commercial context such as Canonical and in a not for profit such as Apache. These Projects each have differing needs which a CCA has to meet. So, we have to accept right at the start that CCAs need to meet not only legal requirements but also these political and social needs.

A brief discussion of Copyright

The main IP (Intellectual Property) right attaching to code is copyright. In some jurisdictions code can also be protected by patents. I do not personally believe that patents offer a good or helpful protection for code. However, this is not the place to argue those views. Patents exist, and therefore need to be considered by Harmony as part of its process around CCAs (whether we like them or not) and that consideration is referenced later. I am focusing on copyright, therefore, in this piece.

The New York law school article in this issue of IFOSSLR clarifies well the overall issues (at least under US law) faced by Projects and businesses in the FOSS world and highlights how these cascade to the individual engineers. I also set out a few pointers below.

Copyright ownership and Licensing

Where someone creates a copyright work, other than in the course of their employment, that person is the holder of the copyright. If a work is created in the course of employment, then the work and the copyright in it belong to the employer in the UK, in most of the US and many other jurisdictions. CCAs need to be cognizant of this, so that the correct rights are transferred from the correct donor to the correct recipient. Copyright can only be held by a legal entity, that is an individual or a company or other organisation having legal standing. Under English law at least, an assignment to an unincorporated entity is likely to result in some form of trust arrangement, whose structure can only be revealed with certainty by the courts.

For some FOSS projects which have not established a legal entity, it would not be possible for the "Project" to own the copyright except in this uncertain and unsatisfactory way. The original creator of the software may hold the software as an individual. One option is to allow a trusted third party to hold the copyright in way which gives the original creator that the third party will deal with it in an appropriate way. The FSF is an example of an organisation that acts as a trusted third party and has an assignment agreement which allows Projects and their contributors to assign copyright to it.

Where multiple people contribute and create the copyright in the code, then without a copyright assignment (or a common employer) there will be multiple owners of the code. In the UK if each person's contribution to the work is not something that can be separated from the others', then , without any transfer of rights, that results in a work of joint copyright. Where a joint copyright

work exists, all of the owners jointly own the whole and all of the owners would have to consent to any use of their rights in the work. So, if they disagree there will be problems.

On the other hand if each contributor's part can be distinguished, then each person owns their own piece of code and the copyright in it under English law. Whilst each owner can make their own decision about this, co-ordinating the multiplicity of interest in relation to issues like Project licensing can be difficult.

In many ways, shared or joint ownership sounds like the ideal, and in line with the collaborative approach taken by FOSS projects. Shared and joint copyright ownership can be problematic not just in terms of enforcement but practically, in terms of tracking owners of the shared copyright. First, where there is not a single owner of copyright in the whole, there is no single party who is able to grant rights to use the code, in the UK at least except subject to the terms a licence all owners may have entered into.

Secondly in some jurisdictions, to enforce the copyright and bring a claim against someone who infringes that copyright: under US law all the joint owners have to consent. An assignment allows the copyright holders an instrument to collectively defend and enforce their copyright without giving away "the equity ownership", as in many jurisdictions you must actually be the copyright holder and not a licensee to bring an action.

This means that to take these actions the project would have to find all of the contributors and get their agreement. In older FOSS projects where there is no "pedigree" *i.e.* we don't even know who all the contributors are this could be a nightmare. In projects where there is a pedigree, the issue would of course be having all parties come to an agreement.

Where multiple people create or contribute to a copyright work, or code with copyright in, then a single entity or legal person can own that code. For this to happen there has to be a legal assignment of the code from the creator of the code (unless they are an employee when this will be automatic) to the holding entity. The holding entity could be a commercial legal entity, or a trusted third party such as the FSF.

In a CCA with an assignment, the contributor assigns copyright and is simultaneously granted a licence back to allow him or her to use the software, from the Project. If this is very broad then the licence back allows the contributor to retain full rights to re-use, distribute, and continue modifying the contributed code. Apart from allowing a single entity to manage the IP of the whole Project as discussed above, this is almost the same as an exclusive licence.

Where an assignment is used there is a legal requirement that this is in writing and signed.

Contributions under an exclusive licence

Projects may not require assignment, but instead have a CCA which includes a licence. If this licence is an exclusive licence then commercially and practically this is to all intents and purposes the same as an assignment.

A Project which is the recipient of code under a CCA granting an exclusive licence, of course has

to consider possible re-licensing to downstream recipients (or changes in the downstream licence) and enforcement. The re-licensing can be covered by a mechanism contained in the CCA.

