

# Open-Source Software: What is it, and how does it work?

*Ben Kremer, Freehills*

---

BSc (Hons), LLB (Hons) (Syd), DPhil (Oxon). Dr Kremer is a solicitor in the intellectual property group in the Sydney office of Freehills. He has programmed computers for over twenty years.

---

Open-source software is playing an increasingly important role in the marketplace. Its legal nature is, however, still subject to misunderstanding at both the lay and legal level. This note will attempt to clarify some of the uncertainty by addressing the two most fundamental issues: what open-source software is, and how it works.

## 1 What is open-source software?

In order to understand the open-source software model, it is necessary to understand what software is. The term “software” refers to two different, but related, things. One is “source code”, which is a set of human-readable and understandable instructions that comprise the “recipe” from which an executable program can be made. The other is “object code”, which is the actual executable program – the bundle of 1s and 0s which are fed to a computer’s microprocessor to make that computer perform various operations.<sup>1</sup>

For various reasons, software is invariably written as source code.<sup>2</sup> That code cannot itself be executed by a computer, but is “compiled” to form object code by a specialised program, called a compiler. The process is hard to reverse (a technique called “decompilation”), as the process of compilation generally strips out things such as comments inserted by the programmers to describe how the program works, and things like the names of variables or subroutines. (This information generally cannot be restored during decompilation, so decompiled source code is often difficult to understand and modify.)

Commonly, various functions can also be incorporated in source code to allow the production of multiple, different versions of object code – such as “demo” programs; program versions with greater or restricted

functionality; or programs optimised for different computers or operating systems – from the same source code with minimal effort.

### 1.1 The “closed-source” model

Historically, software vendors kept source code to themselves (indeed, invariably protecting it as valuable trade secrets) and sold, or licensed, only the object code versions of programs to their customers.<sup>3</sup> Customers can run those pieces of object code, but cannot view or modify the source code (and hence modify the behaviour of the program) without dealing with the software vendor. Licence terms commonly prohibit customers from decompiling the object code. As a result, they can only use whatever is provided to them, along with any modifications they could prevail upon the software supplier to incorporate. This model is called “closed-source” software, and continues to be used by the majority of commercial software companies.

### 1.2 The “open-source” model

“Open-source” software, simply put, is software whose source code is released openly, alongside the object code. Each user thus has access to the original source code (with no need to produce an inferior version by decompilation). The intent is that any user of the software is “free” to modify, enhance or customise the program’s source code, and use it to produce modified object code, ie, to produce a customised, updated or improved version of the program. Hence the common description of open-source software as “free software”: “free” is used in the sense of “free to modify” rather than “free of cost”, although much open-source software is both.

The hardest conceptual problem about open-source software is how to ensure

people play by the rules. There are many models, but the most common is to require any person who redistributes an open-source program (whether in its original form, or with any changes they have made) to also redistribute the accompanying source code. The requirement is necessary to overcome human nature: after all, taking a program, making some changes, and redistributing the modified version while keeping the modified source code to yourself is an intuitively valuable way of creating and preserving a competitive edge over others. Anyone else who wants to replicate your modifications to the code must reduplicate your work themselves. Such a user could thus springboard off the earlier open-source program to produce a closed-source model for themselves.

The answer is easy, but apparently counterintuitive to many: it’s done by *copyright law*.

## 2 Copyright and the GPL

### 2.1 The myth of contract

Many people seem to assume that, if one person (the original producer of some open-source software) wants to control or limit another’s actions (the subsequent modifier of that software), then there must be some kind of *contract* between the two. While it might be possible to achieve such protection by contract, it would not work in practice. As a general matter, to be enforceable, a contract must be entered into voluntarily by each party. There is usually no way to *force* another person into a contract. Thus, if the original producer of open-source code made it available to others to access without restriction, there would usually not be, as a practical matter, any way to force that other person into the relevant contract. Even if there were, the transaction costs involved would be significant.

