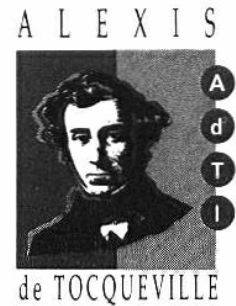


Intellectual Property--- Left?

Considerations Regarding Unchallenged Infringement
In Public Domain Software Development



Part I. A Brief Look at Conflicting Practice and Theory

“...Contributors to an open source project are usually not asked to make warranties about the originality of their code, and often contribute while employed by corporations who can make claims to the code as a work for hire. The good news, however, is that open source developments make code public, so inclusion of code in an open source product is a publicly known fact, thus there can be laches and estoppel arguments to avoid conflicting claims...”

“Buying the Penguin”, Heather J. Meeker, Esq.

Gordon G. Waggett, P.C., Reviewer
www.waggettlaw.com
gordon@waggettlaw.com

By Kenneth P. Brown
Alexis de Tocqueville Institution
www.adti.net
kenbrown@adti.net
April 4, 2005

Executive Summary

Downloadable, free software that accompanies its respective source code (commonly referred to as open source) is a prevalent phenomena. In kind, a pervasive issue regarding the development, use, and distribution of open source software is the question of intellectual property. After a brief glance at much open source software development, it becomes readily apparent that a number of open source practices directly conflict with best practices associated with protecting intellectual property.

In sum, both intentionally and unintentionally, users, developers, and distributors are in conflict with traditional, staid intellectual property law. Licensing, attribution, anonymity, derivative works, and indemnification are just a few of the well-known debates regarding intellectual property and open source products. But this point is second to an even larger issue.

Intellectual property rights are merely legal devices that must be recognized within a legal framework. Until a court says so, a property rights challenge is merely a contention or an assertion. Consequently, 1) a legal infrastructure cannot begin defending a victim without an action being filed and 2) due to rules that govern the timeliness of filing a complaint, courts will not will indefinitely protect a victim of infringement.

Meanwhile, infringement cases regarding open source software have been largely non-existent. Certainly it is improbable (and mathematically impossible) to assume that no infringement is occurring. Thus, we are left to conclude that infringement to date has largely been getting a 'pass'. However, because nothing lasts forever, it is important to discuss the timing (and the trigger) for infringement filings against public domain software to occur.

Another intriguing aspect of this discussion is the widespread practice of corporate employee contributions. Perhaps everyday, an untold amount of employees beholden to strict employee/invention/intellectual property agreements, in their spare time (and even during work-hours) freely give away ideas, code, and products to open source projects.

This phenomena prompts dozens of questions including: Could employers assert legal right to any of these contributions? Could a disgruntled employee use the public domain environment as way to surreptitiously give away company secrets? What time restrictions apply to an employer's discovery that their ideas, products, etc. were contributed illegally to a public domain project?

"Intellectual Property---Left" is a series of papers that look at each of these issues from theoretical, empirical, and financial perspectives.

- Pt. 1 Addresses theoretical questions regarding infringement and contributions.
- Pt. 2 Addresses empirical, real world observations of corporate contributions.
- Pt. 3 Addresses the impact of unchallenged infringement on a company's balance sheet.

Intellectual Property—Left?

By Kenneth Brown

Part I. Theory, Rules, and Scenarios

I. The Risks of Inaction, the Prospect of Action

It would seem that an environment enabling the exchange of hundreds of free software products and millions of lines of source code would be ridden with legal problems. However, in fact, we see the opposite; an overall lack of infringement cases against developers, users, and distributors of downloadable software with its accompanying source code, commonly known as open source software.

Meanwhile, the growth in the use of free, public domain software is widely heralded by reporters, analysts, and journalists in the media. Commentary describes open source as a choice software process and the future of technology. Some proponents have gone as far as proclaiming open source as “liberation technology” as a way for small or disadvantaged countries to finally become competitive with the U.S. and Europe.¹ A check on ‘Google’ or any other popular search engine provides almost endless references to articles supporting the growth of open source software.

Considering this widespread enamor, it can be expected that, regardless of the legitimacy, ANY challenger to open source, faces incalculable public relations concerns. The “fear” of excoriating reprisal in the media, etc. would be enough to keep many silent. Some argue that public relations concerns, not legality, may in fact be the larger reason why many have hesitated to file hostile legal actions against public domain software.

“Sue us...? It’s Not Going to Happen...”

In kind, open source advocates insist quite publicly that developers, distributors, or users of public domain software are unlikely to face significant legal challenges concerning intellectual property infringement. Dan Ravicher, president of Open Source Risk Management, an insurance firm for users of free software comments, “...Further, most patent holders don’t want to kill projects with their patents. Rather, they want money...For those types of patent-holders, suing a small guy who (a) can’t settle for royalties and (b) has little to no money, makes absolutely no sense. Such a revenue seeking patent holder will go after

¹ Niranjani Rajani, “Significance of Free/Libre and Open Source Software/FLOSS/for Developing Countries”
<http://www.maailma.kaapeli.fi/FLOSSReport1.0.html>

deep pockets, who will, by the fact they have deep pockets, be able to adequately mount a defense to the patent.”²

Concurring, intellectual property attorney Stephen J. Frank comments, “...With more and more innovation, companies are retreating even in circumstances traditionally justifying license royalties, the prospect that patents could doom open source grows ever more unlikely...[moreover] a community of developers could collaborate to quickly develop a non-infringing work-around. Such threats, added to the high costs of litigation, questionable returns, and the prospect of widespread rejection of open wallets for open source, will likely engender hesitation among most patent owners.”³

Some open source proponents have issued defiant, brazen challenges to the question of infringement. Stuart Cohen, CEO of Open Source Development Laboratories commented, "Over the past 18 months a handful of companies and individuals who are threatened by Linux's success have tried to argue that Linux may infringe others' software patents. "We find it interesting that none of those companies or individuals have said which patents Linux may offend. Yet patents are, by their nature, public; inventions must be disclosed in exchange for the rights granted by the PTO (United States Patent Trademark Office). Detractors of Linux on patent grounds should be asked to point to the specific patents that they claim infringe."⁴

Legal Challenges to Something Popular

With the likelihood that the exceptional unpopularity of a legal challenge to open source has had a role in discouraging action, the factor of public relations risk is an issue worth examining closer. Ironically, recent events, with curious similarity, address the costs and the benefits of corporations taking exceptional public relations risks to protect their intellectual property that has become “freely” distributed in the public domain.

