



**Tuesday, October 20**  
**4:30 pm–6:00 pm**

## **808 Open Source – Latest Developments**

**Gemma M. Dreher**  
*Senior Staff Attorney*  
BAE Systems IESI Sector

**James Markwith**  
*Senior Attorney*  
Microsoft Corporation

**Jeffrey Ross Stern**  
*Executive Director*  
Morgan Stanley

**Robert Tiller**  
*Vice President and Assistant General Counsel*  
Red Hat, Inc.

## Faculty Biographies

### **Gemma M. Dreher**

Gemma M. Dreher, Esq. is senior counsel for the BAE Systems Electronics Solutions in Nashua, NH. She provides advice concerning the legal matters of the business, and represents the business in adversary proceedings. She serves as a consultant to management and external spokesperson for the organization on major matters pertaining to its policies, plans, and objectives. Ms. Dreher has worked with BAE Systems engineering management to establish an automated system for license review and has written policies on the management of open source software for the business. Recently, she received an Engineering Collaboration award for her work in the area of open source software management. Ms. Dreher has developed a litigation risk assessment plan for the legal department. She has also developed an outside counsel protocol and an internal file management system for the legal department.

Prior to BAE, she was vice president and general counsel for Pragmatech Software, Inc.

She speaks frequently on the subject of intellectual property management, licensing technology to the federal government, and open source software licensing.

Ms. Dreher received her BA from Regis College and a JD from Northeastern University School of Law.

### **James Markwith**

Jim Markwith is a senior attorney for Microsoft's intellectual property and licensing group in Redmond, WA. Mr. Markwith provides legal counsel to business units that use, acquire or license open source software (OSS). His duties include IP licensing transactions, pre and post-close IP due diligence for Microsoft's mergers and acquisitions, and counseling business groups on OSS project participation.

Prior to Microsoft, Mr. Markwith was corporate counsel, world-wide products and marketing at Adobe Systems. Prior to Adobe, Mr. Markwith was an adjunct professor at Santa Clara University where he taught protection of IP, legal aspects of managing technology, the economic analysis of IP law and macroeconomics. Prior to his legal career, Mr. Markwith was a US Navy pilot, and was appointed an instructor of economics at the United States Naval Academy, Annapolis, MD.

Mr. Markwith is on the board of directors of the Eastside Legal Assistance Program (ELAP).

He received his JD from The Santa Clara University School of Law, holds a MBA, and has a BS from The California State University at Long Beach.

**Jeffrey Ross Stern**

Jeffrey Ross Stern is an executive director at Morgan Stanley, in its technology, intellectual property, and e-commerce law group. His areas of practice include outsourcing, information technology, intellectual property, e-commerce, privacy and others.

Prior to joining Morgan Stanley, Mr. Stern was with Latham & Watkins and Weil, Gotshal & Manges.

Mr. Stern is chair of ACC's IT, Privacy and e-Commerce Committee. Mr. Stern participated in the drafting of GPL version 3 as co-chair of the Securities Industry and Financial Markets Association Working Group on Free and Open Source Software. He also is a member of SIFMA subcommittees addressing outsourcing, market data and electronic trading. Mr. Stern is a frequent presenter and is recognized by Who's Who in the World and Who's Who in American Law.

He is a graduate of Harvard College, magna cum laude, and Virginia Law School, where he was a member of the Virginia Law Review.

**Robert Tiller**

Robert Tiller is vice president and assistant general counsel, IP for Red Hat, Inc., the world's leading provider of open source technology to enterprise customers, where he manages patent, trademark, and copyright matters relating to open source software.

Before coming to Red Hat, he was a partner with the law firm of Helms, Mulliss & Wicker, PLLC, where he specialized in commercial and IP litigation. Mr. Tiller is a former clerk for Justice Antonin Scalia of the US Supreme Court and Judge Stephen Williams of the DC Circuit.

He is a graduate of the University of Virginia School of Law.

ACC Association of Corporate Counsel

### AM 808: Open Source - Latest Developments

*"Open source software is everywhere and whether you like it or not, your company is using it. It's one of the hottest areas of technology, both in terms of innovation and the ability to implement deep cost savings. This presentation will review the essentials in-house counsel need to know regarding their company's usage of open source software. The panel will also review the latest legal developments in this fast-moving area and what they mean in terms of how to represent your company."*

2009 Annual Meeting  
October 18-21 Boston

Don't just survive. Thrive!

---

---

---

---

---

---

---

---

ACC Association of Corporate Counsel

### Outline

- Definition of open source software (OSS)
- Review of license models and GPL
- OSS policies and practices of corporations
- Patent litigation involving open source
- OSS trends
- Open standards

2009 Annual Meeting  
October 18-21 Boston

Don't just survive. Thrive!

---

---

---

---

---


---

---

---

ACC Association of Corporate Counsel

### Defining open source software



Open source software is a broad category of third party software

2009 Annual Meeting  
October 18-21 Boston

Don't just survive. Thrive!

---

---

---


---

---

---

---

---



**Defining open source software**

Approaches to defining OSS

- The Four Freedoms
- Open Source Initiative definition
- Open source license lists
- Norms -- community and collaboration

2009 Annual Meeting  
October 18-21 Boston

*Don't just survive. Thrive!*

---

---

---


---

---

---

---

---



**The Four Freedoms**

1. Freedom to run software for any purpose
2. Freedom to modify the program to suit your own needs
3. Freedom to redistribute copies
4. Freedom to improve the software and distribute improvements

2009 Annual Meeting  
October 18-21 Boston

*Don't just survive. Thrive!*

---

---

---


---

---

---

---

---



**Open Source Initiative definition (1 of 2)**

Free Redistribution

Source Code

Derived Works

Integrity of The Author's Source Code

No Discrimination Against Persons or Groups

2009 Annual Meeting  
October 18-21 Boston

*Don't just survive. Thrive!*

---

---

---

---

---

---

---

---

ACC Association of Corporate Counsel

Open Source Initiative definition (2 of 2)

- 6. No Discrimination Against Fields of Endeavor
- 7. Distribution of License
- 8. License Must Not Be Specific to a Product
- 9. License Must Not Restrict Other Software
- 10. License Must Be Technology-Neutral

2009 Annual Meeting  
October 18-21 Boston

*Don't just survive. Thrive!*

---

---

---

---

---

---

---

---

ACC Association of Corporate Counsel

Pragmatic approach

Generally accepted open source licenses

- OSI list
  - Currently 65 licenses
  - <http://www.opensource.org/licenses/alph>
- Fedora Project list
  - Currently 180 licenses
  - <http://fedoraproject.org/wiki/Licensing>

2009 Annual Meeting  
October 18-21 Boston

*Don't just survive. Thrive!*

---

---

---

---

---

---

---

---

ACC Association of Corporate Counsel

Norms -- community and collaboration

Formal or informal OSS development "projects" formed by individuals, groups of independent developers and sometimes corporations

If successful, initiative has a large and active community of contributors, testers, and users

Collaboration takes place publicly over the Internet, using source code revision control systems, mailing lists, bug tracking systems and IRC channels

Code licensed under a OSS license or no license

2009 Annual Meeting  
October 18-21 Boston

*Don't just survive. Thrive!*

---

---

---

---

---

---

---

---

ACC Association of Corporate Counsel

### OSS licenses and “copyleft” principle

Play on words, but copy“right” still applies  
Licensing model supports a OSS commons  
Triggered by distribution:

- If you distribute a modified version of a copyleft-licensed program, you must use the copyleft license
- Output and internal use of copyleft program generally not subject to copyleft restriction

2009 Annual Meeting  
October 18-21 Boston

*Don't just survive. Thrive!*

---

---

---

---

---

---

---

---

ACC Association of Corporate Counsel

### Licenses types

Permissive

- ISC, BSD, MIT
- Simple terms

Strong copyleft

- GPL v. 2 – most widely used OSS license

Weak copyleft

- LGPL, EPL, MPL
- Generally weak copyleft license does not cover files under other licenses and permits inclusion in works under proprietary licenses

2009 Annual Meeting  
October 18-21 Boston

*Don't just survive. Thrive!*

---

---

---

---

---

---

---

---

ACC Association of Corporate Counsel

### OSS licensing continuum

BSD Style      Downstream      Copyleft

Freedom of use (developer perspective)

Permissive      Restrictive or Open

2009 Annual Meeting  
October 18-21 Boston

*Don't just survive. Thrive!*

---

---

---

---

---

---

---

---

ACC Association of Corporate Counsel

### Distributing products incorporating OSS (1 of 2)

Third-Party Component /License type	Application	Applicable License
BSD	BSD Software Product	Product EULA BSD
LGPLv2	LGPLv2 Software Product	LGPLv2 Product EULA Down-Stream

2009 Annual Meeting  
October 18-21 Boston

*Don't just survive. Thrive!*

---

---

---

---

---

---

---

---

ACC Association of Corporate Counsel

### Distributing products incorporating OSS (2 of 2)

2009 Annual Meeting  
October 18-21 Boston

*Don't just survive. Thrive!*

---

---

---

---

---

---

---

---

ACC Association of Corporate Counsel

### Most important OSS license: GPL

Stands for General Public License

Why is it important?

