



Santa Clara High Technology Law Journal

Volume 15 | Issue 2

Article 6

January 1999

The Tension Must Break: The Irreconcilable Interplay between Antitrust, Defenses to Infringement and Protection of Standardized Software Development Tools

Wendy Milanese

Follow this and additional works at: <http://digitalcommons.law.scu.edu/chtlj>

 Part of the [Law Commons](#)

Recommended Citation

Wendy Milanese, *The Tension Must Break: The Irreconcilable Interplay between Antitrust, Defenses to Infringement and Protection of Standardized Software Development Tools*, 15 SANTA CLARA HIGH TECH. L.J. 407 (1999).

Available at: <http://digitalcommons.law.scu.edu/chtlj/vol15/iss2/6>

This Comment is brought to you for free and open access by the Journals at Santa Clara Law Digital Commons. It has been accepted for inclusion in Santa Clara High Technology Law Journal by an authorized administrator of Santa Clara Law Digital Commons. For more information, please contact sculawlibrarian@gmail.com.

**The Tension Must Break: The Irreconcilable Interplay
Between Antitrust, Defenses to Infringement and Protection
of Standardized Software Development Tools***

Wendy Milanese[†]

TABLE OF CONTENTS

I. INTRODUCTION..... 408

II. AN OVERVIEW OF THE TECHNOLOGY..... 410

 A. *Development Tools in Software*..... 410

 B. *Languages and Libraries* 412

 C. *The Implementation of Development Tool Technologies*... 413

 1. *Language Dependent Development Tools* 413

 2. *Stand-Alone Libraries, Modules and APIs*..... 414

 D. *The Standardization Problem*..... 415

III. CURRENT INTELLECTUAL PROPERTY PROTECTION FOR SOFTWARE –
THE RIGHT TO KEEP WHAT YOU OWN..... 417

 A. *Patent Protection for Software* 417

 B. *The Scope of Copyright Protection for Software*..... 418

 C. *The Use of Trademark Licenses as a Pretext for
 Technology Licenses*..... 419

IV. ANTITRUST THREAT TO SOFTWARE STANDARDIZATION 420

 A. *Modern Jurisprudence and its Relationship to
 Standardized Development Tools* 420

 1. *Section 2 Liability for Monopoly Conduct*..... 420

 2. *Particular Issues With Software Standardization*..... 422

V. THE COUSINS OF ANTITRUST: DEFENSES AVAILABLE TO INTELLECTUAL
PROPERTY INFRINGEMENT..... 424

* Copyright © 1999 Wendy Milanese

[†] J.D., Santa Clara University, expected May, 1999. The author wishes to thank the board of Volume 15 for working so hard this past year, for making life as the managing editor of this volume very enjoyable, and for all their help on this comment. The author also wishes to thank her family and Aaron Fritz for all their support.

A. Patent Misuse	425
1. Patent Use Reform Statute.....	426
2. <i>Braun v. Abbott</i> : The Law of Patent Misuse in the Federal Circuit.....	427
3. Interplay Between Antitrust and Patent Misuse.....	428
B. <i>Implied License as Defense</i>	430
C. <i>Equitable estoppel</i>	432
D. <i>The Child of Patent Misuse: Copyright Misuse</i>	433
VI. COMPETING POLICY CONCERNS AND THE CONSTITUTIONALLY GRANTED MONOPOLIES	435
A. <i>Juxtapositions of Contradictions</i>	435
B. <i>A Balance Must Be Struck</i>	436
VII. CONCLUSION	437
I. INTRODUCTION	

Java's write-once-run-everywhere capability, along with its easy accessibility, have propelled the software and Internet communities to embrace it as the de facto standard for writing applications for complex networks.¹

Companies fight tooth and nail for their development tool software technology to become a standard in the software industry. With a "standard" designation comes the developer community's support vital for a company's success,² creation of third party products employing the inventor's technologies, and brand recognition far beyond the initial creation of the standard.

Although it seems to be firm law that a holder of software intellectual property is not required to license its technology to any third party,³ particular questions about this general rule arise when a

1. *Sun Microsystems Press Release* (visited Jan. 31, 1998) <<http://www.sun.com/smi/Press/sunflash/9601/sunflash.960123.10561.html>>.

2. The industry uses the number of software developers using a particular technology as a hallmark of a software technology's success. This measurement was indicative of Apple's downturn in recent years. Developers began to develop fewer Apple programs because of the company's problems, and consumers began to purchase fewer Apple products because there weren't enough applications to run on them; the problem is a vicious circle. The number of developers attending Apple's annual conference had been declining radically, a problem for Apple that is beginning to reverse itself with the company's recent rebound. See, e.g., Connie Guglieimo, *Mac Developers Get Their First Serious Look at New OS*, INTERACTIVE WEEK ONLINE, May 14, 1997, (visited Jan. 20, 1998) <<http://www.zdnet.com/zdnn/content/inwo/0514/inwo0006.html>>.