## 2.2 Enter copyright

However, software, in both its source and object forms, is a copyright work if certain conditions (such as originality and substantiality) are met. The original producer of open-source software will thus own copyright in it.<sup>4</sup> Copyright is a bundle of exclusive rights, including the rights to control the copying and adaptation of the copyright work. The default position is that only the copyright owner may reproduce the work (ie, make a more or less exact copy of it, or a substantial part of it), or make an adaptation (ie, change it into another equivalent but different form of expression, such as translating it<sup>5</sup>). In the context of computer programs, this includes the sole right to copy the source code, make modifications to it, or compile it to form object code, and the sole right to copy the object code thereby produced.

## 2.3 Licences

There is nothing, however, to stop the copyright owner from permitting others to do one or more of those acts. And there is nothing to stop that permission being granted on terms. A grant of permission to a person to do something that that person is otherwise forbidden to do is quite common. In legal terms, that is the precise definition of a licence.<sup>6</sup> Licences can come about by contract, or by oral permission (subject to any applicable statute), and can be simple or convoluted, and narrow or broad, but all share the same basic form: a person who otherwise has the power to exclude another from an act, and who has the power to authorise that other person to do the act, does so.

The power to exclude can come from a number of sources, but the most common is property or statute: the occupier of land can allow another person to come onto that land, but in the absence of such permission could generally sue them for trespass. The owner of a car can authorise another person to use it, when without that permission such use would be a trespass and a conversion of the fuel.<sup>7</sup> And a person who holds copyright – which is itself a form of statutory property – can authorise another person to do some, any or all of the

various rights which that copyright initially vests exclusively in the owner.<sup>8</sup>

The way the open-source model is usually made legally effective is to make the requirement to disclose the source code of any modification of the original source code a term of the permission to deal with the original copyrighted source code. Provided subsequent users adhere to the conditions upon which permission is granted, they are within the scope of the permission, and they cannot be sued for infringing the original user's copyright when they make any copies, or adaptations, of the original source code. Such qualified consent is common: the permission given to enter a person's house to attend a party does not give you the right to live there indefinitely; the permission given to enter a shop does not allow you to remain and start busking; and the permission given to affix a person's trade mark to specified goods does not allow you to affix that mark to any other goods. Stray outside the scope of consent, and you are again subject to the legal regime which gives the owner the right to exclude you.

## 2.4 The GNU Public Licence

The way this operates for open-source software can be clearly seen in the "GPL" (the "GNU Public Licence") one of the most common licences under which open-source software is distributed.<sup>9</sup> In addition to various other requirements, section 3 of the GPL provides:

*"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". (Italics added)*

Although Section 3 allows two other means of compliance, the method in

paragraph (a) is probably the most common. Its effect is simple: if the original program is released under the GPL, then you may copy and distribute it, or a modified version, without infringing the copyright in that program *provided you also distribute the relevant source code*. The provision of your source code is integral to your ability to deal with the original source code. If you fail to distribute your source code, you have gone outside the terms of your original permission to deal with the copyright work, and (in the absence of any applicable defence, or other permission to deal with the work) you become liable for infringing the copyright in it.

To some people, this seems to be a novel use of copyright law. It isn't really: replace "accompany the work with source code" with "pay the copyright owner \$x per copy", and you have something approaching a standard royalty contract. However, the elegant simplicity hides much power: remedies for breach of copyright are often more powerful than remedies for simple breach of contract. There are many subtle areas involved in open-source law and the GPL, but the basic point is quite simple: it's all about copyright.

<sup>1</sup> See Gibbs CJ's definition (now somewhat dated) in *Computer Edge Pty Ltd v Apple Computer Inc* (1986) 161 CLR 171 at 178-9.

<sup>2</sup> Source code may be written in a number of languages, such as BASIC, FORTRAN, Pascal, C or C++. Most languages attempt to mimic English-like structures, for example by using terms such as "if", "else" or "for". Object code is phrased in terms of microprocessor instructions, customised for a relevant microprocessor. Generally, multiple instructions are needed in object code to implement a single instruction in source code. Pieces of source code can often be moved around without needing to change the rest of the code much or at all, while it is often a complex exercise to modify object code even slightly.

<sup>3</sup> See for example the evidence given in *Ni-Tech Pty Ltd v Parker* (1998) AIPC 91-411 relating to protection of confidentiality in the source code, and see also *Quanta Software International Pty Ltd v Computer Management Services Pty Ltd* (2001) AIPC 91-757.

