111 Open Source Software- Everything You Wanted to Know But Were Afraid to Ask

Eben Moglen
Chairman
Software Freedom Law Center

Matthew A. Neco
Vice President & General Counsel
Stirling Bridge, Inc./Morpheus

Mark Radcliffe
Partner, Co Chair Technology and Sourcing Practice
DLA Piper Rudnick Gray Cary US LLP

Faculty Biographies

Eben Moglen
Chairman
Software Freedom Law Center

Matthew A. Neco

Matthew A. Neco is vice president and general counsel for Los Angeles based Stirling Bridge, Inc., and its subsidiaries, including most notoriously StreamCast Networks, Inc., dba Morpheus(tm) - the decentralized public peer-to-peer file search and sharing software application developer and distributor.

Before joining Stirling Bridge Mr. Neco worked for several privately held entertainment, media and technology entities, including one he was co-founder of and he operated a law and mediation office in West Los Angeles. He has focused on various aspects of the entertainment industry in film and TV (representing mostly nascent actors, writers, independent producers), the music industry (representing mostly nascent artists, songwriters, producers, and independent labels), and the Internet and new media, as well as reorganizations, work-outs, litigation, and asset acquisitions in the bankruptcy courts.

Mr. Neco received his B.A. from The University at Albany (State University of New York). He received his J.D. from the University of Wisconsin Law School, in Madison, Wisconsin.

Mark Radcliffe

Mark Radcliffe is a partner and co-chair of the technology transfer group of DLA Piper Rudnick Gray Cary USA, LLP in Houston. Mr. Radcliffe's practice focuses on representing corporations in their intellectual property and finance matters. He has worked with many open source companies and is chair of the open source industry group at the firm. He assisted Sun Microsystems in open sourcing the Solaris operating system and drafting the CDDL. He also represents many open source startup companies. He represented Siemens Venture Capital in their investment in MontaVista Software, Inc. He also serves as outside general counsel for the Open Source Initiative and is the chair of committee C for the Free Software Foundation in reviewing GPLv3.

He has spoken on open source issues at the open source business conference, Silicon Valley Association of General Counsel, and O'Reilly open source conference. The National Law Journal named him one of the 100 most influential lawyers in the United States. He is cited in the 2006 edition of Chambers USA: America's Leading Lawyers for Business, ranking first in California for IT and IT outsourcing. He has been listed in the The Best Lawyers in America, as one of Northern California's top 100 intellectual property attorneys in Super Lawyers, San Francisco Magazine, as one of America's leading lawyers for business, and Who's Who Legal publication, The International Who's Who of Business Lawyers. Harvard Law School also designated him a "distinguished alumni." He has been quoted on intellectual property matters in several major publications. Mr. Radcliffe has also published articles in many legal magazines.

He earned a B.S. magna cum laude from the University of Michigan and a J.D. from Harvard Law School.

ACC Association of Corporate Counsel

Why Should In House Attorneys Be Concerned With Free and Open Source Software ("FOSS")?

- FOSS is a part of our daily lives.
- FOSS is not going away.
- Companies are going to continue to use or include FOSS.

Free and Open Source Software is Apart of Our Daily Lives. Consider These Examples:

- Yahoo!
- Mozilla
- Sending e-mail to a Hotmail account
- Apache
- Use of Domain Names
- The Harvard Law Record
- Morpheus™

Free and Open Source Software is Not Going Away

- Developers like the prestige/reputation associated with developing FOSS.
- Companies can create both a FOSS version and an enhanced or superior proprietary version of their software, and use the FOSS version as a form of promotion for the proprietary version.
- Companies can create a business around providing technical support for their FOSS.
- Companies can create FOSS in order to get peer review of the code.
- Companies might create FOSS in order to gain good will.

Companies Are Going to Continue to Use or Include Free and Open Source Software

- Economical
  - Don't have to pay for proprietary software or spend the time and resources in developing your own code.
- Efficient
  - Readily available and usually easy to implement.
  - Already has been tested by many other developers.
- Flexible
  - Most FOSS licenses allow you to modify the code to fit your business needs.

So Its Your Job to Limit Your Company's Risk

- Get familiar with the various FOSS licensing agreements.
- Be active in your company's software development from day one.
- Stay involved!

Get Familiar With the Various Free and Open Source Licensing Agreements

- GPL, LGPL, BSD and Mozilla Public License are just to name a few.
- Not all FOSS licenses are the same and the differences can be significant for your company.

Be Active in Your Company's Software Development From Day One.

- You should know:
  - What software the developers would like to use;
  - What FOSS licensing agreement is associated with that software; and
  - How the developer would like to use the software.
- You can then advise your company as to:
  - Which FOSS to use from a legal perspective, if any;
  - How to comply with the licensing agreement;
  - How to protect your company's proprietary software from spoliation;
  - How to protect your company from violating the licensing agreement pertaining to the proprietary software it licenses; and
  - How to avoid a situation where your company is using multiple FOSS and the licensing agreements are incompatible.

ACC Association of Corporate Counsel

Stay Involved!

- Developers might think they understand the licensing agreement and, therefore, include FOSS without letting you know their intentions. If you don't stay involved your company might:
  - Be subject to liability for failure to comply with the terms of the licensing agreement and/or
  - Have its proprietary software "spoiled" or "polluted." This is BAD!!!
- New versions of the FOSS may be released under a different licensing agreement. This may result in new terms having to be complied with or even spoliation of your company's proprietary software.