3. See *infra* Part III.A., for discussion on licensing in patent law.

software development tool becomes a standard. The creator or assignee of standardized technology at first glance seems entitled to complete ownership rights under intellectual property laws. However, if the owner of the technology refuses to license its technology to a third party, that party might not be able to compete in the marketplace if compliance with the standard is essential to a product's success. An unauthorized user of the technology could raise affirmative defenses against the owner of the technology in an infringement suit under copyright or patent laws. Alternatively, the technology owner could be held liable under any one of several antitrust theories.⁴ As a result, developers of standardized development tool and language technology effectively may be donating their inventions to the public, lest they face liability under a number of legal theories.⁵

Because of the need for interchangeability of computer programs, standardization is essential in the software industry and is desperately needed. But creators of such standards may pay a heavy price: they may not be able to effectively enforce rights against others to protect what they own. Congress must reconcile conflict between antitrust and intellectual property laws for protection of development tool technology.

This comment will provide an overview of software technology and the function of software development tools. A cursory explanation of intellectual property laws applicable to software technology will be discussed. Next, this comment will outline the antitrust laws potentially applicable to disputes involving software development tools. Finally, defenses to patent and copyright infringement actions will be introduced; several of which are closely related to the antitrust laws. Throughout the discussion, this author will seek to explain the conflicts between the intellectual property laws that protect software, antitrust law, and defenses available to an otherwise infringing defendant. The conclusion of this comment will propose a solution to some of these unreconciled conflicts.

4. For example, an owner of software technology could be liable under the essential facility doctrine. It is illegal for the controller of a properly defined essential facility to deny access to a competitor. An essential facility exists where access to a "facility" is essential for firms to compete. It is perhaps best described by example: courts have held a local electricity monopoly, a stadium, and a railroad to be essential facilities. *See e.g.* *Otter Tail Power Co. v. United States*, 410 U.S. 366, 378 (1973). In *Alaska Airlines v. United Airlines*, 948 F.2d 536, 544-46 (9th Cir. 1991), a computer reservation system was not an essential facility, where denial of access to it merely had the effect of imposing higher costs on United's rival. Alaska Airlines could still have competed without it.

5. *See infra* Parts IV and V.

II. AN OVERVIEW OF THE TECHNOLOGY

A. Development Tools in Software

Software is built by writing lines of humanly-comprehensible source code,⁶ in a particular programming language,⁷ which are then compiled⁸ into binary form⁹ that a computer can understand. This resulting code is known as object code.¹⁰ Object code is then linked¹¹ further by a compiler into code which the computer can execute, known as executable code.¹²

People can write source code in any number of different computer programming languages, depending on the purpose of the resulting application and the the timeframe for development.¹³ Many

6. Source code is defined as “[S]ymbolic coding in its original form before being processed by a computer” WEBSTER’S NEWWORLD DICTIONARY OF COMPUTER TERMS (1988). Modernly, source code is often compiled, but sometimes is automatically translated into computer instructions when a programmer programs in one of the scripting languages, such as Perl, JavaScript, or VBScript. These scripting languages stay in human perceivable form, i.e., source code, and then get automatically translated line by line by a special interpreter which carries out the commands of the scripts at the time the program is run. In contrast, languages such as C, C++, Java, and Pascal run only with compiled code. For a program to be compiled, a programmer must gather together all the necessary files and specifically instruct the compiling program, (e.g., Microsoft’s C++, Borland’s Turbo C, or Symantec’s Café (for Java)) to compile and link the source code into something the computer can understand. *See infra* notes 7-10 and accompanying text.

7. A programming language can be defined as a “set of rules that specify which sequences of symbols constitute a program, and what compilation the program describes.” M. BEN-ARI, UNDERSTANDING PROGRAMMING LANGUAGES 3 (1996).

8. A compiler program “translates the symbols into an object module, which contains machine code instructions for a specific computer.” *Id.*

9. Binary refers to the fact that the instructions the computer reads are in a series of 0s and 1s. Each 0 or 1 is referred to as a bit. If a file is in binary, it is in machine readable form. Different operating systems are often completely binary incompatible (i.e., a series of 0s and 1s may mean something completely different to a Macintosh than to a Windows machine), which is why there are so many different versions of programs for every different operating system. A byte, in contrast to a bit, is defined as a grouping of adjacent binary digits operated on the computer as a unit. *See* WEBSTER’S II NEW RIVERSIDE DICTIONARY 214 (1984).