For example, the music industry began its uphill effort to slow illegal downloading officially on December 7, 1999 when it filed suit against Napster.⁵ At that time, the popularity of Napster exceeded 1 million new users per week, with over 800,000 persons using the system simultaneously.⁶ The popularity of the technology established an inordinate challenge to the music industry. Looking back five years ago, its watershed court victories ending MP3.com and Napster were viewed as unexpected as the music industry’s decision to accede to the widespread demand for unbundled music. Today, not only has the music industry successfully slowed illegally traded music⁷, but stalwart opponents of

² “Patents and the Penguin”, Brown, <http://www.adti.net/penguin.html>

³ “Will Patents Pillage Open Source”, CNET, April 16, 2003, Steven J. Frank, Partner, Tesla, Hurwitz & Thibeault

⁴ “Ballmer Sets Off Skirmish Among Linux Crowd”, November 18, 2004, <http://www.linuxinsider.com/story/38294.html>

⁵ “Recording Industry Sues Music Start-up”, CNET News.com, December 7, 1999

<http://news.com.com/2100-1023-234092.html?legacy=cnet>

⁶ Testimony of Shawn Fanning, October 9, 2000, http://judiciary.senate.gov/oldsite/1092000_sf.htm,

⁷ “Labels Warming to Mp3s”, Wired.Com, June 7, 2000, Brad King

<http://www.wired.com/news/business/0,1367,36858,00.html>

illegal downloads actively partner with web portals to sell discounted unbundled music.⁸ Recording Industry Artists of America's (RIAA) wins have become a template for the Motion Pictures Association of America (MPAA) pursuit of companies that facilitate illegal movie downloads.

The recent Apple Computer controversy brings even greater clarity to the question of risk and reward to filing an unpopular lawsuit regarding intellectual property. In an unexpected ruling with widespread implications, Judge James Kleinberg of the Santa Clara County Superior Court recently ordered three online reporters to provide Apple Computer with their confidential sources for product descriptions that Apple insists were illegally leaked from its company. Apple contended that the leaks were in violation of non-disclosure agreements and possibly the U.S. Trade Secrets Act.⁹ In concurrence with Apple, Kleinberg wrote, "the right to keep and maintain proprietary information as such is a right which the California legislature and courts have long affirmed and which is essential to the future of technology and innovation generally."

Apple Computer's recent win in its effort to sue individuals that shared illegally obtained trade secrets with the public, overlays many of the issues regarding potential infringement in public software development including: 1) the prospect of an un-policed, rogue employee very easily harming a company by placing intellectual property in the public domain, 2) the financial impact of such an act upon a company's assets, 3) the exceptional amount of fall-out, and public scorn that occurs when a company decides to protect and/or reclaim its intellectual property in the public domain; 4) regardless of the unpopularity, or ramifications, if the evidence and argument is sound, there is considerable likelihood that the courts will uphold a plaintiff's right to protect their intellectual property.

...It Could Happen

Filing a legal complaint against popular public domain software is business decision that must weigh costs and benefits, risk and gain. Particularly with regard to the open source model, almost any ruling against a user, distributor, or developer of open source, could result in a precedent that could significantly change public domain software practices. The problem is that any precedent, or intellectual property judgment against the open source model, would inherently change 1) over a decade of widely accepted, un-policed practices 2) the way millions of people use, distribute, and develop open source, 3) open the gateway for deluge of similar, copycat lawsuits.

⁸ "Fightback or Deathrattle?", April 2, 2004, The Economist, http://www.economist.com/agenda/displayStory.cfm?story_id=2552490

⁹ "Apple Wins Trade Secrets Legal Dispute", March 11, 2005 Rachel Konrad, Businessweek Online, http://www.businessweek.com/ap/financialnews/D88P08FG0.htm?campaign_id=apn_home_down

It is safe to surmise that challenging public domain software infringement is as hairy a prospect as suing young teenagers for illegally downloading music, or demanding a reporter's sources for allegedly illegally leaked trade secrets. Simply put, a suit that could permanently change the public domain software model, could definitely happen. The trigger for such an event would more than likely be the set of consequences that led to the RIAA and Apple suits—sue, or face a significant deterioration in company assets.

While proponents would insist that the lack of legal challenges to open source suggest that a discussion about intellectual property infringement and free software is 'much ado about nothing', this is far from correct. The dearth of legal challenges to open source, does not remove the gravity of two looming questions: 1) whether the IT sector's continual deference to infringement has financial consequences and 2) whether the use and development of public domain software will remain a risk-free proposition. These issues become credible metrics to predicting the future of wide-ranging lawsuits against public domain software.