<sup>4</sup> This note does not discuss the various legal provisions affecting ownership of copyright, such as where an employee creates a copyright work in the course of

his employment, or where a work is made "for hire", or created jointly by several authors. See eg *Redrock Holdings Pty Ltd v Hinkley* (2001) 50 IPR 565.

<sup>5</sup> In the context of computer programs, adaptation will likely include re-expressing the work in any other type of code: *Data Access Corporation v Powerflex Services Pty Ltd* (1999) 202 CLR 1 at 39 [108]. Compiling source code probably produces an adaptation of that code, rather than reproducing it: see eg "Copyright and Computer Programs: *Data Access v Powerflex* Before the High Court" (1997)

20 *Sydney Law Review* 296, available online at <[http://www.law.usyd.edu.au/~slr/v20/n2/bhc\\_kremer.html](http://www.law.usyd.edu.au/~slr/v20/n2/bhc_kremer.html)>.

<sup>6</sup> See eg *Federal Commissioner of Taxation v United Aircraft Corporation* (1943) 68 CLR 525 at 533: "A licence provides an excuse for an act which would otherwise be unlawful as, for example, an entry upon a person's land, or the infringement of a patent or copyright. It is an authority to do something which would otherwise be wrongful or illegal or inoperative".

<sup>7</sup> *Scott v Davis* (2000) 204 CLR 333 at 384 [154].

<sup>8</sup> See section 196 of the *Copyright Act* 1968 (Cth).

<sup>9</sup> Accessible at <<http://www.gnu.org/licenses/gpl.html>>. The use of the term GNU in the definition of its own acronym is recursionary, and is a pun playing on the fact that much computer software utilises the process of recursion (where a routine calls itself).

---

## EU: The Bodil Lindqvist Case

---

On 6 November, the European Court of Justice ("ECJ") made a ruling on the scope of the Data Protection Directive (95/46/EC) on the Internet (Case C-101/01, *Bodil Lindqvist*). Of principal interest is that the ECJ ruled that placing of information on a website located in EU is not transfer of data abroad.

Mrs Lindqvist had set up Internet pages on her personal computer, containing information on herself and her colleagues, including their names, a description of work performed by the colleagues as well as their hobbies in mildly humorous terms. She also mentioned that one of her colleagues had injured her foot and was working part-time on medical grounds.

Mrs Lindqvist was fined for processing personal data by automatic means without notifying the Swedish supervisory authority ("Datainspektionen"), for transferring data to third countries without authorisation and for processing sensitive personal data (information of a foot injury and part time work on medical grounds). She appealed to the Göta Court of Appeal, which asked the ECJ seven questions concerning the interpretation of the Directive

95/46/EC of the European Parliament and of the Council of 24 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data (the "Directive").

The ECJ ruled that the act of referring, on an Internet page, to various persons and identifying them by name or by other means, constitutes a processing of personal data wholly or partly by automatic means within the meaning of the Article 3 (1) of the Directive. The ECJ also held that such processing of personal data is not covered by any of the exceptions in Article 3(2) of the Directive (relating to activities falling outside the scope of Community law and purely personal activities). The ECJ confirmed that the reference to the fact that an individual has injured her foot and is on half-time on medical grounds, constitutes personal data concerning health within the meaning of Article 8(1) of the Directive.

The ECJ's interpretation of Article 25 of the Directive, (laying down a prohibition on transfer of personal data to countries outside the EU), has come as a surprise to many. The ECJ held that there is no such transfer within the meaning of Article 25 where an

individual loads personal data onto an Internet page which is stored on an Internet site accessible to people outside the EU. The judgment clarifies that this conclusion only concerns such activities as carried out by Mrs Lindqvist and not those carried out by the hosting providers. The ECJ, however, did not elaborate on this and no definite conclusions can be drawn from the ruling in this respect. It remains to be seen how this judgment will be interpreted in practice.

The ECJ further held that the provisions of the Directive, in themselves, do not conflict with the principle of freedom of expression enshrined in *inter alia* Article 10 of the European Convention on Human Rights and that nothing prevents a Member State from extending the scope of its national legislation implementing the Directive provided that no other provision of Community law precludes it.

*(This article was supplied courtesy of Mats Anderfjärd, Linklaters IT & Communications, Stockholm. Intellectual Property News, Issue 26, December 2003.)*