- Cornerstone of OSS Movement
- License under which Linux kernel distributed
- Used in majority (65%) of OSS

2009 Annual Meeting  
October 18-21 Boston

*Don't just survive. Thrive!*

---

---

---

---

---

---

---

---



**GPL Essentials**

Internal use not restricted

Distribution of binaries must be accompanied by complete corresponding source code or written offer to provide

Derivative works, if distributed, must be licensed under the GPL

- Derivative work versus common distribution

2009 Annual Meeting  
October 18-21 Boston

*Don't just survive. Thrive!*

---

---

---

---

---

---

---

---

**GPL v. 2 and GPL v. 3**

Similar, though v.3 more legalistic and international

More explicit treatment of certain issues in v.3

- Use by contractors is not a distribution
- Contributor grants express patent license for essential patent claims
- Affero GPL
- Tivoization – distributing an object code work in a consumer product
- Waiver of anti-circumvention rights

2009 Annual Meeting  
October 18-21 Boston

*Don't just survive. Thrive!*

---

---

---

---

---

---

---

---

**Business is a large player in OSS**

1. More than 70 percent of total contributions to the Linux kernel come from developers working at a range of companies including Red Hat, IBM, Novell, Intel, Oracle, Fujitsu, among others [www.linux-foundation.org/publications/linuxkerneldevelopment.php](http://www.linux-foundation.org/publications/linuxkerneldevelopment.php)
2. IBM/HP/Intel: spend roughly \$1 billion a year on Linux development
3. Effect on quality, IP issues?

2009 Annual Meeting  
October 18-21 Boston

*Don't just survive. Thrive!*

---

---

---

---

---

---

---

---

ACC Association of Corporate Counsel

### OSS policy for a corporation

- Why have one?
- Part of OSS strategy
- Component of broader IT/software policy
- Use of software by a corporation should be intended and controlled

2009 Annual Meeting  
October 18-21 Boston

*Don't just survive. Thrive!*

---

---

---

---

---

---

---

---

ACC Association of Corporate Counsel

### Defining an OSS strategy

- Internal use or external distribution required, encouraged, or banned?
- What requirements will be imposed upon vendors?
- Relation to internally developed and third party software generally

2009 Annual Meeting  
October 18-21 Boston

*Don't just survive. Thrive!*

---

---

---

---

---

---

---

---

ACC Association of Corporate Counsel

### OSS policy objectives and development

- How will strategy impact policy development?
- Approvals: downloading, use, modification & distribution
  - Release to customers
  - Contributions to community
  - What licenses are authorized under what circumstances?

(see also ACC survey of companies re Managing the use of Open Source Software  
<http://www.acc.com/legalresources/resource.cfm?show=16792>)

2009 Annual Meeting  
October 18-21 Boston

*Don't just survive. Thrive!*

---

---

---

---

---

---

---

---

ACC Association of Corporate Counsel

**Policy stakeholders**

- Front office business where applicable
- Back office IT (including IT security)
- Procurement/sourcing
- Legal and compliance
- Risk management

2009 Annual Meeting  
October 18-21 Boston

*Don't just survive. Thrive!*

---

---

---

---

---

---

---

---

ACC Association of Corporate Counsel

**Others parties relevant to OSS policy**

- Customers
- Open source developers and their advocates
- Other third party IP owners
- Regulators, especially in banking context

(See Federal Financial Institutions Examination Counsel, Risk Management of Free and Open Source Software, <http://www.compliancehome.com/resources/PCI/Articles/10164.html>)

2009 Annual Meeting  
October 18-21 Boston

*Don't just survive. Thrive!*

---

---

---

---

---

---

---

---

ACC Association of Corporate Counsel

**OSS-specific risks and benefits**

OSS-specific risks, e.g.:

- No IP indemnity or reps and warranties
- Forking

OSS-specific benefits, e.g.:

- Source code available with ability to assume control, modify, design around, and conduct more thorough due diligence

2009 Annual Meeting  
October 18-21 Boston

*Don't just survive. Thrive!*

---

---

---

---

---

---

---

---

ACC Association of Corporate Counsel

### OSS development model

- Code control varies from project to project
- Contribution agreements may require assignment or IP reps and warranties
- © chain of title
- In practice, customers of both proprietary software and OSS unlikely to be sued for IP infringement other than license breach

2009 Annual Meeting  
October 18-21 Boston

Don't just survive. Thrive!

---

---

---

---

---

---


---

---

ACC Association of Corporate Counsel

### OSS policy – risk of license breach

- As with third party software, may be enforcement action for exceeding license scope
- May be brought by Free Software Foundation or Software Freedom Law Center



2009 Annual Meeting  
October 18-21 Boston

Don't just survive. Thrive!

---

---

---

---

---

---

---

---

ACC Association of Corporate Counsel

### OSS policy v. other policies

- Procurement policies
- IT/software policies
- Security policies
- Computing resource policies
- External communication policies
- Human resource policies

2009 Annual Meeting  
October 18-21 Boston

Don't just survive. Thrive!

---

---

---

---

---

---

---

---

ACC Association of Corporate Counsel

### Educating employees

- What "free" means
- Warranties, contractual protection, support, updates
- Development restrictions
- Providing source on distribution
- Attribution requirements
- OSS intake: Legal, Security, Due diligence

2009 Annual Meeting  
October 18-21 Boston

*Don't just survive. Thrive!*

---

---

---

---

---

---

---

---

ACC Association of Corporate Counsel

### Open source intake questionnaire

- Brief description of application, licensor, license, due diligence
- What are the intended uses?
  - Modification
  - Distribution
  - Interface with existing applications?
  - Mission critical application?

2009 Annual Meeting  
October 18-21 Boston

*Don't just survive. Thrive!*

---

---

---

---

---

---

---

---

ACC Association of Corporate Counsel

### Internal use of OSS: developer tools

Additional questions for your clients:

1. Does output include OSS-licensed materials?
2. Is output a derivative of the licensed tool? (most likely no)

2009 Annual Meeting  
October 18-21 Boston

*Don't just survive. Thrive!*

---

---

---

---

---

---

---

---

**ACC** Association of Corporate Counsel

### OSS in commercial apps

- Vendor disclosure of OSS components
- OSS warranties?
  - Situational: \$, centrality, support?
  - Identification of components
  - Copies of licenses
  - Contractual out clauses

2009 Annual Meeting  
October 18-21 Boston

*Don't just survive. Thrive!*

---

---

---

---

---

---

---

---

**ACC** Association of Corporate Counsel

### Contributing open source - reasons

Front office business model

- Provision of related services
- Sale of related software

Back office business model

- Promote standardization/reduce cost

Corporate reputation/good citizen

- Establish expertise

2009 Annual Meeting  
October 18-21 Boston

*Don't just survive. Thrive!*

---

---

---

---

---

---

---

---

**ACC** Association of Corporate Counsel

### Contributing OSS - liability

Role of disclaimers

IP liability

System error

Security vulnerability

Reputational harm

2009 Annual Meeting  
October 18-21 Boston

*Don't just survive. Thrive!*

---

---

---


---

---

---

---

---

 ACC Association of Corporate Counsel

**Methods for contributing OSS** (1 of 3)

Developer web site or consortium –

- Contribution agreement?
- Indemnity?
- Copyright assignment?