10. Object code is output from a compiler or assembler that is itself executable machine code or is suitable for further processing to produce executable machine code. *See* WEBSTER’S NEWWORLD DICTIONARY OF COMPUTER TERMS (1988).

11. A linking program collects object files and determines the relation of the various objects to each other so that the combination of them forms an executable file. *See* M. BEN-ARI, *supra* note 7, at 35.

12. On Windows machines, this is often designated as an “.exe” file. An executable file can run all by itself, without aid from other programs. However, an executable file often is not self contained; it may need information from other run-time library files, such as .dll (dynamic link libraries) files, which, as their name implies, are accessed at the time they are needed.

13. For example, Visual Basic is a very easy environment in which to develop applications, but its programs require help from an interpreter and do not offer as robust

programming languages in modern use are owned by the companies that invented them.¹⁴ These companies often freely give licenses to distribute the end result of applications built with their languages.¹⁵

Developers often significantly shorten the development process by purchasing source code or object code tools from third party developers in the form of libraries¹⁶ or other development modules.¹⁷ Commonly used third party tools are graphic libraries¹⁸ and networking APIs.¹⁹ Along with other technologies, these tools incorporate every conceivable software application. These shortcuts for developers can be purchased in many ways: included in compilers, shrink wrapped in retail products, downloaded free off the Internet,²⁰ and licensed in deals for a purchase price of \$5.00 to many millions.

development capabilities as a language like C or C++. As such, if a company wanted to build an application very quickly for in-house use, it might use Visual Basic. But if the application needed to perform complex multi-threaded tasks, or if it wanted to distribute it beyond the company or sell it, C, Java, or C++ might be used.

14. With languages, this is usually patent ownership. The languages Java and Visual Basic are owned by Sun Microsystems and Microsoft, respectively. See *infra* Part III.B.

15. See, e.g., Sun Microsystems's Java licensing policies:

FREQUENTLY ASKED QUESTIONS ABOUT LICENSING JAVA

How do I know if I need a commercial source code license?

Commercial source licensing is required if you want to take our OEM source code and merge it into a commercial product. For example, browser companies that add the Java interpreter and class libraries to their browsers.

You do not need a source code license to write and distribute applets or applications in the Java language. Sun's binary license on our web page permits developers to write software in the Java language, as well as distribution of the binaries for the Java interpreter along with applications, at no cost.

Sun Microsystems Web Site, Press Releases, (visited Feb. 15, 1998) <<http://java.sun.com/nav/business/license-faq.html>>.

16. A library can be defined as a set of algorithms for problems that arise in many contexts. Thus, rather than programmers continually inventing or re-writing code to do the same basic things, such as concatenating words, or finding the cosine of a triangle, they can rely on libraries to do these things for them. These routines may be supplied with the computer system, bought by third party developers, or designed in-house. See P. METT, *INTRODUCTION TO COMPUTING 16* (1990).

17. For example, a class. See *infra* note 37 and accompanying text.

18. See *id.*; see also *infra* notes 39-44 and accompanying text for a description of a specific set of libraries.

19. An API, or applications programming interface, is a set of libraries accessible to a programmer so that he or she may use them to accomplish functions that the main program was intended to do. For example, Microsoft has a set of APIs available for its Office 97 suite so that programmers may program custom features into its products. See e.g., NetManage's TCP/IP software development kit (visited Feb. 18, 1999) <<http://www.netmanage.com/products>>.

20. See, for example, *Netscape Corporation, DevEdge Online* (last visited Feb. 19, 1999) <<http://developer.netscape.com/>> (Netscape's software developer site) where one can download

B. Languages and Libraries

Certain languages and tools are in the public domain and thus, are not owned by anyone. For example, the language BASIC was invented at Dartmouth College in the 1960s as an instructional programming language;²¹ C was invented by Brian Kernigan and Dennis Ritchie at Bell Labs;²² HTML, by Tim Berners-Lee at CERN; and C++, by Bjarne Stoustrup.²³ These contributors felt their language inventions should be part of the public domain, and they retain no ownership rights.²⁴ Each language has its own set of accepted standard. As a result, many public domain libraries and tools available for development exist in these languages,²⁵ and many vendors have incorporated these public domain languages into their products.²⁶

In addition to public domain languages, companies also promote their own languages for widespread use. Perhaps the newest and the most notable is Java, an object-oriented programming language developed by Sun Microsystems.²⁷ It is modeled after C++, but is designed to be small, simple, and portable across platforms and operating systems, both at the source and at the binary levels.²⁸ Java is available to any party as a language with which to write

various SDKs (software development kits) at no charge.