SAMPLE OPEN SOURCE CODE POLICY*

1.0 PURPOSE

1.1 Software – *including open source software* -- is subject to copyright protection, the specific bundle of legal rights that the copyright holder owns, and that no one else may exercise without the copyright holder's permission. When a copyright owner grants permission to another party to exercise one or more of these rights, the holder is granting the other party a license. A software license is a contract which may, depending on its terms, impose affirmative obligations on the licensee, as well as restrict what the licensee may do with the software. The act of downloading, installing or using open source software constitutes an agreement to the terms of the license that comes with the software.

1.2 It is important to note that, contrary to popular belief, *open source software is not in the public domain*. A public domain program is one under which the author has deliberately surrendered proprietary rights, including all copyrights. Since all proprietary rights to public domain software have been relinquished, no license (i.e. permission to use) is required for its use. Thus, programmers may use public domain software as they see fit, without restrictions imposed by the copyright holder (because there is no copyright holder). In the case of open source software, however, the author retains proprietary rights, including all copyrights, and is granting licensees the power to exercise only those rights specifically spelled out in the license accompanying the software.

1.3 The terms of some of these accompanying licenses carry very unsavory consequences, making the use of some open source software inherently risky, especially when open source software is or may be used in products that are distributed beyond internal company use. For example, the vast majority of licenses granting the right to exploit open source software do not include the warranty protections customarily given for commercial products and in fact disclaim them. This is particularly problematic because most open source software is developed without the usual controls present in the commercial software development process. Thus, if a programmer downloads an open source program to which a previous user has added infringing code, the programmer would unknowingly be exposed to liability for infringement, potentially resulting in an injunction or legal damages. Ignorance is not a defense to copyright infringement, only a defense to *willful infringement*.

1.4 Another common consequence of using open source software is known as "GPL taint." One of the most ubiquitous open source licenses, the GNU General Public License ("GPL") contains terms which mandate that modifications to code derived from the GPL code be distributed under the same terms as the original GPL open source software. In this way, the license accompanying the GPL software turns the other software of the licensee (not derived from the GPL open source software) into GPL software when the two are combined. The consequences of GPL taint are potentially severe. A company could lose rights to its proprietary code, could be forced to disclose its trade secrets, and might even lose the right to the exclusive use of its own underlying code since that code must be disclosed in source code publicly.

1.5 Additionally, each open source license has requirements which must be met in order for a programmer to use the licensed software. Some licenses have more requirements than others. Failure to pay close attention to these requirements and to fulfill them all may result in the voiding of the license. Use of the licensed software under these circumstances would constitute copyright infringement.

1.6 In order to avoid these and other legal risks inherent in the use of open source software, Cardinal Health Clinical Technologies and Services is publishing this Policy specifying permissible and impermissible uses of open source software in the products and services we distribute to customers and the manner in which to comply with open source licenses as well as a method (i.e. decision tree) for evaluating the business impact and technical requirements prior to using a particular open source software in a product that will be distributed to customers.

DEFINITIONS

1.7 **Open Source** – The term "open source" software refers not only to the software itself, but also to the terms under which the copyright owner of the software permits others to copy, modify and distribute that software. Generally speaking, referring to a software program as "open source" means that the license to that software meets the following criteria: (i) the software is available for free redistribution; (ii) software includes source code, and allows distribution in source code as well as compiled form; (iii) the license allows modifications and derivative works, and allows those modifications and derivatives to be distributed under the same terms as the license of the original software; (iv) the license permits distribution of software built from modified source code; (v) the license does not discriminate against any person or group of persons; (vi) the license does not restrict any one from making use of the program in a specific field of endeavor; (vii) those rights apply to all to whom the program is redistributed without the need for execution of an additional license by those parties; (viii) those rights do not depend on the program's being part of a particular software distribution; (ix) the license does not place restrictions on other software that is distributed along with the licensed software; and (x) no provision of the license is predicated on any individual technology or style of interface. A comprehensive, industry-wide accepted definition of "open source software" is copyrighted by the Open Source Initiative and provides a detailed explanation of each component of the definition summarized above.

1.8 **Derivative Work** - A Derivative Work is a work of authorship based upon one or more preexisting works, such as a translation, abridgment, condensation, or any other form in which a work may be recast, transformed, or adapted. A work consisting of revisions, annotations, elaborations, or other modifications which, as a whole, represent an original work of authorship, is a 'derivative work'. (Also see 17 U.S.C. §101, et seq.)

1.9 **License Category Breakdown** – Based upon this industry-accepted definition of "open source" software, the open source software being used or proposed to be used by the development teams at Cardinal Health Clinical Technologies and Services have been placed into one of the following four categories: (1) Restrictive; (2) Restrictive Hybrid; (3) Permissive; and (4) Non-Open Source License("Non-OSL").

1.9.1 **Restrictive Licenses** –"Restrictive licenses" are those open source licenses which place the heaviest limitations on initial and subsequent licensees. The primary example of a restrictive license is the GNU General Public License ("GPL") which has the distinction of being the first open source license. This license restricts licensees from licensing modifications under any license other than the GPL. This means that derivative works of software derived from the GPL open source software must always remain "open". Furthermore, the GPL prohibits licensees from combining software derived from the GPL open source software with proprietary software or with open software under a license incompatible with the GPL.