Implied license?

2009 Annual Meeting  
October 18-21 Boston

*Don't just survive. Thrive!*

---

---

---


---

---

---

---

---

 ACC Association of Corporate Counsel

**Methods for contributing OSS** (2 of 3)

Contribute through third party vendor  
Targeted distribution/contribution  
Inadvertent distribution (oops)  
Low versus high profile

2009 Annual Meeting  
October 18-21 Boston

*Don't just survive. Thrive!*

---

---

---


---

---

---

---

---

 ACC Association of Corporate Counsel

**Methods for contributing OSS** (3 of 3)

“Fire and forget” – Microsoft Codeplex  
Contributing to an established OSS project

- A. Microsoft’s Moodle Live Services Plug-in
- B. Apache projects

Establishing an OSS Project

- A. IBM – Eclipse Foundation
- B. Yahoo – Free BSD Project/Foundation

2009 Annual Meeting  
October 18-21 Boston

*Don't just survive. Thrive!*

---

---

---

---

---

---

---

---

ACC Association of Corporate Counsel

### Patent litigation and open source

Few patent lawsuits involving OSS

- FireStar v. Red Hat
- IP Innovations v. Red Hat and Novell
- Trend Micro v. Barracuda
- Network Appliance v. Sun
- Microsoft v. TomTom
- Software Tree v. Red Hat

2009 Annual Meeting  
October 18-21 Boston

*Don't just survive. Thrive!*

---

---

---

---

---

---

---

---

ACC Association of Corporate Counsel

### Firestar patent lawsuit settlement - history

- Firestar Software, Inc. alleged infringement in 2006
  - A mysterious entity – a troll?
- Accused product – JBoss Hibernate, middleware
  - Allows applications and operating systems to communicate with each other
  - Web 2.0
- U.S. Patent No. 6,101,502 – Object Model Mapping and Runtime Engine for Employing Relational Database with Object Oriented Software

2009 Annual Meeting  
October 18-21 Boston

*Don't just survive. Thrive!*

---

---

---

---

---

---

---

---

ACC Association of Corporate Counsel

### Patent licensing concerns

LGPL and GPL address patent licenses

- Both prevent patent licenses that contain "conditions . . . that contradict the conditions of this License"
- Both bar patent licenses that "would not permit royalty-free redistribution"
- GPL preamble: "any patent must be licensed for everyone's free use or not licensed at all"
- LGPL preamble: "any patent license . . . must be consistent with the full freedom of use specified in this license."

No precedents on meaning of this language, or on how to structure an open source settlement

2009 Annual Meeting  
October 18-21 Boston

*Don't just survive. Thrive!*

---

---

---

---


---

---

---

---





**Settlement elements**

Dismissal of lawsuit  
License and covenant not to use  
Grantor

- DataTern, successor in interest to Firestar
- Amphion, affiliate of DataTern

2009 Annual Meeting  
October 18-21 Boston

*Don't just survive. Thrive!*

---

---

---


---

---

---

---

---



**License**

Perpetual, fully paid-up, royalty-free, worldwide license to engage in any and all activities related to the covered products, including using, selling, distributing, and importing (5.1)

2009 Annual Meeting  
October 18-21 Boston

*Don't just survive. Thrive!*

---

---

---


---

---

---

---

---



**License relates to**

- All patents on which DataTern or its affiliates relied on in litigation, other patents owned by DataTern prior to the effective date, and any related applications, continuations, or foreign counterparts (1.22)
- Community members are all persons that are licensors and licensees of, or who contribute to, sell, distribute, use or import, a Covered Product (1.18)
- Covered Product: any product sold or distributed by Red Hat under its brand, a predecessor version of its products (including an upstream predecessor), and a derivative or combination product of these (1.20)

2009 Annual Meeting  
October 18-21 Boston

*Don't just survive. Thrive!*

---

---

---

---

---

---

---

---

**ACC** Association of Corporate Counsel

### License coverage

Licenses include both Red Hat and Community Members

- CM is any entity that contributes to a Red Hat licensed product or is a licensee as to such product
- Thus, all upstream developers and downstream customers

Coverage of license is broad, but note one limitation:

- License does not apply where there is infringement without use or reference to any portion or functionality of a Red Hat Product
- Thus combining an infringing product with a Red Hat Product does not immunize the infringing product

2009 Annual Meeting  
October 18-21 Boston

*Don't just survive. Thrive!*

---

---

---

---

---

---

---

---

---

---

**ACC** Association of Corporate Counsel

### Release and covenant not to sue

- Release by DataTern and Amphion of all claims against any community member related to covered patents (6.1,6.2)
- Current and near future intellectual property – Patents currently owned or acquired within five years by either DataTern or Amphion (1.3)
- Future intellectual property – Patents acquired more than five years after the agreement – No suits for five years and no damages for five-year period following the agreement (1.9)

2009 Annual Meeting  
October 18-21 Boston

*Don't just survive. Thrive!*

---

---

---

---

---

---

---

---

---

---

**ACC** Association of Corporate Counsel

### OSS Trends (1 of 2)

1. Written policies – controlled internal use
2. Automated code management/tracking
3. Shift to the Cloud - SaaS models increasing
4. Carefully scoped OSS project participation

2009 Annual Meeting  
October 18-21 Boston

*Don't just survive. Thrive!*

---

---

---

---

---

---

---

---

---

---

ACC Association of Corporate Counsel

OSS Trends (2 of 2)

- 5. New business models
- 6. Debate over license models continues
- 7. Technical due diligence (TDD) innovation
- 8. Open standards

2009 Annual Meeting  
October 18-21 Boston

Don't just survive. Thrive!

---

---

---

---

---

---

---

---

ACC Association of Corporate Counsel

Open standards

- Not Open Source
- Metcalfe's Law/network effects
- OSS and commercial software
- Standard setting organizations
- De facto standards

2009 Annual Meeting  
October 18-21 Boston

Don't just survive. Thrive!

---

---

---

---

---

---

---

---

ACC Association of Corporate Counsel

APPENDIX

2009 Annual Meeting  
October 18-21 Boston

Don't just survive. Thrive!

---

---

---

---

---

---

---

---

**ACC** Association of Corporate Counsel

### Open Source Initiative definition of OSS

- Free Redistribution**  
The license shall not restrict any party from selling or giving away the software as a component of an aggregate software distribution containing programs from several different sources. The license shall not require a royalty or other fee for such sale.
- Source Code**  
The program must include source code, and must allow distribution in source code as well as compiled form. Where some form of a product is not distributed with source code, there must be a well-publicized means of obtaining the source code for no more than a reasonable reproduction cost preferably, downloading via the Internet without charge. The source code must be the preferred form in which a programmer would modify the program. Deliberately obfuscated source code is not allowed. Intermediate forms such as the output of a preprocessor or translator are not allowed.

2009 Annual Meeting  
October 18-21 Boston

*Don't just survive. Thrive!*

---

---

---

---

---

---

---

---

---

---

**ACC** Association of Corporate Counsel

### OSI definition of open source (cont'd)

- Derived Works**  
The license must allow modifications and derived works, and must allow them to be distributed under the same terms as the license of the original software.
- Integrity of The Author's Source Code**  
The license may restrict source-code from being distributed in modified form only if the license allows the distribution of "patch files" with the source code for the purpose of modifying the program at build time. The license must explicitly permit distribution of software built from modified source code. The license may require derived works to carry a different name or version number from the original software.
- No Discrimination Against Persons or Groups**  
The license must not discriminate against any person or group of persons.

2009 Annual Meeting  
October 18-21 Boston

*Don't just survive. Thrive!*

---

---

---

---

---

---

---

---

---

---

**ACC** Association of Corporate Counsel

### OSI definition of open source (cont'd)

- No Discrimination Against Fields of Endeavor**  
The license must not restrict anyone from making use of the program in a specific field of endeavor. For example, it may not restrict the program from being used in a business, or from being used for genetic research.
- Distribution of License**  
The rights attached to the program must apply to all to whom the program is redistributed without the need for execution of an additional license by those parties.