II. Rebellion – the Roots to “IP – Free” Software Development

A look at the history of open source provides some clues as to whether public domain software will ultimately become the target of a floodgate of lawsuits. Tellingly, the origin of open source is a model that was spawned from a movement away from standard intellectual property rules. The success of the public domain model diminishes the need (and value) of the proprietary model. Thus, if the growth of open source is consistent with its roots, an impending showdown between proprietary companies and/or intellectual property owners is highly probable.

Before there was ever a discussion about free software, there was a clash between inventors, a tension that predates patent law itself. There were inventors solely interested in the public good; and openly sharing their discoveries with other inventors. Conversely, there were inventors that chose instead to protect and commercialize inventions for profit and personal gain.

The genesis of open source code and open source software is very similar. As early as the 1970's, technology companies began building software and software-related technology for commercial purposes. At the same time, the popularity of the commercial software model included a growing group of software enthusiasts interested in freely exchanging ideas, source code, and techniques to improving software.

Software that was closed could not be improved or studied. So, in the interest of seeing the human readable code, software described as “open source” became very valuable and sought after by programmers. Today, open source refers to software that is freely downloadable, as well as including the

human-readable source code.

In his paper, "The Collaborative Integrity of Open Source Software", Associate Professor Greg R. Vetter provides a snapshot of how open source was founded upon the complete opposite thinking of its proprietary counterparts. Vetter writes,

"Given their incentives, companies tend to keep secrets, including computing secrets, at least until disclosure has a benefit or potential benefit. Rebellion against this practice is an originating factor for the open-source movement. Source code secret-keeping, and its anti-collaborative effects, however, became endemic to early computing due to two factors present from the start of computing through the early 1980s: the centralized nature of the technology and its increasing commercialization. The centralized computing design was a technological constraint during this era. This constraint, however, contributed to a regime of control over the computing assets, especially as their commercial value grew.

Corporate law requires companies to shepherd assets and employ them productively. As computing assets grew in importance, so did company practices to restrict access to various aspects of the technology, including physical access to the machine, as well as user access to the operating system and the source code of computer programs on the machine. Reducing access decreased operational disruption risks. Information technology and the computing assets became more valuable and mission critical. As a result, companies ratcheted up the control over these assets. Operating systems responded to this need by offering a hierarchy of user levels, allowing administrators into the system with full "power" to change and configure the system, while restricting users and programmers to specific environments within the hierarchy. This was the era when "big iron" dominated computing and defined the era of centralized computing. The processor was secreted away somewhere, access was via terminals, and only a few administrators had the power to range throughout the entire system. This hierarchy of control, it was thought, better protected the corporate assets. It also facilitated secret-keeping for source code."¹⁰

In the 1980's, open source software use became so popular, that a number of licenses were written to allow developers and users to freely exchange software with source code for research and commercial use. Popular examples of these licenses include the MIT open source license and the BSD open source license¹¹.

¹⁰ "The Collaborative Integrity of Open Source Software, Greg Vetter, Utah Law Review, Aug. 22, 2004, http://www.law.uh.edu/faculty/gvetter/documents/Vetter.CollaborativeIntegrityOfOpenSourceSoftware_8.22.2004.pdf

¹¹ www.opensource.org/licenses

The most controversial license, the General Public License¹² (GPL), which came about during this period was distinguishably different from other open source licenses, because its deliberate design was to thwart any attempt to own or reap ownership rights. The GPL license governs software in the tradition of open source licenses, but with proprietary license restrictions. The GPL controls future use, ownership, and distribution of its accompanying software. The dual open source and proprietary features of the General Public License explain references to GPL software as ‘hybrid source’ as opposed to open source.¹³

III. A Sizeable Machine, That Runs On Air

Undoubtedly, the proprietary software industry has grown significantly. Today, the software industry boasts over \$300 billion annually in revenues.¹⁴ Accompanying the growth in the use of proprietary software is an equally noticeable growth in the use of free software and organizations that promote the development of free software.

For example, last year, the Mozilla Foundation reported that its release of Firefox, an open source, free Internet browsing software, had achieved over 1 million downloads in a day.¹⁵

Today, it is estimated that there are over 1 million programmers that contribute source code, research, and development to improving public domain software.¹⁶ This number is especially compelling considering that there are only 2.96 million computer related jobs in the U.S.¹⁷ Another comparison that makes the size of the open source community compelling is its value in man/hours. A study that surveyed contributors to the hybrid source operating system Linux called, “Motivation of Software Developers in Open Source Projects: The Case of the Linux Kernel”, surmised that on average, developers donated 18.4 hours per week.¹⁸ At average salaries of just \$35/hour, one million moonlighting programmers donate time valued at over \$2.5 billion dollars every month to public domain software projects.

Noticeably, a number of IT companies are filled with people that have believed in free, open source software for over thirty years. In the interest of embracing the talents of these individuals, corporations such as Hewlett Packard and Sun Microsystems have underwritten collaborative efforts similar to public domain projects. In the interest of mixing cultures and achieving comity, it is not unusual

¹² www.gnu.org/licenses/licenses.html#TOCGPL

¹³ “Bottom Line: Hard Buy, Soft Sell”, Gregory Fossedal, July 26, 2004, <http://www.washtimes.com/upi-breaking/20040724-021913-7198r.htm>

¹⁴ www.siiia.net

¹⁵ www.mozilla.org

¹⁶ Mark Barrenechea, Sr. VP, Computer Associates, interview, UK.Builder.com, May 6, 2004, <http://uk.builder.com/manage/business/0,39026582,39128942,00.htm>

¹⁷ Bureau of Labor and Statistics, Occupations with the Fastest Job Growth, 2000-2010, 2.4-4.4 million computer related jobs by 2010, <http://www.dol.gov/wb/factsheets/hotjobs03.htm>

¹⁸ “Motivation of Software Developers in Open Source Projects: The Case of the Linux Kernel” <http://opensource.mit.edu/papers/preso-hertel.pdf>

for many of the fiercest opponents of intellectual property and proprietary technology to be neatly tucked away at prominent corporations dependent upon trade secrets and intellectual property.