1.9.2 **Restrictive Hybrids** – "Restrictive Hybrid licenses" are less restrictive than the GPL. Restrictive Hybrid licenses, however, still limit the license under which derivative works may be distributed. The GNU Lesser Public License ("LGPL") for example is less restrictive than the GPL since it allows licensees to link software derived from the LGPL open source software with other software which may be open or proprietary. Yet, the LGPL does require that copies and derivative works be licensed under either the LGPL or the GPL. The Mozilla Public License ("MPL") is another example of a Restrictive Hybrid license. Like the LGPL, it allows licensees to combine software derived from the MPL open source software with other open or proprietary software; however, it also requires that derivative works be licensed under the MPL.

1.9.3 **Permissive Licenses** – Permissive licenses do not have the limitations present in the above licenses. They are all modeled after the Berkeley Software Distribution license ("BSD") which allows licensees to license derivative works however they please. This means that BSD licensees may modify open BSD software and make that derivative proprietary or "closed." BSD licensees may also combine software derived from the BSD open source software with other works as they see fit. In other words, these licenses have little to no restrictions other than a few general conditions such as requiring the licensee to include copyright notices and disclaimers.

1.9.4 **Non-Open Source Licenses (Non-OSL)** – These licenses do not fall within the industry-accepted definition of open source software (reproduced above) and, therefore, fall into their own category. Generally these licenses prohibit licensees from copying and/or distributing source code or from creating derivative works from the original licensed work and distributing the derivative works.

1.9.4.1 Both "freeware" and "shareware", while often only available in object code, are sometimes available in source code. Given that freeware and shareware rarely meet the requirements to constitute true open source software, they are considered Non-OSL.

1.9.4.2 "Freeware" is software that is available for download and unlimited use, free of charge but only for personal use. Although freeware is available for free, the author retains the copyright, which means that it cannot be altered or sold.

1.9.4.3 "Shareware" is software that is available on a free limited trial basis. Sometimes this is a fully featured product, other times it lacks some of the features of the commercial version. Shareware software is distributed on an honor system, distributed without charge for an evaluation period but requiring payment if the licensee continues to use the software beyond the evaluation period. After paying the registration fee, licensees are often eligible for technical assistance and updates.

© 2006 Cardinal Health, Inc., and/or one of its subsidiaries

RESPONSIBILITIES

1.10 **Research & Development Management** – Responsible for ensuring this procedure is adhered to by the company.

1.11 **Legal Department** – Responsible for assisting with requirements.

PROCEDURE

OPEN SOURCE LICENSE DECISION TREE

STEP 1

Is the license an Open Source License within the definition in Article I, Section A?

- If Yes, then the license is covered by this Policy. Proceed to **Step 2**.
- If No, then the license is not covered by this Policy. Determine the rights and obligations of a licensee under the license agreement terms itself and comply. Contact the legal department for assistance, if required.

STEP 2

Determine under which category (described in Article I, Section B) the license fits.

- If the license is Restrictive, proceed to **Step 3**.
- If the license is Restrictive Hybrid, proceed to **Step 4**.
- If the license is Permissive, proceed to **Step 7**.

STEP 3

Do not download software licensed under a Restrictive License. Contact the Legal Department before accepting the terms of any Restrictive License.

STEP 4

Is the license a LGPL license?

- If Yes, go to **Step 5**.
- If No, proceed to **Step 6**.

STEP 5

Are you merely linking your software to the LGPL software?

- If Yes, identify obligations and comply.
- If No, **do not download software licensed under the LGPL. Contact the Legal Department** before accepting the terms of the LGPL if you are integrating the LGPL software into proprietary software.

STEP 6

Do not download software licensed under a Restrictive Hybrid License. Contact the Legal Department before accepting the terms of any other Restrictive Hybrid License.

STEP 7

For Permissive Licenses, identify the conditions set forth in the license and comply. Such conditions may include, but are not limited to: retaining all copyright and attribution notices, providing copies of the license, and providing notice of modified files.

REFERENCE DOCUMENTS

The following reference documents are applicable to this document.

Document Storage Number or Location	Title
	License Comparison Chart

© 2006 Cardinal Health, Inc., and/or one of its subsidiaries

License Comparison Chart

Explanation of certain columns in the License Comparison Chart:

DWS REQUIRED TO REMAIN OPEN? THE COLUMN ENTITLED " DWS REQUIRED TO REMAIN OPEN?" DESIGNATES WHETHER THE LICENSE REQUIRES DERIVATIVE WORKS ("DWS") CREATED FROM THE LICENSED SOFTWARE TO REMAIN OPEN. FOR INSTANCE, THE GPL LICENSE REQUIRES THAT ALL DWS CREATED USING GPL SOFTWARE MUST BE DISTRIBUTED UNDER THE GPL LICENSE. THE DEVELOPER OF THAT DWS IS NOT PERMITTED TO MAKE THE DW PROPRIETARY (I.E., MAY NOT BE TAKEN PRIVATE) SINCE THE GPL LICENSE REQUIRES THE DW TO REMAIN OPEN. A "YES" IN THIS COLUMN INDICATES THAT THE DW MUST BE KEPT OPEN.