2009 Annual Meeting  
October 18-21 Boston

*Don't just survive. Thrive!*

---

---

---

---

---


---

---

---

---

---



### OSI definition of open source (cont'd)

8. License Must Not Be Specific to a Product  
 The rights attached to the program must not depend on the program's being part of a particular software distribution. If the program is extracted from that distribution and used or distributed within the terms of the program's license, all parties to whom the program is redistributed should have the same rights as those that are granted in conjunction with the original software distribution.

9. License Must Not Restrict Other Software  
 The license must not place restrictions on other software that is distributed along with the licensed software. For example, the license must not insist that all other programs distributed on the same medium must be open-source software.

10. License Must Be Technology-Neutral  
 No provision of the license may be predicated on any individual technology or style of interface.

2009 Annual Meeting  
 October 18-21 Boston

*Don't just survive. Thrive!*

---

---

---

---

---


---

---

---

---

---



### OSI definition of open standards

**No Intentional Secrets:** The standard MUST NOT withhold any detail necessary for interoperable implementation.

**Availability:** The standard MUST be freely and publicly available (e.g., from a stable web site) under royalty-free terms "at reasonable and non-discriminatory cost"

**Patents:** All patents essential to implementation of the standard MUST:  
 - be licensed under royalty-free terms for unrestricted use, or  
 - be covered by a promise of non-assertion when practiced by open source software

**No Agreements:** There MUST NOT be any requirement for execution of a license agreement, NDA, grant, click-through, or any other form of paperwork to deploy conforming implementations of the standard.

**No OSR-Incompatible Dependencies:** Implementation of the standard MUST NOT require any other technology that fails to meet the criteria of this Requirement.  
<http://www.opensource.org/osr>

2009 Annual Meeting  
 October 18-21 Boston

*Don't just survive. Thrive!*

---

---

---

---

---

---

---

---

---

---

# Open Source Software: IP Due Diligence, Litigation, and Industry Trends

---

**Jim Markwith**

**OPEN SOURCE SOFTWARE** (OSS or Open Source) is simply software licensed under an “Open Source” license. Although there is no single agreed-upon definition of what constitutes an Open Source license, the Open Source Initiative (OSI) lists over 65 licenses that, according to the OSI, meet the OSI’s Open Source Definition.

<http://www.opensource.org/docs/definition.php>. When thinking about types of Open Source licenses, it is helpful to divide them into two categories, Reciprocal and BSD-style licenses.

## **Reciprocal Licenses**

“Reciprocal” licenses (sometimes referred to as “Copyleft” or “Viral” licenses) often include significant restrictions and limitations. Reciprocal licenses typically include a copyright license (and possibly a patent license) to use or modify the materials, but in return (hence the term Reciprocal) requires products derived from (or combined with) those materials to be licensed only under the terms of that same Reciprocal license. Such a restriction could preclude the use of other license terms by the user, such as those found in standard end user license agreements (EULAs). Reciprocal licenses also typically require the licensor to distribute or make available the source code to the licensed materials, and any improvements made thereto. Reciprocal licenses can require the user to grant broad rights to their related IP, and may preclude the user from enforcing their related patents, even in a defensive manner. The most commonly used Reciprocal license is Version 2 of The GNU General Public License (GPLv2).

## **BSD-style Licenses**

“BSD (Berkeley Software Development)-style” (also called non-Reciprocal or Attribution) licenses are generally very flexible and present few restrictions or obligations. Such licenses typically grant the user broad rights to use and modify the software, provided that the user includes the copyright attribution specified by the author. Products that use code licensed under a BSD-style license can usually be modified and redistributed under any license, such as

Based on a paper the author wrote for the June 2009 Edition of the ABA Practical Lawyer Magazine.

a traditional EULA, or under another Open Source license.

### **Microsoft Open Source Licenses**

Microsoft recently had two of its own licenses approved by the OSI, the Microsoft Public License (Ms-PL) (<http://www.opensource.org/licenses/ms-pl.html>), and the Microsoft Reciprocal License (Ms-RL) (<http://www.opensource.org/licenses/ms-rl.html>). The Ms-PL would be categorized as a BSD-style license, while (as the name implies) the Ms-RL is reciprocal and contains additional conditions. See Microsoft Reciprocal License (Ms-RL), Paragraph 3 (Conditions and Limitations). However, the Ms-RL is relatively simple when compared to many popular Reciprocal licenses such as the GPL licenses. For example, the reciprocal obligations under the Ms-RL apply only to the specific file that contains the Ms-RL licensed code, whereas the GPLv2 (and GPLv3) can apply to the entire application or program that incorporates the GPLv2 licensed code, causing the entire program to be governed by the terms and conditions of the GPLv2.

### **IP Rights and Software**

Software can be protected by patents, copyright, trademarks, and trade secrets. Copyright protection applies automatically upon the creation of a copyrightable work (when it is affixed in some tangible medium) and software has been deemed copyrightable by Congress and the courts. (See U.S. Code: Title 17, Chapter 1, §101: A work is “fixed” in a tangible medium of expression when its embodiment in a copy or phonorecord, by or under the authority of the author, is sufficiently permanent or stable to permit it to be perceived, reproduced, or otherwise communicated for a period of more than transitory duration.) Once a copyrightable work is created, the work is automatically protected by copyright law, and the copyright holder has certain exclusive rights to the work (specifically, to reproduce, create derivatives, distribute, and publicly perform or display the work). U.S. Code: Title 17, Chapter 1, §106. Because the author has the exclusive rights (subject to some limitations), anyone else who wishes to use the work needs the permission of the author. (See U.S. Code 17, Chapter 1, §107. Limitations on exclusive rights: Fair use). Without such permission the user could be deemed a copyright infringer. Traditionally, such permissions are in the form of licenses by which the IP holder grants the licensee particular rights while retaining others.

Because different OSS licenses may have very different terms, it is critical to recognize what license applies to the OSS you want to use. Each type of license confers different rights and imposes different obligations, and software applications can be governed by many different OSS licenses. So even though two software applications may be licensed as OSS, if they are licensed under different licenses, the terms and conditions governing the use of one may be significantly different than the terms and conditions governing the other. The rights you have, and thus the limitations on what you can do with that software, are completely dependent upon

Based on a paper the author wrote for the June 2009 Edition of the ABA Practical Lawyer Magazine.

the terms and conditions of the licenses that control the code in question.

When Microsoft acquires a company with OSS in its code base, we perform what we call technical due diligence (TDD), which includes a process to identify third-party code and the licenses that apply to such code, along with an analysis of the code's quality and security. Once we have identified the third party code and the licenses that control that code, we determine if the code is being used in a manner that is consistent with the terms and conditions of the applicable licenses, a process generally referred to as IP due diligence, a subset of TDD. Through our experience with TDD, we have gained significant insight into the general trends relating to software development today, and understand many of the specific issues, risks and misunderstandings related to IP and OSS. In 2008 Microsoft acquired over 500 million lines of code. In any particular deal the code contained between 10 percent and 90 percent third-party code, which included both OSS and non-OSS third-party code. On average, about one third of the third-party code was Reciprocal code.

### IP Chain of Title Issues

From an acquisition perspective, OSS is merely third-party code, and the need to conduct IP due diligence on OSS is the same as with other third-party code. The challenge is to find out what is in the code, determine if it is properly licensed, and whether or not that license allows you to use it in the manner you desire. An additional challenge with OSS comes from the collaborative nature of OSS development. (See Eric Raymond, *The Cathedral and the Bazaar*. Open source evangelist Eric Raymond suggests a model for developing OSS known as the Bazaar model. In such a model users are treated like co-developers and so they are to have access to the source code and are encouraged to submit contributions to the software, including bug fixes and reports for the software and documentation.) Most OSS projects are Web-based, and many are accessible by anyone who wants to contribute to them. If a project is unregulated, it may be difficult if not impossible to determine the origin of the code that resides in such a project. The amount of control over contributions to any particular OSS project varies from project to project. Some projects have tight control over what can be introduced into that project's code base, while other projects have little or no oversight. The important thing to understand is that OSS code may include unidentified third-party content that is not properly licensed. Standard TDD during an acquisition includes a disclosure report from the company being acquired listing known third-party materials (including OSS). On average, the amount of OSS found upon forensic examination of acquired code by Palamida, Inc. (a company that specializes in forensic examination of software) is three to five times the amount disclosed. See, Mark E. Tolliver, CEO, Palamida, Inc. <http://www.palamida.com/about>. If you acquire a company that uses OSS and do not know the origin (and composition) of that code, or whether it is being used in compliance with the license(s) that governs its use, you run the risk of buying an infringement claim.