IV. The Open Source Contributions Dilemma

As can be expected, most volunteers contributing their time, ideas, and intellectual property to the public domain, sustain themselves as contractors and/or employees of IT firms or organizations with sizeable IT departments. It is uncertain how many employees of major corporations contribute to the public domain. Many contributors are employees for large publicly traded corporations such as Toyota Europe, Kodak, and NEC.¹⁹ The phenomena of employees for large publicly traded companies donating intellectual property is particularly ironic considering the fact that most are intimately acquainted with strict employment contracts governing employee invention and ownership.

Therefore, the question of corporate employees contributing intellectual property to the public domain creates innumerable questions including: Are moonlighting employees that donate their ideas and intellectual property to public domain projects always within their employment agreements? Is anyone checking?

Background On Employee/Invention Agreements

There is a considerable range of law regarding employment and employee invention. Every contract is different (see attachment).²⁰ The general idea is that the employer is hiring an employee in exchange for his skill set to create new products or ideas of value to the corporation. The agreement for that period could include legal rights to a number of items that could be considered intellectual property such as copyright, patents, designs, trademarks, confidential information, trade secrets, and even know-how. Employers that view these products as assets, as best as they can, will use employment agreements to stipulate (as a condition of employment) that these assets are either irrevocably shared, or remain the private property of the employer.

While the theory is clear, the practice isn't; as there are innumerable variations of these agreements that stipulate ownership, assignee rights, and the residual financial benefits of the intellectual property created or improved by employees.

The discussion becomes even more varied considering the number of countries that have different laws governing the issue. Unlike Japan and Germany (which itself has an inventor compensation law, albeit one less lucrative for inventors than Japan's) the U.S. does not have any significant statutory provisions to compensate employees for their inventions. (One narrow federal statute does

¹⁹ Linux Credits Files, www.kernel.org

²⁰ Attachment – Catuity – Robert Kosnik Contract, CLE Course
<http://contracts.onecle.com/catuity/kosnik.emp.2000.05.17.shtml>

exist, which mandates compensation for inventions made by federal employees.)²¹

An article by Mayer Brown discussing German inventors and employment, pointed out, "In Germany, as in other industrialized countries, the vast majority of inventions and other creative developments are made by employees. In comparison to the laws regarding employee inventions in many other countries, German laws are unusual: as a matter of principle, an invention made by an employee belongs to the employee. The invention and the proprietary rights relating to it remain the property of the employee until they are transferred to the employer, which will only happen if the employer claims the invention in return for monetary compensation".²²

Gordon G. Waggett of Waggett Law notes, "If you are a German company with German employees operating in Germany, it is understandable that you will follow the laws of Germany, and that the employer will understand the rights of its employees in Germany. Where the issue becomes sensitive and interesting is within the multi-national or international companies that have inventors in multiple countries. Laws of this type can complicate a company's desire to have a uniform, company-wide intellectual property policy that is equally applicable to all employees."

Australia laws are interesting as well. Stuart Gibson and Virginia Wallin of the Middletons Law firm of Australia write, "When a business engages an employee or an independent contractor to develop its IT to increase its business capabilities or for subsequent sale, it is expected that they will own the right to reproduce it. However, the general principle running through intellectual property law that the author or inventor is the owner of the intellectual property means that the assumption 'I paid for it, therefore I own it' does not always ring true."²³

Gibson and Wallin continue, "In order to ensure that your business owns the intellectual properties of its employees and independent contractors the following check list sets out the more important points that should be covered or considered.

1. Check that all agreements with employees and/or independent contractors are carefully drafted and that all intellectual property rights are assigned to the business.
2. Ensure that the business obtains appropriate warranties as to ownership of any intellectual property developed by external contractors and that there is no "embedded" intellectual property that may restrict your future use or ownership.

²¹ "Smart Pills: Shifting the Balance", IP Law and Business, Paul Morico and Thomas Morrow, June 2004, <http://www.ipww.com/texts/0604/smartpills0604.html>

²² "The German Employee's Invention Act, Beware of Employee's Rights to Inventions", Mayer, Brown, Rowe & Maw, LLC. Mondaq, LTD June 11, 2004, www.mayerbrownrowe.com

²³ http://www.middletons.com.au/_site/main.html

3. Ensure that the business maintains an appropriate intellectual property register and that its intellectual property is regularly audited for ownership purposes.
4. Ensure that where a business is engaging "employees" for the purposes of creation of intellectual property that they are being engaged pursuant to a contract of service.²⁴

A Practice of Contradiction

"...That's a start. But it stops short of transferring copyright, which the OSDL says it doesn't need, and it leaves open the question of whether Linux needs to be rewritten, top to bottom, with proper attributions, permissions, and paperwork for every line of code. Torvalds and his army are "certainly capable of doing that," Cohen says, but it would demand a huge amount of time and manpower. That might be a small price to pay to protect Linux."²⁵

Thomas Goetz, Wired Magazine, July 2004

As mentioned earlier, it is estimated that there are over one million contributors of ideas, code, etc. to software projects around the world. Large software projects in the public domain are run by organizations such as the Mozilla foundation that accept code, ideas and time from contributors. Each organization has different rules in place as far as maintaining source code, diligence, and regulating authorship rights.