INTEGRATION PROHIBITION? THE COLUMN ENTITLED " INTEGRATION PROHIBITION?" DESIGNATES WHETHER THE LICENSE PROHIBITS THE LICENSED SOFTWARE FROM BEING COMBINED WITH CLOSED/PROPRIETARY SOFTWARE. A "YES" IN THIS COLUMN INDICATES THAT THE LICENSE CONTAINS SUCH A PROHIBITION.

DISTRIBUTION OF OBJECT CODE OR EXECUTABLE FORM? THE COLUMN ENTITLED " DISTRIBUTION OF OBJECT CODE OR EXECUTABLE FORM?" DESIGNATES WHETHER THE LICENSE PLACES ANY RESTRICTIONS ON THE DISTRIBUTION OF OBJECT OR EXECUTABLE CODE. THE TEXT IN THIS COLUMN WILL EITHER SPELL OUT ANY RESTRICTIONS THE LICENSE PLACES ON SUCH A DISTRIBUTION, OR IT WILL STATE "NO RESTRICTIONS" IF THE LICENSE HAS NO SUCH CONDITIONS.

LICENSE NAME	CATEGORY	COMPATIBLE W/ GPL?	DWS[1] REQUIRED TO REMAIN OPEN?	INTEGRATION PROHIBITION?	DISTRIBUTION OF OBJECT CODE OR EXECUTABLE FORM?	UNIQUE FEATURES	RESTRICTIONS/ REQUIREMENTS
Apache License Version 2.0 http://www.apache.org/licenses/LICENSE-2.0.html	Permissive	Likely Not (b/c allows licensee to provide additional or different license terms)	No	No	No restrictions	licensee's modifications may provide additional or different license terms and conditions for use provided licensee's use otherwise complies with License	To reproduce and distribute copies or DWs LICENSEE must: - Provide copy of License - Provide notice of modified files stating they have been changed from original - Retain all CR[2], patent, trademark, and attribution notices
BSD http://www.opensource.org/licenses/bsd-license.php	Permissive	Yes	No	No	No restrictions	Arguably one of the most permissive Licenses	Redistributions must retain CR notice, a list of the License's conditions, and a disclaimer No Endorsement: No use of UCB's name or contributors' names for endorsement/promotion
GNU GPL http://www.gnu.o	Restrictive		Yes (and must be licensed	Yes	Ok provided licensee does	- First OSL - Most Restrictive	Copies: To distribute copies

[1] Derivative Work
[2] Copyright

© 2006 Cardinal Health, Inc., and/or one of its subsidiaries

LICENSE NAME	CATEGORY	COMPATIBLE W/ GPL?	DWs[1] REQUIRED TO REMAIN OPEN?	INTEGRATION PROHIBITION?	DISTRIBUTION OF OBJECT CODE OR EXECUTABLE FORM?	UNIQUE FEATURES	RESTRICTIONS/ REQUIREMENTS
rg/copyleft/gpl.html			under GPL only)		one of the following: - provide complete source code, - provide 3yr offer to provide source code, or - provide the information licensee received as to the offer to distribute source code (provisions of source code must be under GPL license)	- Viral, all copies and modifications must be licensed under GPL - licensees cannot add additional restrictions on licenses for modified work	LICENSEE must: - Conspicuously publish (1) CR notice and (2) warranty disclaimer - Keep all notices referencing the GPL and the absence of warranty intact - Provide copy of GPL Modifications: To distribute DWs licensee must: - Provide prominent notices of modification - License whole work under GPL (i.e. must make source code of DW open) - Interactive works must display CR notice, warranty disclaimer and copy of GPL
GNU LGPL http://www.gnu.o	Restrictive Hybrid	Yes	Yes (and may be licensed	Can be linked to non (L)GPL	Ok under terms of	DWs can be licensed under	Copies: To distribute copies

LICENSE NAME	CATEGORY	COMPATIBLE W/ GPL?	DWs[1] REQUIRED TO REMAIN OPEN?	INTEGRATION PROHIBITION?	DISTRIBUTION OF OBJECT CODE OR EXECUTABLE FORM?	UNIQUE FEATURES	RESTRICTIONS/ REQUIREMENTS
rg/copyleft/lesser.html			under LGPL or GPL) Note: software which merely link to LGPL'd software do not have to be (L)GPL'd just DWs	software (which may be free or proprietary)	License if licensee provides complete source code and releases object code under same terms in Section 1 and 2 of License. If distribution of object code is made by offering access to copy from a designated place then offering equivalent access to source code is ok.	LGPL or GPL	licensee must: - Conspicuously publish (1) CR notice and (2) warranty disclaimer - Keep all notices referencing the LGPL and the absence of warranty intact - Provide copy of LGPL Modifications: To distribute DWs licensee must: - Provide prominent notices of modification - License whole work under LGPL or GPL (i.e. must make source code of DW open) - Interactive works must display CR notice, warranty disclaimer and copy of LGPL
MIT (Expat License) http://www.opensource.org/licen	BSD-Style Permissive	Yes	No	No	No restrictions	Just like BSD except no Non-endorsement clause	To copy, modify, and distribute work licensee must:

LICENSE NAME	CATEGORY	COMPATIBLE W/ GPL?	DWs[1] REQUIRED TO REMAIN OPEN?	INTEGRATION PROHIBITION?	DISTRIBUTION OF OBJECT CODE OR EXECUTABLE FORM?	UNIQUE FEATURES	RESTRICTIONS/ REQUIREMENTS
ses/mit-license.php							- Include CR notice and permission notice in copy or DW distribution.
Mozilla Public License http://www.mozilla.org/MPL/MPL-1.1.html	Restrictive Hybrid	No	Yes (code copied or modified must stay under MPL)	No (may be combined with proprietary files)	Ok but must make source code available under terms of License on same media as executable version or via an accepted Electronic Distribution Mechanism (must remain available for at least 12 months)	-Requires inclusion of a Legal.txt file describing third party intellectual property claims to covered/original work	Copies: To distribute copies of work licensee must: - distribute under MPL only - include copy of MPL - not offer or impose new License terms(but may offer additional rights) - duplicate the notice in MPL's exhibit A in each file of the source code Modifications: To distribute modifications of work licensee must: - distribute under MPL only - include file stating origin of work, that work has been changed and documenting changes made - include

LICENSE NAME	CATEGORY	COMPATIBLE W/ GPL?	DWs[1] REQUIRED TO REMAIN OPEN?	INTEGRATION PROHIBITION?	DISTRIBUTION OF OBJECT CODE OR EXECUTABLE FORM?	UNIQUE FEATURES	RESTRICTIONS/ REQUIREMENTS
							Legal.txt file describing all known third party rights re: work - duplicate the notice in MPL's exhibit A in each file of the source code

[*] Thanks to Hillary Wilson of Cardinal Health, Inc. for these materials.

© 2006 Cardinal Health, Inc., and/or one of its subsidiaries

Free and Open Source Software: Risks Your Company's Software Developers Might Not Be Aware Of[1]

By Matthew A. Neco and Wendy Millar Goodkin[2]

Depending on the size of the company and legal department you work for as in-house counsel you might think you are aware of almost every contract and licensing agreement that your company has become a party to since you started with the company. You might think that you or some other lawyer has reviewed and advised on every agreement. But if your client is a software development company or a company that has an information technology department that modifies or tweaks software your company owns (that is, that your company developed) or licenses as a licensee, you might be in for a little surprise. The fact of the matter is that there may be licensing agreements that have been entered into on behalf of your client company without the legal department even knowing it.

No, it's not that your CEO and business development teams are signing contracts before a lawyer reviewed and advised. (Well, maybe they are.) But there are threats you might not have thought about. They are your company's very independent thinking software developers, engineers, and IT staff entering into agreements governing software licensed as Free and Open Source Software ("FOSS") and then using the FOSS in certain ways. Engineers are logical thinkers, so if you take the time to explain it to them they'll get it. Unless you've got them well educated and sensitive to certain issues involving FOSS they may be posing silent threats to your client, entering into software licensing agreements in a willy nilly fashion.

This may have profound implications for proprietary software owned by you company or which your company may be licensee to. Your first thought might be that there can't be much harm in software that's "Free" and "Open," other than thinking You get what you pay for: If it's free how good can it be? You might worry about trojans, back-doors, root-kits, hidden key-stroke loggers, or other malware or spyware issues if the particular piece of OSS or derivatives come from unsavory characters. (We aren't impugning the OSS development community. The vast majority of this community is comprised of serious engineers who would never include any malicious code.) You think you'll leave it up to the engineers and IT staff to deal with all the bugs, and other existing or potential program problems that must be in this free, unwarranted, software. Otherwise, you think, if the engineers do their job, even though most FOSS licenses disclaim any warranties and liability, it should be safe for your company to use FOSS. You might be wrong. Unlike software that has truly been released into the public domain and is, therefore, not protected under copyright or patent, FOSS software is typically subject to copyright law and licensing agreements. Licensing agreements that might require your company to abide by or perform certain obligations. Licensing agreements that might, inadvertently or otherwise, expose your company's proprietary software (or software that it has licensed as licensee) to what is pejoratively known as spoilation (spoliation in the litigation world), infection or pollution.

What are the possible results of infection, spoilation or pollution? If your client has developed and owns proprietary software, or is the licensee of non-FOSS, which is then "integrated" with some FOSS licensed under some FOSS licenses, and distributed at any time, your company might be required to disclose or publish some or all of its integrated proprietary code. Your client might be required to make its software code available to the world for free under the same license that governed the FOSS used by the engineers. There go the trade secrets. There goes the software product your company sells. There goes your job.

With potential risks in mind, you may be thinking that the best advice would be to warn your company that it shouldn't ever use FOSS. This advice just might be bad advice too. FOSS can offer your company many business advantages. First, FOSS is economical in that your company will not need to spend money on licensing another company's proprietary software or spend valuable time and resources developing and testing its own software, or components for software that it owns or is licensee of. Second, FOSS is readily available. All you need to do is conduct a search using your favorite search engine or Morpheus™ [3] to see how easy it is to find

[1] The article and the contents herein are meant to provide the reader with general information. Nothing herein shall be construed as legal advice.

[2] Matt is Vice President & General Counsel and Wendy is Staff Attorney of Stirling Bridge, Inc. and its subsidiaries including StreamCast Networks, Inc. (dba Morpheus™). The opinions, viewpoints, warnings, etc., expressed herein are solely those of Matt and Wendy, and are not those of Stirling Bridge, StreamCast (except, perhaps at Footnote 3), or any other client of Matt or Wendy.