On a related note, we are seeing an increase in the inability of acquired companies to produce the licenses associated with the third-party code that is identified during TDD, a

Based on a paper the author wrote for the June 2009 Edition of the ABA Practical Lawyer Magazine.



phenomenon that is related to the lack of knowledge about the existence of third-party code (including OSS) in their code base, and the importance of such record-keeping. We find that many developers either do not have procedures that require them to keep track of licenses associated with third-party code, or they do not follow the procedures that they do have. It is also common for developers at some companies to pull code off the Internet with little or no management oversight as part of their standard development practices. When developers create software, they often do not create code from scratch, but rather combine different existing and independently licensed applications together. Such development is often done without a thorough understanding of the rights and obligations related to the use of the code. In such a case, it is not uncommon for management to be surprised by the actual content of their code.

In many cases, the above issues aren't necessarily deal killers, but in some cases they are. Most Microsoft acquisitions involve OSS code. Whether or not the OSS is problematic depends upon how we intend to use it and whether the OSS can be used in that manner. During the TDD process we assess the compatibility of the code with our own business models and product requirements. For example, if we want to include particular code in a shipping commercial product, but the license associated with that code prohibits commercial use, we would either need to remove it, or replace it with similar code that is licensed in a manner that is compatible with commercial use.

### **Confusion and Complexities Surrounding OSS Licenses**

Although there are currently over 65 licenses listed by the OSI as Open Source licenses, during our due diligence efforts we have reviewed about 1,000 different variations of those licenses. Attorneys familiar with the GPL licenses know that they are complex licenses that are often difficult to interpret. If you consider the complexity of licenses like the GPL licenses, the sheer number of different licenses that can apply to a single code base, and the lack of knowledge about the content of many code bases (and thus the licenses that control it), it becomes clear why we are seeing a great amount of OSS license non-compliance.

There is also much confusion about the difference between "open standards" and OSS. An open standard is a publicly available "specification," whereas OSS is software that is subject to an "open source" license, and which may be used to implement an open standard in a particular product or service. Whether a standard qualifies as "open" has nothing to do with the type of software used to implement that standard or the license controlling that software. In fact, open standards are agnostic with regard to software licensing or business models, welcoming all models and favoring none, and so it is equally possible for an open standard to be implemented in proprietary software as it is in OSS. For example, HTML is an open standard. Firefox is a Mozilla OSS Web browser that implements the HTML standard. Similarly, Internet Explorer, a Web browser supplied by Microsoft which is not OSS, also implements the HTML standard.

Based on a paper the author wrote for the June 2009 Edition of the ABA Practical Lawyer Magazine.

## OSS Related Litigation and Enforcement Actions are Commonplace

The litigation between Trend Micro, Inc. and Barracuda Networks, Inc. is proof that OSS and other “Free” software are not free of litigation risks. In fact, litigation is a commonplace occurrence regardless of the software development model used or the license under which the software is obtained. See Francis M. Buono & McLean Sieverding, *Trend spotting: Recognizing the Growing Risk of IP Litigation Facing OSS Developers and Implementers*. The Metropolitan Corporate Counsel, September 2008, available at <http://www.metrocorpcounsel.com/current.php?artType=view&artMonth=September&artYear=2008&EntryNo=8702>. IP disputes related to OSS are also not limited to disputes between commercial enterprises. The Free Software Foundation (FSF) and the Software Freedom Law Center (SFLC) have aggressively pursued and even sued certain companies, even very small OSS developers, for failure to comply with the terms and conditions of the GPL license. See, e.g., *Anderson et al. v. Monsoon Multimedia, Inc.*, No. 1:07cv8205 (S.D.N.Y. 2007) (settled); *Andersen et al. v. Verizon*, No. 1:07-cv-11070 (S.D.N.Y. 2007) (settled); *Andersen et al. v. Xterasys Corp.*, No. 07-CV-10455 (S.D.N.Y. 2007) (settled); *Andersen et al. v. High-Gain Antennas*, No. 07-CV-10456 (S.D.N.Y. 2007) (settled); *Anderson v. Extreme Networks, Inc.*, No. 08-CV-6426 (S.D.N.Y. 2008); *Anderson v. Super Micro Computer, Inc.*, No. 1:08-cv-05269-RMB (S.D.N.Y. 2008); *Anderson v. Bell Microproducts, Inc.*, No. 08-CV-5270 (S.D.N.Y. 2008); *MDY Industries, LLC v. Blizzard Entertainment et al.*, No. CV-06-2555-PHX (D. Ariz. 2008).

The OSS enforcement efforts initiated by the FSF and the SFLC that have concluded have ended in settlement, and the terms of those settlements have reportedly imposed significant burdens on the OSS developers involved. See [http://www.news.com/8301-13580\\_3-9808378-39.html](http://www.news.com/8301-13580_3-9808378-39.html). In their recent GPL Compliance Guide, the SFLC set forth its “standard” settlement demands: 1) compliance with all OSS copyrights, not just the program at issue; 2) notification to past recipients of the program; 3) appointment of a GPL compliance officer, and; 4) periodic compliance reports. See: Kuhn, Williamson & Sandler, *A Practical Guide to GPL Compliance*, (Software Freedom Law Center 2008) available at <http://www.softwarefreedom.org/resources/2008/compliance-guide.pdf>. Reportedly, the prior FSF and SFLC settlements have also included a significant monetary component. It is interesting to note however, that the Guide to GPL Compliance only addresses the clear cases of GPL violation. It expressly avoids addressing the more complicated and problematic ambiguities in the GPLv2, such as when using a GPLv2 licensed program with a proprietary program will cause the combination of the two to be governed by the GPLv2. (See Section 3.1 of the Guide: “Most companies accused of violations, however, lack a basic understanding of how to comply even in the straightforward scenario. This document provides that fundamental and generally applicable prerequisite knowledge. For answers to rarer and more complicated legal questions, such as whether your software is a derivative work of some copylefted software, consult with an attorney.” See also The FSF’s FAQ to the GPLv2 at <http://www.gnu.org/licenses/old-licenses/gpl-2.0-faq.html#MereAggregation>. “What constitutes

Based on a paper the author wrote for the June 2009 Edition of the ABA Practical Lawyer Magazine.

combining two parts into one program? This is a legal question, which ultimately judges will decide. We believe that a proper criterion depends both on the mechanism of communication (exec, pipes, rpc, function calls within a shared address space, etc.) and the semantics of the communication (what kinds of information are interchanged).”). In light of such ambiguities and the aggressive stance taken by the FSF and the SFLC (and the extent of their settlement demands), commercial companies should be careful when dealing with GPL-licensed software.

**THE COEXISTENCE OF OSS AND PROPRIETARY SOFTWARE** • The software industry is often depicted as irreconcilably divided into mutually exclusive, rival camps of proprietary software and OSS. (“Proprietary software” is software that is subject to licenses that typically restrict the licensee’s right to copy, redistribute, or modify the software, and normally do not grant access to the software’s source code. These restrictions help to protect the developer’s investment in the software by preventing third parties from expropriating the software’s economic value without the developer’s authorization.) Market forces, however, are rendering this portrayal obsolete. Both models have proven beneficial to the software market, which has determined that they should coexist in healthy competition, and has even driven them to embrace each other's principles in certain respects. For example, estimates are that leading companies such as IBM, Hewlett-Packard, Intel, Fujitsu, Red Hat, and Novell currently spend at least \$1 billion a year on the development of Linux. The growth in OSS-related revenues has been tremendous.