Corporate policies are not uniform nor are they comprehensively able to review after-hours employee contributions. An employee could be doing anything after-hours--- which is why it is called – “after hours”. In a white paper by Skadden, Arps lawyers Stuart Levi and Andrew Woodward suggest an even more ironic solution to the problem commenting, “...A company also may find, whether or not it elects to use open source, that its employees are contributing source code to the open source community. An open source policy should prohibit employees from making such contributions without appropriate approvals. In addition, to the extent that employees want to contribute code to the open source community that is unrelated to their job responsibilities, the policy should state that all such development work be done using third-party resources and email accounts and that employees not disclose the identity of their employer. This policy should complement any rules that the company may already have established regarding the right of employees to perform work in their personal capacities.”²⁶

²⁴ “Employers beware - IT Intellectual Property Ownership, Employees and Contractors”, Stuart Gibson and Virginia Wallins of Middleton Lawyers, Australia, www.middleton.com.au, <http://www.findlaw.com.au/articles/default.asp?task=read&id=8732&site=GN>

²⁵ “The Linux Killer”, Wired Magazine, July 2004, from essay entitled, “Anonymous Source”

²⁶ Stuart D. Levi and Andrew Woodward, “Open Source Software: How to use it and control it in the corporate environment”, Stuart D. Levi, partner, Andrew Woodard, associate in the Information Technology and E-Commerce Group in the New York office of Skadden, Arps, Slate, Meagher & Flom LLP.

Levi and Woodward's suggestion illustrate the lack of answers to effectively dealing with the problem. Suggesting that corporations implement an anonymous contributions policy in the interest of protection, paradoxically, reduces the employers' ability to trace infringing contributions placed within the public domain. If employees do not use an identifiable email, employers are left to 'trust' that employee contributors are not (and would not) donate infringing code. Meanwhile, what better employee to 'trust', than one who likes the fact you can't track his activities and wants to keep it that way.

Regardless of its obvious drawbacks, this in fact, is the prevalent practice for thousands of people that contribute open source. A look at most contributions boards, such as Sourceforge.net²⁷, is the location of endless numbers of programmers, developers, etc. contributing to open source projects...most of which with email addresses that have no link or means to tie them back to their employer.

Commenting on obvious policy problems, Waggett comments, "Part of the anonymity (as to the programmer's employer identification) appearing in these email addresses may stem not from a particular policy regarding open source, but from a commonly used policy stating that an employee is not entitled to use resources of the company (e.g., its email system) for anything other than company business. As such, requiring an employee to use the company email to conduct non-company open source work would require yet another level of approval. It would appear that the best position an employer can take is to create a sound policy, ensure that it does its best to inform each employee of the policy, and to encourage that employees having any issue concerning such policy raise them with management. It will be up to management to create and uniformly enforce such policy. However, the mere existence of a policy, or even a written agreement, does not guarantee that such policy or agreement will remain unbroken – the hope with such policies and agreements is that honesty will prevail. If honesty does not prevail, then enforcement must follow."

In sum, employee contributions are a perpetual intellectual property infringement problem regarding public domain software for three reasons:

- 1) There are a wide range of employee/employer agreements regarding authorship and ownership of intellectual property, with no apparent uniformity regarding open source contributions. Meanwhile, companies have no way or means to govern contributions because at present the situation is without focus and amorphous.
- 2) Organizations managing free software projects do not engage in thorough diligence regarding contributions. Moreover, they have no means, risk, or gain to do so.

²⁷ www.sourceforge.net

- 3) Due to the widespread use of anonymous identities, even if corporations were interested, they could not properly police employee contributions.

V. Hypothetical Questions

The environment of contributions to the public domain precipitates a range of possibilities, many pointing to considerable legal implications. The following are a set of scenarios and commentary to address some of the more conspicuous problems with the model.

Scenario #1 Hiring/Contracting Activist Employees

Today, a number of members of the scientific, academic, and hobbyist community who are openly adverse to traditional intellectual property law practices, are employees at proprietary technology companies. These activists are not just advocates for open source software but 1) are against software patents and 2) believe that digital, electronic technology should not be owned by anyone, thus should all be free.

This of course is a precarious situation for employers. After-hours projects by nature are characterized by a time period that an employer has substantially less ability to survey an employee's work product, or supervise his time. As previously mentioned, in most cases, the employer relies on an employment contract covering intellectual property and inventions to clarify the relationship between the employer and employee after-hours as well give the employer a right to enforce any breaches of the agreement. But the larger question is, "if you are company whose business model leveraged the value of trade secrets, would you hire a confessed anti-intellectual property activist"?

James Kirk, an intellectual property attorney comments, "these individuals are usually the leading minds in the industry. Corporations need to recognize this potential conflict at the onset and structure their hiring process accordingly. I am not saying don't hire people with this mind set, but it does need to be addressed in the hiring negotiations. The leading minds are working for corporations to earn money in return for their development of product. A solid employment agreement should be ironed out between the employer and the employee clearly stating the ownership of intellectual property and the consequences for distributing this property in violation of the employment agreement."

When asked if hiring employees with an activist mindset was precarious, William Jensen of Shook, Hardy, and Bacon comments, "...Not if the employees are required to sign an employment agreement that restricts their developments to internal use, and impose strict standards of confidentiality. The US patent statutes are tailored to improve technology through the open disclosure of inventions in exchange for patent rights that exclude others from profiting from

the invention for a term of 17 or 20 years. Open Source software is similarly tailored to improve technology through the open disclosure of inventions in exchange for rights that exclude others from profiting from the invention if they use or improve the copied software. It's a contractual quid pro quo. Those that don't like it are free to develop and patent their own software.”