However the contributors' rights will to a large extent depend on the licence back given by the Project. Unlike a sole licence, an exclusive licence to a Project will mean that the contributor is not free to use his contribution. The effect of an exclusive licence on the contributor can be as restrictive as an assignment.

CCAs containing a Non-Exclusive Licence

If the licence is non-exclusive, then the usefulness of such a licence to the Project will very much depend on its wording. The rights must be wide enough to allow the Project to use the contribution itself as appropriate, to distribute it under an outbound licence. The CCA should, of course, deal with any outbound commitment in terms of the types of outbound licence under which the Project may employ.

How well this will work practically, commercially and legally will depend entirely on the drafting.

The Project may have difficulty in enforcing the copyright in its work against infringers where it has non-exclusive licences in its CCAs (in the UK, the Copyright Designs and Patents Act 1988 does anticipate that a copyright owner can authorise a non-exclusive licensee to enforce copyright in certain circumstances, but the scope of the provision is unclear and will provide the alleged infringer with scope for argument). If the CCA does not deal adequately with outbound licensing the Project may also find itself unable to release under different or additional licences both now and in the future.

Outbound Licences

One further challenge faced by FOSS projects specifically and which was encountered by Mozilla, when they wished to increase flexibility by enabling their codebase to be licensed under more, and more liberal licence: where a CCA provides that the Project use a specified outbound licence. it may not be possible for the Project to change the licence for a later version or to release under other licences in parallel

It may be that over time or for a particular instance, the Project wants to move to a new or additional licence. If the Project does not own the copyright or have a right in the CCA to do this then unless all of the owners can be found and persuaded to agree, this cannot happen (in the absence of a licence granting that right, or without extracting from the code-base the contributions of errant contributors). Ownership of copyright means that the copyright owner can more easily change the licence under which the software is released. Through a contractual term in the CCA that right may be conditional – *e.g.* it must include an OSI approved licence – or unconditional, depending on the nature of the Project.

CCAs may also extend this right to a Project without assignment and these agreements often include the right to change the licence. Whether assignment or licence, the ability to license under an outbound licence of many different flavours can lead to concerns amongst contributors. When

contributors make a contribution, they tend to know what the licence the software Project they are committing to is on and may be relying that their contribution will also be released under that licence.

Some contributions deal with this by making a commitment to contributors to release the contribution initially under the Project licence current when the contribution is made. Once software is released under a FOSS licence that licence cannot be reneged upon and the software will continue to be released under that licence. This licence commitment does not, however, remove the right for a Project to release that or later versions of the software under different licences, often in parallel with the original licence. This is often referred to as “dual licensing”. It may involve the same software being distributed under more than one FOSS licence or a FOSS and any other including a closed source licence. Many CCAs include statements around this, known as the "outbound commitment".

If a project is started with joint copyright, in England at least, that is very hard to unravel ownership issues in the future.

Examples of Issues Which Harmony Intends to Address

In my experience, discussions starting “IANAL, but...” frequently involve the following issues. We intend to address these through Project Harmony both by producing appropriate wording in the CCA, where appropriate, and addressing the issue in the FAQs.

"There is a minimum amount of code required to attract copyright" This is untrue. The question we need to consider in establishing if copyright applies is whether there is an original work in the UK and in the US, in most cases is a three point test, discussed in depth in the New York law school article.

This is not purely a question of size, but concerns content and originality of the code. Copyright does not consider if the code is inventive and does not protect facts, ideas or systems and methods of operation. In the UK copyright arises once that creative work is fixed in some medium, *i.e.* is somehow written down. So the code in your head is not protected, but once it is typed up copyright automatically arises.

Work for Hire

This is a US concept. In the US, the commissioner of a work owns the work. In the UK, this is not the case (at least not for copyright): the creator owns it unless there is an assignment or the work is created in the course of employment.

Moral Rights

The right to be identified as an author. The UK excludes these by statute with respect to software.

Public Domain

The ability to dedicate works to the public domain exists in the US but this ability does not extend to all places outside the US. For example, in UK, it is generally accepted that copyright works can only enter the public domain once their relevant term has expired. Pending such expiry, copyright cannot disappear and there is always a copyright owner (even if it is impossible to determine who it is).

Need to register Copyright

In the UK at least. Even US lawyers I have discussed this with have found it confusing. In some countries, such as the US there are registration requirements but these don't exist in the UK.

The laws of copyright are not be the same in every country in the world, despite the prevalence of the Berne Convention, and other international treaties. Even though the law of copyright as it applies to computer programs is subject to the Computer Programs Directive 2009/24/EC in Europe, there are still significant differences in copyright law and practice from one European country to the next.