The OSS community has likewise moved toward traditional commercial business models. Although it may sound appealing to have software without vendor control, the reality is that customers, whether corporate or consumer, want high value, low risk, investment protection, along with interoperability and predictability. These demands result in vendors offering “open source” solutions that rely on many of the same principles and business controls as the traditional proprietary software model. For example, enterprise customers demand service-level support agreements from their software vendors. To deliver stated service levels, the vendor must make sure that the source code in the supported environment is maintained in a stable state. Therefore their support contracts often specify that the customer may not modify source code without invalidating their support contract.

As defined earlier in this paper, OSS is simply software licensed under an Open Source license. Therefore, Microsoft does not compete with OSS; rather Microsoft competes with providers of software and services that are similar to Microsoft’s offerings, such as RedHat, IBM, and Sun Microsystems, all of which (like Microsoft) use OSS in some capacity. Ultimately, the demands of software customers will bring about the most innovative and cost-effective software products and services, which will be made from both OSS and proprietary software. The software ecosystem as a whole will continue to benefit from the coexistence of OSS and proprietary software, and in the end, software makers will use either OSS or proprietary software components based upon total cost, quality, efficiency, and the ability to quantify and control IP risks.

Based on a paper the author wrote for the June 2009 Edition of the ABA Practical Lawyer Magazine.

Competition in the software industry will likely result in a trend toward more flexible OSS licenses. Many current business models are designed to work around the restrictions of the GPL licenses, which for example preclude per-unit licensing fees and may prohibit DRM technology. But consumer demand for products and business models that take advantage of emerging technologies will require more flexible licenses than the GPL licenses, and the industry will likely trend away from the GPL and toward more commercially friendly BSD-style licenses.

**CONCLUSION** • We are entering an era in which software development, licensing, and acquisitions are more mature, more diligent, and more focused on the business objectives of vendors, acquirers, and customers. OSS and proprietary software are not mutually exclusive, but can and do co-exist in ways that complement each other. The market demand for converging and heterogeneous IT technologies will continue to increase, and should cause a trend away from restrictive Reciprocal licenses and toward more flexible BSD-style licenses. IP is critical to both OSS and proprietary software, and your ability to identify the origin of the IP associated with specific OSS may be difficult or impossible. Because IP applies to software of all types, and IP litigation is prevalent regardless of the licensing model involved, it is important to determine the licenses that apply to code you want to use, and to understand how to comply with such licenses. Software developers should strive to develop practices and policies that can unlock the benefits of OSS to developers, while respecting third-party rights, and in a manner that protects their own strategic IP.

Based on a paper the author wrote for the June 2009 Edition of the ABA Practical Lawyer Magazine.

# Open Source Software: Definitions, Due Diligence and Distribution

By James W. Markwith, Esq.

October 1, 2009

---

**WHAT IS OPEN SOURCE SOFTWARE?** Open Source Software (OSS or Open Source) is simply software licensed under an “Open Source” license. Although there is no single agreed-upon definition of what constitutes an Open Source license, the Open Source Initiative (OSI) lists over 65 licenses that, according to the OSI, meet the OSI’s Open Source Definition.<sup>1</sup> When thinking about types of Open Source licenses, it is helpful to divide them into three major categories, BSD-style, Reciprocal, and Downstream.

## BSD-style OSS Licenses

“BSD-style” licenses are generally very flexible and present few restrictions or obligations.<sup>2</sup> Such licenses typically grant the user broad rights to use and modify the software, so long as the user includes the copyright attribution specified by the author. Software that is licensed under a BSD-style license can be modified and redistributed under any license, such as a traditional end user license agreement (EULA), or under another Open Source license, offering maximum flexibility from a developer’s perspective.

## Reciprocal OSS Licenses

“Reciprocal” licenses, sometimes referred to as “Copyleft” or “Viral” licenses, often include significant restrictions and limitations. They typically include a broad copyright license and an implied or express patent license to use or modify the materials, and in return requires products derived from or combined with those materials to be licensed only under the terms of that same Reciprocal license. Such restrictions preclude the use of additional or different license terms by the user, such as those found in a EULA. Reciprocal licenses also may require

---

<sup>1</sup> <http://www.opensource.org/docs/definition.php>.

<sup>2</sup> BSD stands for Berkeley Software Development. This category is also referred to as non-Reciprocal or Attribution-style. Examples include the BSD, MIT and MS-PL licenses.

the licensor to distribute or make available the source code to the licensed materials, and any improvements or contributions made thereto. Reciprocal licenses can require the user to grant broad rights to their related IP that applies to the distributed Reciprocally licensed code, and may preclude the user from enforcing their related patents, even in a defensive manner. The most commonly used Reciprocal license is Version 2 of The GNU General Public License (GPLv2).

### **Downstream Licenses.**

Downstream licenses are not as flexible as BSD-style licenses, but they may offer the developer more flexibility than Reciprocal licenses do. Downstream licenses usually either require the license under which you will distribute the Downstream licensed code to be “compatible” with the Downstream license, or requires the Downstream-licensed code to be distributed under the terms of that Downstream license, on a file-by-file basis. In other words, you can combine Downstream code with your proprietary product code, but the Downstream code itself must be licensed under its original license, and the license you use for the remainder of the code must not be incompatible with the terms of the Downstream license. In such a case, you may be able to use your EULA, but might also have to distribute the Downstream code under its original license. The risks associated with Downstream licensed code are generally associated with the complexity of the licenses, and the ease by which the terms and conditions can be inadvertently violated. One example of a downstream license is the Lesser GNU General Public License (LGPLv2). Under the LGPLv2, it may be possible to use the code in a manner that causes that license to function as a Downstream license, but if you use the code incorrectly, the license becomes, from a licensing standpoint, the equivalent of its Reciprocal big brother, the GPLv2.

### **OSS Does Not Compete with Proprietary Code**

OSS is not a product or industry in itself. As defined earlier, OSS is simply software licensed under an Open Source license. Therefore, Microsoft for example, does not compete with OSS; rather Microsoft competes with providers of software and services that are similar to Microsoft's offerings, such as RedHat, IBM, and Oracle, all of which, like Microsoft, use OSS in some capacity. Ultimately, the demands of software customers will bring about the most

innovative and cost-effective software products and services, which will be made from both OSS and proprietary components. The software ecosystem as a whole will continue to benefit from the coexistence of OSS and proprietary software, and in the end, software makers will use either OSS or proprietary software components based upon total cost and quality. Calculating total costs associated with developing software should include any costs associated with vetting OSS materials<sup>3</sup>, and the risks associated with potential IP infringement should not be ignored. Using any third-party code involves the risk that such code contains unauthorized infringing content, but unlike many commercial licenses, OSS licenses do not include reps and warranties relating to the non-infringement of third-party IP.<sup>4</sup>

### **OSS is Big Business**

As explained above, OSS does not compete against proprietary software, but traditional software makers do use OSS strategically to compete against each other. While some still depict the software industry as irreconcilably divided into mutually exclusive, rival camps of proprietary software and OSS<sup>5</sup>, the reality is that some of the largest proprietary software manufacturers are responsible for the development of the most important OSS projects. For example, estimates are that leading companies such as IBM, Hewlett-Packard, Intel, Fujitsu, Red Hat, and Novell spend at least \$1 billion a year on the development of Linux<sup>6</sup>, and the growth in OSS-related revenues has and will continue to be tremendous.<sup>7</sup>

---

<sup>3</sup> Costs could include forensic code scanning services such as those available from Black Duck or Palamida, Inc. Costs associated with such services vary depending on the size and complexity of the code base. In my experience the costs have ranged from \$10,000 to as much as \$70,000 depending on the size of the job. Factors involved in the decision to perform a forensic analysis could include, among many other things, the importance of the product involved and the ease by which it can be withdrawn from the market in the event an IP infringement claim is made. OSS materials used strictly for internal purposes and which are not distributed are in a much lower risk category than materials that are to be distributed.

<sup>4</sup> Due to the restrictions in the terms of the most common Reciprocal licenses, competition in the software industry will likely result in a trend toward more flexible OSS licenses, such as BSD-style licenses. While many current business models are designed to work within the restrictions of the GPL licenses, such as its prohibition on per-unit licensing fees, consumer demand for products and business models that take maximum advantage of emerging technologies will require more flexible license terms like those found in the BSD License.

