

# SOURCE MATERIAL

*The issue of ownership is the biggest point of contention for many intent on using open source software.*

Open source development has undeniably changed computing – and the evidence is abundant. Linux now runs over 50% of the world's web servers. IBM claims to have already recouped the \$1 billion it invested in Linux projects during 2001. Apple has based the latest version of the Macintosh operating system on the open source FreeBSD and Mach dialects of Unix. And version six of Netscape's Internet browser is a product of the Mozilla open source project.

But to executives at software giant Microsoft, open source is "a cancer", "an intellectual property destroyer" and "un-American". They, in common with other detractors, accuse open source of "gobbling up intellectual property like Pacman". They point to the abundance of cases where open source projects have ground to a halt, half-finished, after developers fell out and abandoned their projects or quit to work on their own version, as proof of this argument.

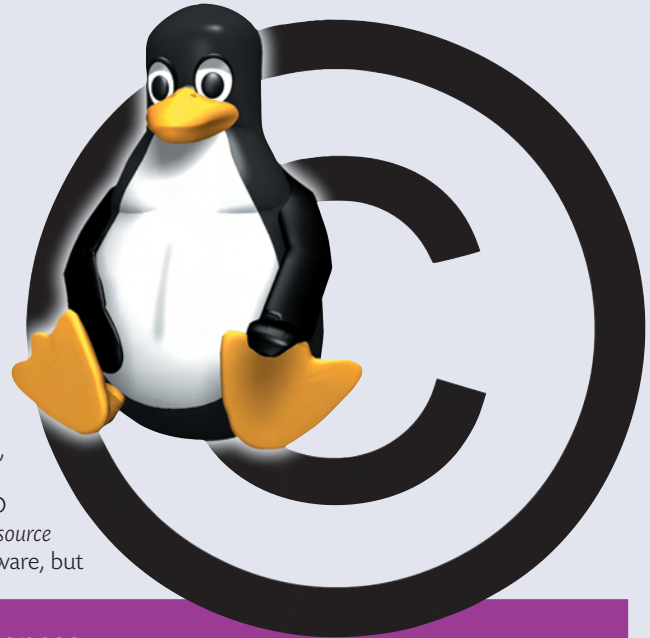
FreeBSD, for instance, started after a dispute between its founders and the originator of 386BSD, which the former were trying to repair.

Businesses have been reluctant to adopt open source software for a number of reasons, including a lack of commercial support services – an issue Red Hat, IBM, and other Linux vendors have attempted to address. But the issue of ownership is the biggest point of contention for many developers and users.

At issue are a series of critical questions the answers to which will ultimately determine the influence of open source within the corporation: What happens if the obtained software contains code the developers had no right to use? Can developers use and amend open source software in their own programs without being forced to offer the end result on an open source basis? And can an open source product's developers change the terms of their licence to demand royalties

from customers if the product needs patches or upgrades? Might they even be able to withdraw it?

The answers frequently depend on what open source licence is used. Microsoft, for example, uses open source code released under the BSD licence (see box, *Open source licences*) in its own software, but



## Open source licences

**WHENEVER** companies talk about open source software, they are usually referring to software released under the GNU general public licence (GPL) used for Linux. However, there are many more licence types that qualify as open source because the software involved is released in source form and can be used and modified for free.

Listed below are the main open source licences. The first four are the 'classic' licences most commonly used for open source software; in addition, the Mozilla Public Licence has now become widely used since its introduction by Netscape in 1998. The GPL, which also requires any derived software to be released under an equivalent licence, is commonly seen as the most restrictive set of conditions, while the BSD licence, which simply asks for a copyright message to be included with any derived software, is seen by many as the least restrictive and is used by many companies, including Apple and Microsoft, as a result.

- The GNU General Public Licence (GPL)
- The GNU Library or 'Lesser' Public Licence (LGPL)
- The BSD licence
- The MIT licence

- The Artistic licence
- The Mozilla Public Licence v. 1.0 (MPL)
- The Qt Public Licence (QPL)
- The MITRE Collaborative Virtual Workspace Licence (CVW Licence)
- The Ricoh Source Code Public Licence
- The Python licence (CNRI Python Licence)
- The Python Software Foundation Licence
- The zlib/libpng licence
- The Apache Software Licence
- The Vovida Software Licence v. 1.0
- The Sun Industry Standards Source Licence (SISSL)
- The Intel Open Source Licence
- The Mozilla Public Licence 1.1 (MPL 1.1)
- The Jabber Open Source Licence
- The Nokia Open Source Licence
- The Sleepycat Licence
- The Nethack General Public Licence
- The Common Public Licence
- The Apple Public Source Licence
- The X.Net Licence
- The Sun Public Licence
- The Eiffel Forum Licence
- The W3C Licence
- The Motosoto Licence
- The Open Group Test Suite Licence
- The Zope Public Licence

its gripe is with the terms of the GPL (GNU general public license), which the Linux community and the majority of open source developers use. This mandates that if a product makes use of GPL code, the completed product must be released under the GPL, the developer must provide the source code for the software for free, and other developers can use part or all of the code in their own products. So the necessity for companies that want to retain a closed source model is to ensure they know exactly what licence the original developers used for their code and ensure that it is compatible with their own business aims.