Certainly, inside and outside counsel for corporations have created policies for employers and employees to jointly participate in projects that donate intellectual property. But are these programs enough? Jensen comments, “An employee can't give away what he/she doesn't own. If the employer has a solid employment agreement providing that the employee assigns, or agrees to assign, what he/she was hired to develop and the assignment is recorded before the employee attempts to give away the invention, then any subsequent assignment by the employee is invalid and unenforceable against the employer.”

Elaborating on this point, Waggett notes, “Although it is possible to extract a written assignment from the employee for each tangible development that was invented by the employee, as a practical matter, the only specific assignments that are typically prepared and later “recorded” publicly are those pertaining to inventions made the subject of a patent application filed in an applicable patent office. If the development is “confidential” or a trade secret of the employer, it is highly unlikely that there will exist any “public recordation” of any assignment from employer's employee pertaining to such trade secret or confidential information, and even if there was, it certainly would not provide disclosure of the secret. As such, it is difficult for an employer to realistically rely on placing future third party potential bona fide purchasers of technology on notice of its (rather than its employees') superior ownership rights. In this scenario, the importance to the employer is in the employment of one of the brightest minds in the industry. Assuming that the employer is aware of the “sharing” propensities or mindset of this prospective employee, the most important action in my view is for the employer to discuss from the outset a clear understanding of the employee's legal boundaries regarding technology that is regarded as important to the employer, and to reduce such discussion into a written agreement. However, unless the employee is being hired for a highly definable job/task/project, it will be difficult to set out in an initial employment agreement all of the specific details that an employee might be called on to develop/invent. Additionally, although it is always the best practice to provide as much detail as possible in an employment agreement, an employee may nonetheless be subject to restrictions that are implied at law even without an express written agreement – one notable restriction being that in many jurisdictions, an employee does not have a right to improperly disclosure confidential information or trade secrets belonging to its employer during or after employment, and doing so may also subject the employee/former employee to criminal prosecution for theft of trade secrets. In a nutshell, ironing out a firm, written understanding with employees regarding their duties and obligations respecting intellectual property will assist in managing these potential risks, and may actually provide an avenue of safe harbor for the

employee to participate in activities that are “outside the scope of employment” (translated: outside the economic interests of the employer).”

Scenario #2 Reclaiming Code in the Public Domain

The ability of an employment agreement to govern independent work that is created and donated to the public domain is a compelling issue as well. The following scenario is an example of this complexity:

Company XYZ develops technology for oil processing. Mr. Scientist on his off-hours (with his own resources) has developed a software for oil processing, but he wants to give it away. He wants to donate it to a non-profit as a learning tool, with no restrictions.

Can XYZ company assert ownership to this technology?

Kirk comments, “It depends on why Mr. Scientist was hired. I believe there is a strong argument that Mr. Scientist was hired specifically to develop oil processing software, as such, the development of the new code is the property of the XYZ corp. While Mr. Scientist used his own time and resources to develop the software, he probably used his knowledge gained through XYZ to develop this software. There is also a likelihood that Mr. Scientist is developing a similar software for XYZ and has learned from the development of this software to develop his own software. While I believe that there is an argument to be made that this software belongs to XYZ, it will be an uphill battle if the employment agreement is vague as to the responsibilities of Mr. Scientist.”

Noting the complexity of this situation, Waggett comments, “I agree that a vague agreement can add difficulties to the enforcement of the agreement, but keep in mind my earlier comments on the unrealistic probability that an employment agreement will spell out in detail all of the tasks that an employee might undertake while employed. This difficulty would be exacerbated if the agreement were completely silent on any duty to assign inventions (whether patentable, copyrightable or not). With a written duty to assign all inventions, I believe that XYZ company would have the upper hand in arguing that this invention relates to its business and therefore belongs to XYZ company even if technically developed “during Mr. Scientist’s off hours”.

Waggett continues, “This issue would also depend highly on whether Mr. Scientist was exposed to the specific XYZ corp code at issue, or perhaps even whether he was exposed (for the first time while at XYZ Corp.) to the need for the development of such code – all of which might be regarded as confidential information of XYZ company. If confidential or trade secret information of XYZ Company is involved, Mr. Scientist would have a legal obligation to not use or disclose such information except for the benefit of his employer. Additionally, if

Mr. Scientist used existing copyrightable code belonging to XYZ Corp., then his development could be regarded as a derivative work which would place copyright law restrictions on his use of such derivative work (independent on whether or not he owned the new code).”

The problem with this scenario of course goes back to the intent of the employee. Obviously intent again brings into question again whether its in an employer’s best interest to hire a demonstrably activist employee. Waggett points out, “If Mr. Scientist’s off-hours development was completely independent of any information he gained at XYZ Corp., this may provide Mr. Scientist with a stronger position of ownership, particularly if Mr. Scientist’s scope of work at XYZ Corp had nothing to do with software development. However, if his “independent” invention relates to the same technology area as that of XYZ Corp., and would otherwise appear to fall within the area of his employment, I believe that Mr. Scientist will have a more difficult time arguing that he came up with this invention outside of the scope of his employment (perhaps it would also be argued that Mr. Scientist was not using his best efforts while on the clock, but rather reserved those best efforts for his after hours endeavors. If Mr. Scientist offered the idea to his employer and the employer expressed no interest, Mr. Scientist would have an argument, if not an express agreement, that XYZ Corp has given him all rights to exploit the invention.”

Ultimately, the employment agreement is very clear about ownership. “Even if Mr. Scientist has a rightful claim of ownership to the technology, XYZ Corp may, under applicable law, have a “shop right” which would give it a right to use the technology”, says Waggett. “Apart from any written obligation to assign, Mr. Scientist might also have a clause in his employment agreement requiring that he take no action that would be detrimental to the employer – a duty of loyalty and non-competition (and to use best efforts). This clause might prevent Mr. Scientist from engaging in such independent research to the extent that the result of such research is not intended to go to XYZ Corp,” Waggett adds.