<sup>5</sup> "Proprietary software" is software that is subject to licenses that typically restrict the licensee's right to copy, redistribute, or modify the software, and normally do not grant access to the software's source code. These restrictions help to protect the developer's investment in the software by preventing third parties from expropriating the software's economic value without the developer's authorization.

<sup>6</sup> Linux is a generic term referring to Unix-like computer operating systems based on the Linux Kernel. See: <http://en.wikipedia.org/wiki/Linux>

<sup>7</sup> An IDC study is forecasting that the standalone OSS market is poised to grow from \$1.8 billion in 2006 to \$5.8 billion in 2011, a compound growth rate of 26% per year.

The OSS community has also adopted elements of traditional software business models. For example, enterprise customers demand service-level support agreements from their software vendors, but in order to deliver promised service levels, the vendor must make sure that the source code is maintained in a stable state. Therefore, support contracts covering OSS applications often specify that the customer may not modify the source code without invalidating their support contract.

**INTELLECTUAL PROPERTY AND OSS** • Software can be protected by patents, copyright, trademarks, and trade secrets. Copyright protection applies automatically upon the creation of a copyrightable work (when it is affixed in some tangible medium) and software has been deemed copyrightable by Congress and the courts.<sup>8</sup> Once a copyrightable work is created, the work is automatically protected by copyright law, and the copyright holder has certain exclusive rights to the work, specifically, to reproduce, create derivatives from, distribute, and publicly perform or display the work.<sup>9</sup> Because the author has the exclusive rights, subject to some limitations, anyone else who wishes to use the work needs the permission of the author.<sup>10</sup> Without such permission the user could be deemed a copyright infringer. Traditionally, such permissions are in the form of licenses, by which the rights holder grants the licensee particular rights, while retaining others.

Because different OSS licenses may have very different terms, it is critical to recognize what license applies to the OSS you want to use. Each type of license confers different rights and may impose different obligations, and any particular software application may be governed by multiple OSS license(s). So even though two software applications may be referred to simply as OSS, if they are licensed under different licenses, the terms and conditions governing the use of one may be significantly different than those governing the other. The rights you have, and thus the limitations on what you can do with that software, are completely dependent upon the terms and conditions of the licenses that control the code in question.

---

<sup>8</sup> See U.S. Code: Title 17, Chapter 1, §101: A work is “fixed” in a tangible medium of expression when its embodiment in a copy or phonorecord, by or under the authority of the author, is sufficiently permanent or stable to permit it to be perceived, reproduced, or otherwise communicated for a period of more than transitory duration.

<sup>9</sup> U.S. Code: Title 17, Chapter 1, §106.

<sup>10</sup> See U.S. Code 17, Chapter 1, §107. Limitations on exclusive rights: Fair use.



## IP Due Diligence

Most commercial software developers have a process to review and approve third-party code that is used during the development process. While practices vary widely, at a minimum the review process should include a way to identify all third-party materials and their applicable licenses. Technical due diligence (TDD) is the process by which the third-party code and their applicable licenses are identified, and it may also include the analysis of the code's quality and potential security vulnerabilities. Once identified, all third-party licenses must be reviewed to determine if the code is being used in a manner that is consistent with their terms and conditions.

I have observed some general trends relating to modern software development, as well as some common issues, risks and misunderstandings related to the use of OSS.<sup>11</sup> For example, during an acquisition it is common to require a disclosure report from the company being acquired, which is supposed to list all third-party materials, including any OSS. Most of the time, acquisition partners vastly underreport the amount of OSS included in their code, and they are frequently unable to produce the licenses associated with the code they do identify. The phenomenon is not typically related to an intentional underreporting, but rather is due to a genuine lack of knowledge about the existence of third-party code, including OSS, in their code base, and a lack of processes that require third-party licenses to be properly archived. Many developers either do not have procedures that require them to keep track of licenses, or they just do not follow the procedures that they do have.

It is important to understand that when developers create software, they often do not create the code from scratch, but rather combine existing and independently licensed applications or components together. Most of the time those applications are found on the Internet, and they are downloaded and used with little or no management oversight. Such development is often done without a thorough understanding of the rights and obligations related to the use of the code. As a result, management is often surprised by the actual content of their code. In fact, on average, the amount of OSS found upon forensic examination of code by Palamida, Inc., a company that specializes in forensic examination of software, has been on average three to five

---

<sup>11</sup> For the last few years I have been the primary Microsoft Attorney responsible for TDD for Microsoft's M&A activities, including over 50 acquisitions and major license-in transactions. In 2008 alone Microsoft acquired over 500 million lines of code. In general the acquired code contained between ten and ninety percent third-party code. About one half of the third-party code was OSS.

times the amount disclosed.<sup>12</sup> If you acquire a company that uses OSS, and do not know the composition and origin of that code, or for that matter whether it is being used in compliance with the licenses that govern its use, you run the risk of buying an infringement claim.

In most cases the existence of OSS in a particular code base should not jeopardize an M&A deal, but it is always a challenge to determine the exact makeup of the code, and it is almost always necessary to mitigate risks by removing and replacing certain components of the code prior to taking delivery. Whether or not OSS is going to be problematic depends upon how the code is intended to be used by the acquiring party, and whether that use is consistent with the OSS licenses that apply. During the TDD process the compatibility of the code with the acquiring party's licensing models are examined. Incompatible code can be removed, rewritten, or replaced with code of equivalent functionality.

### IP Chain of Title Issues

As stated earlier, OSS is merely third-party code which is subject to third-party IP rights like other third-party code, but OSS has a unique set of challenges due to the collaborative nature of OSS development.<sup>13</sup> Most OSS projects are Web-based, and some are accessible by anyone who wants to contribute to the project. If a project is unregulated, it may be difficult if not impossible to determine the origin of the code that resides in the project. The amount of control over contributions to any particular OSS project varies from project to project. Some projects have tight control over what can be introduced into that project's code base, while other projects have little or no oversight. The important thing to understand is that OSS code may include unidentified third-party content that is not properly licensed. Some OSS projects have good reputations for quality and sound IP management practices, while others have known IP infringement issues. Therefore, in addition to identifying the license that applies to an OSS application, it is also important to investigate OSS products or projects for known IP pedigree concerns before distributing that code.<sup>14</sup>

---

<sup>12</sup> Information provided by Mark E. Tolliver, CEO, Palamida, Inc. <http://www.palamida.com/about>.

<sup>13</sup> See Eric Raymond, *The Cathedral and the Bazaar*. Open source evangelist Eric Raymond suggests a model for developing OSS known as the Bazaar model. In such a model users are treated like co-developers and so they are to have access to the source code and are encouraged to submit contributions to the software, including bug fixes and reports for the software and documentation.

<sup>14</sup> Some OSS projects have been repudiated by their authors following IP infringement actions against their projects, while other projects have known IP issues such as outstanding patent infringement allegations.

## Confusion between Open Standards and OSS

There is often much confusion about the difference between “open standards” and OSS. An open standard is a publicly available “specification,” whereas OSS is software that is subject to an “Open Source” license, and which may be used to implement an open standard in a particular product or service. Whether a standard qualifies as “open” has nothing to do with the type of software used to implement that standard or the license controlling that software. In fact, open standards are agnostic with regard to software licensing or business models, welcoming all models and favoring none, and so it is equally possible for an open standard to be implemented in proprietary software as it is in OSS. For example, HTML is an open standard. Firefox is a Mozilla OSS Web browser that implements the HTML standard. Similarly, Internet Explorer, a Web browser supplied by Microsoft which is not OSS, also implements the HTML standard.