[3] Yes, that's a shameless little plug. And here's an anti-inducement warning: Do NOT use Morpheus™ to download content that is not authorized by the rights holders for free downloading, or is not otherwise authorized for downloading. Copyright and patent infringement are not condoned. This article is not intended to actively induce copyright infringement or the violation of <u>any</u> law or the rights of any person or entity.

and download FOSS. In addition, unlike developing your own software, FOSS may have been read, tested, tried, redistributed, improved, and modified by many software developers, resulting in higher quality software than your company could produce on its own, though it almost always comes disclaiming any warranties and liability. Lastly, FOSS is typically somewhat flexible. Most FOSS licenses allow your company to modify the code to fit its business needs. Although your company must be careful as to what is required of it when such a derivative work is made from modifying the software[4], the benefits are something that may not typically be available when you are a licensee of proprietary software either because it would be a violation of the licensing agreement or because proprietary software only includes the object code (as compared to FOSS which usually includes both the object code and the source code[5]) which does not provide the information necessary for a developer to alter or modify the software.

So now that you know that your client may be using FOSS and you can't simply ignore it, or have it go away, what can you do? Well, it's time to get educated and talking. Visit http://www.opensource.org/ and review the numerous FOSS licenses that are available. Remember not all FOSS licenses are the same and the differences can be significant for your company. For example, Section 2 of the Gnu General Public License v2 ("GPLv2") states:

> 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. (Emphasis added.)[6]

Therefore, integrating your company's software with FOSS licensed under GPLv2 may "infect," "pollute," or "spoil" your company's proprietary software requiring your company to disclose or publish some or all of what it thought was its proprietary code if such software is

[4] Whether modifications constitute derivative works has some relevance. The draft of GPLv3 would apply when modifications are made, a lesser standard than determining what constitutes a derivative work. GPLv2 applied when derivative works are created. This distinction may prove to cause some controversy.

[5] Open Source Software includes both object code and source code to be considered open source software, although the source code may be available by well publicized means rather than disclosed with the FOSS itself, see http://www.opensource.org/docs/definition.php and http://www.gnu.org/philosophy/free-sw.html. Source code is computer program in its original, human-readable form, and may be written in any one of many possible computer languages including Java, C++, Pascal, Basic, Unix. Source code is turned into object code, binary code (ones and zeroes) which can be used by a computer in different ways depending on whether the language is compiled or interpreted. In the proprietary world end-users are typically not able to view the original program (source code) and are, therefore, unable to alter it. In the FOSS realm the source code is available to all so that they can compile the application themselves or alter it.

[6] The GNU General Public License (GPL), Version 2, June 1991, Copyright (C) 1989, 1991 Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA (See http://www.opensource.org/licenses/gpl-license.php for GPL License)

distributed or published outside of the strict confine of the single client that licensed, used and modified the FOSS. A funny thing about software is that your client might not have a present intention to distribute software that contains FOSS and that it modifies, however, for a variety of reasons that intention might later change. Does your client have an affiliate or parent entity that it allows to use the software? Are certain assets of your client, including the modified FOSS sold, acquired, or used, or is there a merger, and does any of that that constitute an assignment or distribution? And so the issue of what does or may constitute a distribution must be considered.[7]

After you are familiar with the various licenses, you should talk to your company's developers and create and circulate a FOSS Policy. You should know what software the developers would like to use, what licensing agreement(s) may be associated with that software, and how the developer would like to use that software – including how **modular**, or tightly "**integrated" or intertwined** the FOSS is likely to be with the proprietary software. What constitutes "***mere aggregation***" (see bold italics, above) versus this idea of integration is a legal decision, and there's not a lot of precedential case law out there. Yet. And you don't want to be a test case. However, to help you and the developers to make this determination the Free Software Foundation ("FSF") has suggested that you look at the **mechanism of communication**, including whether the modules are included in the same executable file (suggesting integration), whether the modules are **designed to run linked together in a shared address space** (suggesting integration), or whether there are pipes, sockets and/or command-line arguments used to communicate between the proprietary software and FOSS (suggesting "*mere aggregation*"). In addition, you will also need to discuss with the developers what type of information is being exchanged. For example, if there is an exchange of complex internal data structures this could be a basis for showing integration.[8]

Once you have that information, you then have the knowledge to advise you company, from a legal perspective, whether FOSS is right for it, or advise utilizing the FOSS in a more modular – merely aggregated - manner, rather than tightly intertwined or integrated, so as to attempt to minimize the risks of spoilation. If the business decision is to use FOSS, then you will need to advise regarding how to comply with the licensing agreement, how to use FOSS in conjunction with the company's proprietary software to avoid spoilation. However, it is not just your client's proprietary software that you need to be concerned about. If your client is the licensee of someone else's software that might be proprietary in whole or in part your client might be violating the licenses by incorporating FOSS with the licensed proprietary software. In addition, if your client is already the licensee of FOSS, your need to review the license to ensure that any future FOSS licenses will not be incompatible with the first.

And don't forget, you can't just talk to the developers and walk away. You must stay involved. If you don't stay involved the developer may have already included open source software thinking that he/she understood the implications. In addition, new versions of the FOSS may be released under a different licensing agreement, thus, requiring new and different compliance requirements.

# # #

[7] What should also be considered is what impact using FOSS might have on any possible mergers, acquisitions or asset sales in the near or distant future: whole cottage industries are developing around software review to determine whether FOSS has been used, what FOSS has been used, and how it has been used, delaying deals, driving up costs, and sometimes causing proposed acquisition prices to decrease.