In many cases, scope of employment arguments would eliminate Mr. Scientist’s rights to give “his” IP away with out XYZ’s approval. But what happens if it is determined that a breach has in fact occurred? Posed with the question of whether XYZ company could assert ownership of this technology, Jensen comments that XYZ company would prevail. Jensen states, “...most likely if the employee was hired to invent, was contractually obligated to assign the software to the employer, or was in a position owing a fiduciary duty to the employer.” Concurring, Waggett comments, “I agree. If a breach has occurred, and XYZ Company has superior legal rights, I would envision that XYZ Company would be in a position to take legal action, if it chooses to do so, to prevent such third parties from using same. XYZ company will also have to consider taking appropriate action with its employee.”

Scenario #3 User Liability

Logically, the next question becomes, could misappropriated intellectual property in the public domain become a liability to users? The question of innocently using intellectual property thought to be “free” is compelling. Companies invest time, labor, and considerable resources into the development of free software. However, the corporation that has been using free open source code it downloaded years ago could face a compelling situation as well.

The scenario is:

The same company XYZ finds out that rival, ABC corporation, is using the technology put in to the public domain by Mr. Scientist without XYZ's approval. Does ABC have any legal concerns? How much claim can XYZ exert to their property illegally placed in the public domain?

Jensen comments, “Sure-if it was misappropriated by ABC through the employee, knowing if belonged to XYZ, or it was patented by XYZ.” David Ostfeld an intellectual property attorney with Adams and Reese comments, “depending on the facts, the user may or may not be subjected to liability, but if the owner never gave it away, the owner has the right to reclaim the property.”

Elaborating on further implications of the scenario, Waggett comments, “If Mr. Scientist had no rights in the invention to begin with, then ABC can not acquire any better rights than those of Mr. Scientist. In the case of a patent, if Mr. Scientist has no ownership or other right in the invention that was patented, then ABC cannot claim to have any sort of license right from Mr. Scientist. However, if the issue remains unclear as to whether or not Mr. Scientist retained any ownership of the invention, then you can expect a defensive fight from ABC company if the economics of continued use of the technology justify the fight. An interesting issue might be where Mr. Scientist occupied an official role within XYZ company, such as in the capacity of an officer (e.g., vice president of technology) in which case, third parties receiving permission from Mr. Scientist might perceive such permission as permission from XYZ company – but if such permission was not given in the name of XYZ company, but instead was held out to be the personal property of Mr. Scientist, then this argument will be difficult for ABC to rely on. Given that ABC corporation is a rival of XYZ company, if ABC corporation knows that this code came from Mr. Scientist, and it knows Mr. Scientist works for its rival XYZ company, in my view, ABC corporation would be on notice of a potential claim by XYZ company to superior rights in the technology. If ABC is at all surprised to see such code apparently being given away by an employee of its competition, it would have to consider whether such donation was legitimate or not, and should perhaps raise an internal red flag as to whether incorporating such technology might create a legal liability.”

The Bottom Line

The legal issue of after-hours contributions is tied directly to the enforceability of the employment contract and subsequently, the leniency of the employer. These brief scenarios point to three realities:

1) If an employee has signed an employment contract that covers inventions and intellectual property, an employer could have legal claim to any and all work produced by the employee within the scope of employment.

2) If the employment agreement clearly spells out ownership, an employer retains rights to their intellectual property. An unwillful donation of an employer's intellectual property into the public domain by an employee, does not nullify the employer's ownership. In fact, the employer retains legal recourse to reclaim its property.

3) A user of infringed property may have invested significant time and resources into source code he assumed was "free software". However, if an owner could prove that their IP was illegally donated by employee, the employer could "reclaim" ownership, and the user could face legal challenges from the employer.

In layman's terms, if Bill is hired to build cars for Joe:

#1) Bill can't give away Joe's car without Joe's authority--- because it is not Bill's car to give away.

#2) If Joe's car is given away without Joe's authority--- Joe has a right to reclaim his property

#3) If Bob has innocently titled Joe's car, that he thought was free, Joe still has a right to reclaim the car, because Joe never "legally" gave it away. Unless Bob can work out something with Joe, unfortunately, Bob may just have to get another car.

Estimating the Size of the Breach

With the considerable number of employees working after-hours on open source projects, it is unlikely that all of these employees, all of the time, can be working within their employment contracts. It is unlikely that every breached employment agreement would be challenged due to a wide range of factors, particularly, monetary remuneration. An effective metric to estimate the number of infringement problems with public domain software would not be to approximate

the number of employee breaches, but rather to estimate the number of legal challenges employers would follow through with.

In a study of almost three million patents called, “Valuable Patents”, it was determined that 1% of all patents filed were litigated and defended in court by their owners.²⁸ 1% is not a very significant number, however, it is not negligible. Any percentage of the total amount of people working part-time for open source projects makes this infringement considerable. Using this study as a metric, if 1% of all people contributing to public domain projects led to contested infringement, this would suggest that there could be as many as 10,000 independent challenges to software in the public domain based on patent theories alone – this number would arguably increase if one were to factor in all non-patent related challenges that could be brought relative to the software (including copyright infringement and theft of trade secret source code, etc.).

With the predominant lack of interest in filing suit, it is likely only a corporation faced with an absolutely unavoidable situation would be a plaintiff in an infringement case against an employee. Waggett elaborates on this point, “I am not so sure that a company would bring a suit against a present employee for infringement. I suspect, if the circumstance is egregious enough to warrant filing suit, then the employee would likely have been discharged from employment. Also, the more likely target is typically the business entity that stands to profit most from the infringement, e.g., a new employer of company’s former employee, or the recipient of the infringing material.” Meanwhile the others will either take disciplinary action, not act, or not pay any attention to the issue at all. The question of waiting or not to wait, brings the discussion a final intriguing set of scenarios.