## OSS RELATED LITIGATION AND ENFORCEMENT ACTIONS ARE

**COMMONPLACE** • The litigation between Trend Micro, Inc. and Barracuda Networks, Inc. and between Microsoft Corporation and TomTom NV<sup>15</sup> is proof that OSS and other “Free” software are not free of litigation risks. In fact, litigation is a commonplace occurrence regardless of the software development model used or the license under which the software is obtained.<sup>16</sup> IP disputes related to OSS are also not limited to disputes between commercial enterprises. The Free Software Foundation (FSF) and the Software Freedom Law Center (SFLC) have aggressively pursued and even sued certain companies, even very small OSS developers, for failure to comply with the terms and conditions of the GPL license.<sup>17</sup>

The OSS enforcement efforts initiated by the FSF and the SFLC that have concluded publically have ended in settlement, and the terms of those settlements have reportedly

<sup>15</sup> See *complaint*: <http://assets.bizjournals.com/seattle/pdf/techflash/tomtomComplaint.pdf>

<sup>16</sup> See Francis M. Buono & McLean Sieverding, *Trend spotting: Recognizing the Growing Risk of IP Litigation Facing OSS Developers and Implementers*. The Metropolitan Corporate Counsel, September 2008, available at <http://www.metrocorpocounsel.com/current.php?artType=view&artMonth=September&artYear=2008&EntryNo=8702>.

<sup>17</sup> See, e.g., *Anderson et al. v. Monsoon Multimedia, Inc.*, No. 1:07cv8205 (S.D.N.Y. 2007) (settled); *Andersen et al. v. Verizon*, No. 1:07-cv-11070 (S.D.N.Y. 2007) (settled); *Andersen et al. v. Xterasys Corp.*, No. 07-CV-10455 (S.D.N.Y. 2007) (settled); *Andersen et al. v. High-Gain Antennas*, No. 07-CV-10456 (S.D.N.Y. 2007) (settled); *Anderson v. Extreme Networks, Inc.*, No. 08-CV-6426 (S.D.N.Y. 2008); *Anderson v. Super Micro Computer, Inc.*, No. 1:08-cv-05269-RMB (S.D.N.Y. 2008); *Anderson v. Bell Microproducts, Inc.*, No. 08-CV-5270 (S.D.N.Y. 2008); *MDY Industries, LLC v. Blizzard Entertainment et al.*, No. CV-06-2555-PHX (D. Ariz. 2008).

imposed significant obligations on the OSS developers involved.<sup>18</sup> In their recent GPL Compliance Guide, the SFLC set forth its “standard” settlement demands, which includes: 1) compliance with all OSS copyrights, not just the program at issue; 2) notification to past recipients of the program; 3) appointment of a OSS compliance officer, and; 4) periodic compliance reports.<sup>19</sup> Reportedly, previous FSF and SFLC settlements have also included a significant monetary component. It is interesting to note however, that the Guide to GPL Compliance only addresses the clear cases of GPL violation. It avoids addressing the more complicated and controversial ambiguities in the GPLv2, such as exactly when using a GPLv2 licensed program with a proprietary program in a particular manner will cause the combination of the two to be governed solely by the GPLv2. For example, Section 3.1 of the GPL Compliance Guide states: “Most companies accused of violations, however, lack a basic understanding of how to comply even in the straightforward scenario. This document provides that fundamental and generally applicable prerequisite knowledge. For answers to rarer and more complicated legal questions, such as whether your software is a derivative work of some copylefted software, consult with an attorney.”<sup>20</sup> In light of such ambiguities, the aggressive stance taken by the FSF and the SFLC, and the nature of previous settlement demands, developers and users of GPL-licensed software should be careful to understand and comply with all conditions and obligations of those licenses.

**OPEN SOURCING PROPRIETARY SOFTWARE** • Software developers may want to release code that was previously distributed only under a proprietary license under an OSS license, or “Open Source” the code. Some common reasons to Open Source code may include the desire to help their related products achieve widespread industry adoption, to allow the community to develop a product they are not interested in monetizing, or the desire to simply give back to the open source community in exchange for the use of Open Source materials. Although there may be many viable business reasons to Open Source a product,

---

<sup>18</sup> See [http://www.news.com/8301-13580\\_3-9808378-39.html](http://www.news.com/8301-13580_3-9808378-39.html).

<sup>19</sup> See: Kuhn, Williamson & Sandler, A Practical Guide to GPL Compliance, (Software Freedom Law Center 2008) available at <http://www.softwarefreedom.org/resources/2008/compliance-guide.pdf>.

<sup>20</sup> See also The FSF's FAQ to the GPLv2 at <http://www.gnu.org/licenses/old-licenses/gpl-2.0-faq.html#MereAggregation>. “What constitutes combining two parts into one program? This is a legal question, which ultimately judges will decide. We believe that a proper criterion depends both on the mechanism of communication (exec, pipes, rpc, function calls within a shared address space, etc.) and the semantics of the communication (what kinds of information are interchanged).”

Careful business and legal review should be considered prior to such a release in order to help identify and balance the potential benefits with potential adverse impacts on the distributor's IP. Distribution of software under the GPLv2 for example, could include a broad license to the licensor's copyrights and an implied license to the licensor's patents necessary to use the code distributed under the GPLv2. The effort required to identify a licensor's patents that read on the distributed code may be costly and time consuming depending on the technology involved.

Also, source code often contains comments and internal references inserted by its developers. When code is developed for internal use, or is compiled into object code prior to shipping, those comments never see the light of day. Because such comments were not intended for external consumption, they could contain content that is inappropriate or embarrassing. Therefore, it usually is a good practice to review source code and documentation for inappropriate content.

### **Third-Party Code and License Unification**

Developing software often involves licensing-in or purchasing third-party code that becomes part of the product code base. Some common commercial licensing terms governing such code prohibit re-licensing under an OSS license or disclosing the source code. Open Sourcing or disclosing the source code for such code may constitute a breach of contract or cause the original license to lapse. Additionally, software that has been in inventory for many years may contain code of unknown origin, some of which may include such restricted third-party code, for which the licenses may be difficult to locate. In order to prevent the inadvertent distribution of third-party code that could trigger a breach of contract or give rise to a copyright infringement claim, it may be prudent to conduct TDD on your own code to identify all third-party components and their applicable licenses prior to Open Sourcing.

### **Distributing Build Environments and Documentation**

In addition to releasing the core binary and source code to an application, some applications that have been developed and maintained for internal use have associated build environments and documentation that is used by developers during ongoing development and maintenance. If the client wants to allow the OSS community to continue the development or maintenance of the product, the community may need access to the build environment, including developer

tools and documentation. Because most build environment materials were not created with the intent to distribute outside the firm, it is important to review the materials for inappropriate or confidential content before distribution.

**CONCLUSION** • IP is critical to both OSS and proprietary software, and your ability to identify the origin of the IP associated with specific OSS may be difficult or impossible. Because IP applies to software of all types, and IP litigation is prevalent regardless of the licensing model involved, it is important to determine the licenses that apply to code you want to use, and to ensure compliance with such licenses. Software developers should strive to develop practices and policies that can unlock the benefits of OSS while respecting third-party rights, and in a manner that protects their own strategic IP. Open sourcing proprietary code should be done only after careful IP analysis, and IP due diligence on your own code may be recommended before disclosing your code and build environment materials to the public.

About the author:

Jim is an Attorney with Greenberg Traurig's Silicon Valley office in their Intellectual Property & Technology Practice. Markwith's practice focuses on intellectual property (IP) transactions, with an emphasis on software and hardware development, open source, open standards and information technology (IT). Prior to joining Greenberg Traurig, he was with Microsoft Corporation, where he was a Senior Attorney in the Intellectual Property and Licensing Group and served as Open Source Licensing Counsel. From 1995 to 2005, he served as an adjunct professor at the Santa Clara University's Leavey School of Business and Administration, where he taught "The Legal Aspects of Managing Technology," a course he developed, and Macroeconomics. He also taught "Protection of IP" for Santa Clara University School of Law's LLM program. Prior to his legal career, Markwith served as a pilot in the U.S. Navy and was appointed an Instructor of Economics at the Naval Academy in Annapolis, Maryland. Markwith received his J.D. from Santa Clara University School of Law.

## ACC Extras

Supplemental resources available on [www.acc.com](http://www.acc.com)

Open Source Software- Best Practices Checklist.

Quick Reference. December 2007

<http://www.acc.com/legalresources/resource.cfm?show=16485>

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

Program Material. February 2007

<http://www.acc.com/legalresources/resource.cfm?show=20155>

Open Source Software-Managing Risks in M&A Transactions.

Quick Reference. December 2007

<http://www.acc.com/legalresources/resource.cfm?show=16484>

Please note, these additional resources are provided by the Association of Corporate Counsel and not by the faculty of this session.