[8] See, http://www.gnu.org/licenses/gpl-faq.html

© 2005 - 2006 M. NECO & W. GOODKIN

Open Source

Mark Radcliffe

DLA Piper Rudnick Gray Cary USA LLP

mark.radcliffe@dlapiper.com

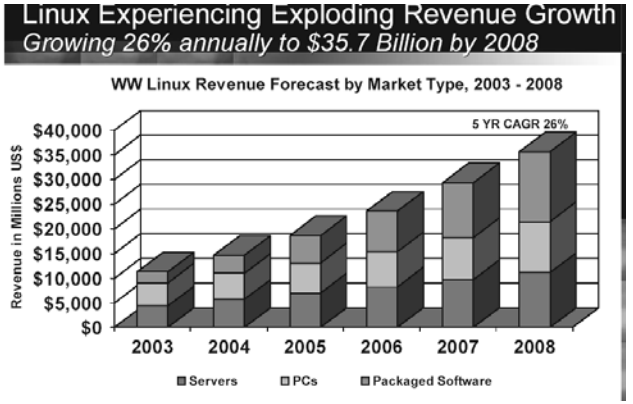
DLA PIPER RUDNICK GRAY CARY US LLP

- **Second Largest Law Firm in the World**
  - 3100 attorneys
  - 22 countries/59 cities
- **Strong Intellectual Property Practice (165 lawyers):**
  - 40 IP Transactional Lawyers
  - 45 Patent Prosecution Attorneys
  - 80 IP Litigation Attorneys
- **Strong Open Source Practice (5 partners):**
  - Assisted Sun with Open Solaris
  - GC of Open Source Initiative/Chair of Committee C for GPL Review
  - Represent SugarCRM, Zimbra, Qlusters, Icesoft, Socialtext, Univa
- **Strong Mergers & Acquisition Practice (274 lawyers)**
  - 2005: 3rd in US and Globally in completed M&A deals
  - 274 deals in US and 311 globally

Critical Open Source Issues

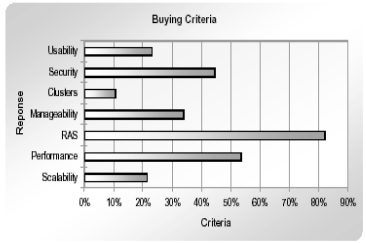
- **Open source software is becoming ubiquitous (despite what your CTO/CIO tells you): television sets to semiconductors**
- **Increasing demand by customers to know what is in the product**
- **Compliance problems**
  - GPL, most widely used OS license, is very ambiguous
  - GPL terminates automatically
- **Incompatibility of licenses (frequently referred to as "license proliferation")**
  - Software distributed under the GPL cannot be used with software distributed under the MPL/CDDL
- **For projects like Linux, multiple licensors with potentially multiple interpretations of the license**
- **Patent infringement: patent "trolls" and Microsoft (Intellectual Ventures)**
- **M&A: New issue (30% price reduction by IBM for Think Dynamics)**
- **Special role of the OS community**

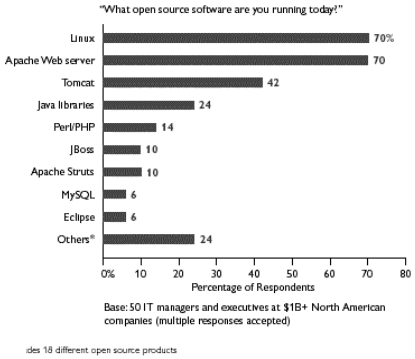
Linux Adoption – (IDC)



Open Source Adoption

Open Source Adoption





ACC's 2006 Annual Meeting: The Road to Effective Leadership October 23-25, Manchester Grand Hyatt

Applications – Horizontal/Infrastructure

- **Database** – MySQL, PostgreSQL, Firebird, Ingres, Sleepycat, MaxDB, SQLite
- **Middleware** – JBOSS, Jonas (ObjectWeb), Geronimo, BeeHive, Apache, Resin
- **ERP/CRM** – Sugar, Compiere, OpenERP, ERP5
- **Tools** – Zend, Exadel, Eclipse
- **Management** – Cassatt, Open Country, IT Groundworks (Nagios)
- **Content Management** – Zope, Plone, Mambo
- **Business Intelligence** – JasperReports, BIRT
- **Productivity** – OpenOffice, Firefox, Gnome
- **Telecom –** Synch4J(Funambol), Asterisk(Digium)

Open Source Eco-System: Community

Non profit entities play an important role in the open source community

- **Free Software Foundation (FSF)**
  - Developed General Public License (GPL)
- **Open Source Initiative (OSI)**
  - Approves licenses as "open source"
- **Open Source Development Lab (OSDL)**
  - Commercial open source trade association
- **Apache Foundation**
  - Key open source community – developed and run by the community
- **Eclipse Foundation**
  - Less than two years has become a standard - commercially supported

Commercial Eco-system

- **OEMs**
  - IBM, Sun, SGI, HP, Apple
- **ISVs**
  - Oracle, SAP, CA, Adobe
- **Service providers**
  - IBM, IGS, CSC
- **Open source companies**
  - Redhat, MySQL, JBOSS, Sleepycat, Trolltech, MontaVista
- **Technology consumers**
  - Wells Fargo, Charles Schwab, Cargill, Google, Best Buy
- **Investment community**
  - VCs, strategic investors
- **Startups and emerging growth**
  - Zimbra, SugarCRM, JasperSoft, Zend, Laszlo