²⁸ http://www.researchoninnovation.org/tiip/archive/2003_4_d.htm

VI. Laches, Estoppel, and Public Domain Software

*“Filing a claim promptly, and without first going through the cease-and-desist process, also lessens the risk of a **laches** defense...In sum, in order to protect its rights, a patent holder or licensee should make reasonable efforts to police the marketplace for infringers and, upon discovering that its patent is being infringed, promptly file suit...”*

*“Resting on Your Rights May Cost You”, John P. Halfpenny, Esq.,
James L. Griffith, Sr. Esq.
March 2002, Intellectual Property News, www.klettrooney.com*

*“Time only seems to matter when it's running out.”
--Peter Strup*

While there is comity between the proprietary community and the open source community, there are firewalls between both communities; offensive and defensive strategies for self-preservation, and perpetuation. As the proprietary community grapples with the growing competition to proprietary products, it faces another dynamic of unchallenged infringement that exponentially increases the gravity of the issue.

Laches, an effective legal defense against intellectual property infringement challenges, limits the window of time intellectual property holders have to challenge infringement. Unreasonable delay of a matter widely known and recognized could forfeit a claim of infringement. Laches and estoppel can come into play a number of ways. Laches will bar recovery by the patentee of any damages prior to the filing of the suit, while estoppel can act as a complete bar to recovery by the patentee, including the denial of an injunction against further infringement. Some problematic laches and estoppel issues include (1) whether another pending lawsuit will excuse laches; (2) when silence and inaction by the patentee can create estoppel; and (3) when the patentee has "knowledge" of infringing activity such that the clock for laches begins to tick.²⁹ Simply put, if you "sit on your rights," you lose them.

As Strup suggests, the issue only matters when time becomes a concern. There is a six-year statute of limitations for damages from patent infringement, while copyright infringement has a three-year statute of limitations (17 USC 507).³⁰ Generally speaking, the period of delay for laches for a copyright or patent infringement claim runs only from the time that the plaintiff knew or should have known about an actual or impending infringement.

This added consideration complicates the question of asserting ownership even further. John Orange, an intellectual property lawyer with Blake, Cassels & Graydon comments, "...There is also the PR nightmare if a company tries to enforce its rights against the distribution of code in which it may have a proprietary interest. For real legal gymnastics, a situation I have had is where a

²⁹ "Laches and Estoppel: The Patentee", Eric Gutttag, American Intellectual Property Law Association, www.aipla.org. October 2004

³⁰ AdTI correspondence with Alan Unikel, Seyfarth Shaw, LLP, November 15, 2004

third party contributes code to an open source environment that, if used, would infringe my client's patent rights. If you try to give notice you incur the wrath of the community, if you do nothing you have a laches and acquiescence issue, if you go after the individual, you have no material redress and the PR nightmare, or do you go after the company whose employee contributed the code on the basis of inducement to infringe? What an interesting time we live in!"³¹

For proponents of public domain software, laches and estoppel is not just a firewall against lawsuits, but something that many are using as a calculated risk to continue promoting free software projects. In a paper entitled, "Buying the Penguin", Heather J. Meeker summarizes this point writing:

, "...How do I perform diligence on all the contributions to an open source project? This will depend on how the project has been run by the target. A well-run software development project, open source or otherwise, has a "gatekeeper" who determines what modifications go into what releases. The ability to diligence the contributions (and clear the copyright interests in them) will depend on the procedures used to accept contributions. It is often quite difficult, or impossible, to get the level of comfort on this issue with an open source project that an acquiror would demand from a traditional software development. Contributors to an open source project are usually not asked to make warranties about the originality of their code, and often contribute while employed by corporations who can make claims to the code as a work for hire. The good news, however, is that open source developments make code public, so inclusion of code in an open source product is a publicly known fact, thus there can be laches and estoppel arguments to avoid conflicting claims."³²

Corporations that actively participate in open source projects and actively recruit employees who are public advocates for free software, will be faced with an inexorable task of demonstrating that they were unaware that their intellectual property was being "donated" to the public domain. Meanwhile, the Meeker paper provides a clear connection between corporate negligence regarding employee contributions and the anticipation by open source advocates that corporations (for whatever reason) will not react in time to even address any infringement.

³¹ AdTI correspondence with John Orange, Blake, Cassels and Graydon, November 12, 2004

³² <http://www.gtlaw.com/pub/articles/2003/meekrh03b.asp>

VII. Conclusion

Although the decision to pursue an infringement case against an inventor is complicated, the outcome is binary. Either the corporation finds a way to protect its intellectual property, or it is lost. "Valuable Patents" suggests that the pursuit of patent infringement is directly connected to the owner's calculation that there are significant gains or losses involved.

The defensive tactic of laches and estoppel argues something as significant. Certainly a \$200 million claim is cost-beneficial to pursue, but what about a claim that is worth just \$10,000 yearly to a company? Does the decision not to sue remove \$10,000 from the value of the company? Projecting that there are thousands of infringement cases that owners decide not to pursue, it would not be far-fetched to surmise that billions of dollars in company assets are indirectly erased annually due to laches and estoppel.

The compelling nature of these scenarios prompts us to study real world practices at both open source non-profit organizations as well as corporations with employee contributors to determine how real the problem is. From there, after assessing the problem, it becomes easier to predict the financial impact, if any, of unchallenged infringement on the balance sheets of small, mid-size, and large corporations.