Alternatively, they can 'ring-fence' their own code – open source licences cannot, by definition, determine the licences of other bundled software. As a result, even the GPL cannot force developers to open source their software if it is merely on the

panied, and had released modified versions of the software, had to withdraw them.

"Source code is a work much like a book written by one or more people," explains Ben Adida, part of the licensing group of the OpenACS community. "An author can decide to distribute source code if they want to, but that does not give the user of the source code any rights other than to make use of it. Users can freely download it and use it, but they cannot create derivative works or redistribute them."

The risks may be small, but they cannot be ignored, believes Joe Buck, a member of the steering committee in charge of the open source compiler gcc. "Some contributor to a project may send something that he has no right to, especially if he works as a programmer in the US," he suggests. "In the worst case, his employer could sue the person

not their copyright – so once their code becomes part of the main package, the project owner will have to ask for their permission to make that part of the package available.

### OPEN AND SHUT CASE?

But other licences are not so stringent and there are projects with few developers. Before Red Hat bought its assets in February 2002, ArsDigita changed the licence for its content management system so it was no longer open source. It was able to do this because it had not used code from any contributors and so held the copyright on the source code. But since ArsDigita released its earlier code under the GPL, it is still legal for people to distribute and modify earlier versions.

Similarly, upgrade paths also disappear when the community developing a project loses interest or the original vendor goes under. "Vendors vanish. That much is a fact of life," says consultant Gary Murphy, who advises on open source strategies. "When a proprietary vendor vanishes, the code vanishes and the applications are screwed. When open source teams disband, no one really notices."

His own web site uses the now-defunct open-source portal system SIPS. "SIPS quickly fell apart as a project. The code was not extensible and... within weeks, it was obvious we were on our own. That was three or four years ago and our site still runs SIPS. Because it was open source, the failure of the 'vendor' did not impact us – we knew the dangers, accepted the risk, and while we'd love to replace it, it works and for a price we can afford."

Open source users share many of the same problems as closed source software users. It may be easier to spot when someone has violated a licence agreement if the software's source is freely available, but major companies are still caught 'borrowing' others' proprietary code.

In 1998, for example, Microsoft was found using code from Apple's QuickTime in its Windows Media Player software, and Trio Systems spotted in 2001 that Adobe had used its C-Index technology in its InDesign program. But if end users have access to the source code, it is far easier for them to overcome these problems. ⓘ

## "When open source teams disband, no one really notices."

same CD, for example, or forms part of a wider package. In contrast, Mozilla uses different licences for its four main modules, since the Mozilla public license (MPL) allows software that links to (rather than modifies) MPL software to be released under different licences. Some developers even avoid the need to publish changes to source code by not directly distributing their software. Instead, they offer executable modules on an application service provider basis.

The drawback to this ability to mix and match licences is that developers who do not check all the licences of the software they are using may make the false assumption that they have rights to code that is actually proprietary. In May 2001, members of the open source community were surprised when the author of the IPFilter firewall bundled with several distributions of Unix amended the licence for his software to state that "modified and derivative work are not permitted without the author's prior consent". The author, Darren Reed, pointed out that he had never released his software under the BSD licence and was just clarifying his existing rights. Those who had assumed the software was available under the same licence as the operating system it accom-

panying the project for financial damages; at minimum, the tainted code might need to be ripped out and work restarted."

There have already been open source projects found guilty of misappropriating code, although most cases have been settled amicably. In 2001, a programmer for the largest Linux distributor, Red Hat, took the source code for FreeBSD's RAID disk system support, removed the copyright notice and annotations, modified it, and submitted it to Linux head Linus Torvalds for inclusion in his operating system. After the original developer found out, Red Hat replaced the copyright notice and re-submitted it to Torvalds.

Buck says the risks are minimised for companies using software available from the Free Software Foundation because the group asks for legal assignments and disclaimers from contributors before incorporating their code. But if an open source project does contain code that it later has to withdraw, end users may find they cannot upgrade because various features they have come to rely on have been removed from the newer version.

Once a project has become open source, it is almost impossible for it to become closed source again. Developers only grant licences to use their patches –

C O N T A C T

Article by Rob Buckley

Email:rbuckley@infoconomy.com