Key Issues in CCAs which are being considered by Project Harmony

The intention here is not to provide a detailed report on our discussions or findings so far, but instead, to set out a brief overview of some of the matters being considered.

- What is contributed, by whom and to whom
individual or company making the contribution?
if an individual, (1) is the contribution as an employee or not and (2) confirmation that the signatory has the right to make the contribution
the rights of minors to make contributions
- IP - Intellectual Property rights in code
- What is being contributed - code or more?
- How is it contributed - only stated contributions or flag if not a contribution
- How is it legally contributed - assignment, exclusive licence, non-exclusive licence
- What if any, commitment is made to the contributor with respect to the distribution of their contribution, (i.e any outbound commitment)
- What if any commitment should be made by contributor re IP eg warranties and indemnities
- Trade Mark – should any trade mark rights be granted or accepted? (We are reaching the conclusion that trade marks are irrelevant in the context of a CCA).
- Copyright law - see above

- Patents - CCAs may rely on an implied patent licence that runs with the code and a personal obligation not to sue us for breach of their patents existing at the time of the contribution or in the future (called a hold harmless or non-assert) from the contributor. At present any need for a patent licence, a hold harmless or non assert or duty to notify of an infringement are all under discussion. For example, should the scope of the non-assert or hold-harmless be limited to patents owned or controlled by the contributor personally, or extend to the entity he or she is employed by?)
- Boiler Plate
- governing law
- excluding the US and UK: implied warranties of merchantability/ satisfactoriness and fitness for purpose
- any need for termination

Project Harmony Process and Participation

The documentation goal is to launch our output to delegates at the European Legal Network's conference on 8 April 2011, and simultaneously to the public through the Project Harmony website. After this time it will be more appropriate to discuss in detail the discussions and reasoning of the Harmony Project in some depth.

Participating in Project Harmony is open to all. We have a network of around 80 participants, being a mix of lawyers and developers. Mark Radcliffe of DLA Piper has taken over the mantle of drafter from SFLC, who led drafting through the first half of this process. Both SFLC and Mark Radcliffe have been in the role of third party editor and cannot dictate or choose the terms. The content of the document is something that the Harmony Project is collectively responsible for..

We hold weekly legal drafting meetings on Wednesdays at noon Eastern Time, which all can participate in, and the draft is available for comment to all participants through coment. Meetings are held by conference call and IRC.¹

Face to face meetings have been helpful to advance drafting and to ensure everyone is aware of the latest developments. Meetings have already been held in Boston, London and Palo Alto, and two more are scheduled for 2nd February and 2nd March in the lead up towards a presentation of outcomes at the European Legal Network Conference in early April.

Project Harmony does not have a CCA.

Our only rule is the Chatham House Rule, means that although Harmony's activities can be freely discussed, no comment may to any individual or organisation. We have a Wiki which stores our meeting minutes, related articles and our thoughts. The Wiki is open to all participants. We have a web site at <http://harmonyagreements.org>.

¹ Internet Relay Chat

Anyone who wants to participate should send me an email, amanda.brock@canonical.com with their request and a short Launchpad name which they can obtain by registering on Launchpad.net. This will give you access to the mailing list, the WIKI and the ability to participate in meetings.

We will include an update on our output in the next edition of IFOSSLR.

About the Author

Amanda Brock studied law at the University of Glasgow, New York University and Queen Mary and Westfield College, University of London and is admitted as a Solicitor in England and Scotland. She is General Counsel of Canonical Group and has worked in-house since 1999.

Amanda has lectured and written widely on commercial, IP and IT law in both business and academic contexts. She was on the Editorial Board of Butterworth's Electronic Business Law, is a founding Editor of IFOSSLR and is on the Advisory Board of the Open Source Centre of Excellence at the Centre of Commercial Law Studies, Queen Mary and Westfield, London.

Licence and Attribution

This paper was published in the International Free and Open Source Software Law Review, Volume 2, Issue 2 (December 2010). It originally appeared online at <http://www.ifosslr.org>.

This article should be cited as follows:

Brock, Amanda (2010) 'Project Harmony: Inbound transfer of rights in FOSS projects', *IFOSS l. Rev.*, 2(2), pp 139 – 150
DOI: [10.5033/ifosslr.v2i2.49](https://doi.org/10.5033/ifosslr.v2i2.49)

Copyright © 2010 Amanda Brock.

This article is licensed under a Creative Commons UK (England and Wales) 2.0 licence, no derivative works, attribution, CC-BY-ND.

As a special exception, the author expressly permits faithful translations of the entire document into any language, provided that the resulting translation (which may include an attribution to the translator) is shared alike. This paragraph is part of the paper, and must be included when copying or translating the paper.