Elements of an Open Source Strategy

- **Corporate strategy – Is open source an enabling technology or business model?**
  - Business model
  - Product strategy
  - Licensing model
  - Community strategy
  - Implementation/Go-to-market
- **Special considerations**
  - External
    - M&A
    - Partnering and channels
  - Internal
    - Research and development
    - Sales and marketing
    - Compliance

Reasons for an Open Source Policy

- **Role of a policy**
  - Manage risk
  - Ensure strategic flexibility
- **Unusual OSS risks**
  - Automatic termination of GPL
  - Uncertain scope of GPL
  - Broad scope of patent termination in MPL
  - Forking of code
- **Customers are demanding to know what is in your product**
- **Compliance important for financings/M&A**
- **IT staff turn over and difficulty of following up**
- **FSF undertakes 50 enforcement actions a year (Cisco: Linksys)**

Open Source Legal Myths (Courtesy of Karen Copenhaver of Black Duck Software, Inc.)

- **You cannot use open source software in a proprietary environment [or you will die].**
- **All open source licenses require the release of source code for everything.**
- **The easiest answer is to "just say no."**
- **None of these agreements are enforceable so it doesn't really matter anyway.**
- **No one will ever know.**
- **If I even begin to think about all of these obligations, I will give up. To survive, you have to accept some risks and just move on.**

Critical Issues in Open Source Policy

- **Role of OSS: internal/customer/products**
- **Different approvals for different uses**
- **Effect on other strategies: patent "peace" provisions**
- **Criteria for selecting**
  - Testing of OSS
  - Strength of "project"/company
  - Strength of community
- **Audit**
- **Education**
- **Implementation**
  - Internal use
  - Third party licensors and contract developers
  - M&A

Open Source Licenses

- **FSF v. OSI**
- **OSI Approved Licenses: 50+ licenses**
  - BSD/label
  - Reciprocal
  - Other
- **Important Licenses**
  - GPL
  - LGPL
  - MPL
  - Apache
  - BSD
- **OSI**
  - Developing categories
  - Reducing license proliferation
  - License selection wizard

General Public License: GPLv2

- **Contains a great deal of political dialogue**
- **Scope of "based on" work**
  - Ambiguity of "derivative work"
  - Use of "collective work"
  - Linking issues
- **Disclaimer of all warranties**
- **Disclaimer of liability**
- **Patent license: uncertain**

General Public License: GPLv2

- **Never enforced in US court (two victories in Germany)**
- **Potential problems**
  - Scope of "based on" (definition of derivative work/collective work)
  - Many potential enforcers for some projects, like Linux
  - Ambiguity of treatment of patents
  - Lack of choice of law
  - Legal effect of FAQs
  - Automatic termination
  - No choice of law

ACC's 2006 Annual Meeting: The Road to Effective Leadership

October 23-25, Manchester Grand Hyatt

GNU GPLv3

- **First Published in January 2006: Open Review Process**
  - First revision June 2006
  - Final version January 2007
- **Definition of source code to be distributed ("Corresponding Source Code")**
- **Patents**
  - Shift from direct license to covenant not to sue (rev 1)
  - Pass through of third party patent licenses if used with "knowledge"
- **Scope: shift from derivative work to "modification"**

GNU GPLv3

- **Modification to permit compatibility with certain other license**
  - Warranties
  - Trademark use/attribution
  - Broader patent retaliation
  - Prohibition of trademark use
  - Web services: functioning facilities
    - Potential major issue for "behind the firewall" applications provided as services (Google)
- **Prohibition of use of license for DRM**

ACC's 2006 Annual Meeting: The Road to Effective Leadership

October 23-25, Manchester Grand Hyatt

Lesser General Public License: LGPL

- **Very similar to GPL with limited exception for "linking"**
  - Specifically designed for "libraries"
  - Modifications to library treated the same as GPL
- **Section 5 exceptions for "small uses"**
  - Data structure layouts/small macros/inline functions
- **Scope 6 (linked LGPL program)**
  - Permit modifications for customers own use
  - Make source code or object code available

Mozilla Public License

- **Reusable**
  - Used for Zimbra/SugarCRM as well as Mozilla
- **Reciprocal**
- **Scope based on files (with some ambiguity)**
- **Numerous notice requirements**
- **Patent termination (patent "peace")**
  - Patent suit against another contributor for contributions to project
  - Major potential problem: patent suit against software/hardware/device (not OSS contribution) of contributor

M&A: New Issues

- **M&A Open Source Issues**
  - A few have been discussed publicly
    - Cisco: Linksys
    - IBM: Think Dynamics (30% reduction in price)
  - Many more problems have occurred that have not been discussed
- **M&A Risks**
  - Lose deal
  - Delay deal
  - Reduced price

Conclusions

- **Open source is established and growing**
- **Open source is fundamentally disruptive**
- **Consumer and vendors of software should have an open source strategy; many major companies such as IBM, Sun, HP, Wells Fargo and Charles Schwab already have such a strategy**
- **Many critical issues remain uncertain**
  - Who will control the definition of "open source"
  - What business models will be successful
  - Role of governments
  - Response by Microsoft and other proprietary vendors
  - Enforcement of licenses
  - Acceptance of GPLv3

For additional resources on this session's subject matter please visit: http://gplv3.fsf.org