21. See METT, *supra* note 16, at 44.

22. See BRIAN KERNIGHAN AND DENNIS RITCHIE, *THE C PROGRAMMING LANGUAGE* (2d ed. 1988). The C language was invented in 1970, superseding a language they were formerly using called "B". See *id.* at 1.

23. See STEVE OUALLINE, *PRACTICAL C++ PROGRAMMING 4* (1995). C++ gets its title from the arithmetic instruction in C to increment a variable by one by using the "+" operator. Therefore, C++ is one generation away from C.

24. The standards for the definition of these languages are often governed by various standard setting bodies, such as the American National Standards Institute (ANSI), which set an ANSI standard for C in 1988. See KERNIGHAN AND RITCHIE, *supra* note 22, at 2. The Internet Engineering Task Force, publishes RFCs (Requests for Comments) on what should make up the official version of the language. However, vendors rarely, if ever, comply with the official, approved industry version. For example, various browser companies have greatly expanded HTML (Hypertext Markup Language) by adding additional "tags" (the formatting specifiers in HTML) to the language. The standards bodies often incorporate the new tags into the next version of standards. Because technology cycles are so fast and standards bodies take so long, the actual standard for a rapidly evolving language is never clearly set.

25. See, e.g., *The GNU Project Homepage*, (last visited June 11, 1998) <<http://www.gnu.org>>. GNU is an organization and community that maintains public domain code.

26. See, e.g., Microsoft's Visual C++ product, *Visual C++ Start Page*, (visited June 11, 1998) <<http://www.msdn.microsoft.com/developer>>.

27. See generally, *JavaSoft Web Site*, (visited May 20, 1998) <<http://java.sun.com>>.

28. See *supra* notes 6 and 9 and accompanying text; See also LAURA LEMAY & CHARLES PERKINS, *TEACH YOURSELF JAVA IN 21 DAYS* (1996).

applications, but it is not available for redistribution.²⁹

Scripting languages such as JavaScript³⁰ and VB Script³¹ have also become widespread. Scripting languages are simpler versions of languages, which are interpreted only at the time they are used, and are never compiled — they forever remain in source code form. If a software development company wants to provide scripting capability in an application, such as in a web browser, it would have to obtain a license to do so from the respective owner.³² However, anyone is free to write HTML pages on their site and incorporate these scripting languages in them. It is probable, if not certain, that a Web browser³³ not able to interpret these scripting languages is doomed to marketplace failure.³⁴

C. The Implementation of Development Tool Technologies

1. Language Dependent Development Tools

Libraries,³⁵ APIs,³⁶ or other modules used for computer programming can generally be categorized into two groups of tools. One group enables a programmer to write in a particular language. These tools would allow a programmer to copy or store data, or allocate memory on a machine necessary to run programs. For

29. See *Frequently Asked Questions About Java and JDK*, (last modified Jan. 1997) <<http://java.sun.com/products/jdk/faq/html>>. The FAQ currently states:

D1. Is Java free? Where can I get information on licensing?

The Java Development Kit (JDK) is free to download and use for commercial programming, but not to re-distribute. . .

D2. Can I use the Java logo?

Sun does not authorize the use of the Java logo. . .

Id.

30. Owned by Netscape Communications. The Java portion of the language was licensed from Sun Microsystems.

31. Owned by Microsoft Corporation.

32. See Steve Reichenenthal & Tom Bennett, *SBI: The Small Basic Interpreter*, DR. DOBBS' JOURNAL, June 1997, at 54.

33. A web browser is an application that allows one to see the graphical representation of HTML, which formats text when read by an HTML interpreter. A browser acts as an HTML interpreter.

34. This is because the use of these scripting languages is so wide spread by Web sites. A browser downloads pages of a server onto the local hard drive of the visitor's computer. The browser then interprets code embedded in the page. If a Web page incorporates JavaScript, the local browser must know how to interpret the code. If it does not, the visitor will not experience the full functionality of the Web site.

35. See *supra* note 16 and accompanying text.

36. See, e.g., Microsoft Excel's product documentation, discussing the APIs available to customize Excel. See also *supra* note 19.

example, to write code in the language Java, you *must* have the Java class³⁷ libraries. These libraries are essential to operate in a specified language; without them, there is no executable program. In other words, when a company licenses a language, it licenses the libraries that are essential to run the programs written in it. Therefore, all developers are utterly dependent on some outside libraries, even in writing the most basic of programs. This is common to all languages, regardless of type.³⁸

2. Stand-Alone Libraries, Modules and APIs³⁹

The other group of tools consists of libraries that “stand alone” and are language independent.⁴⁰ Some of these tools enable a programmer to cut time or develop something that is beyond his or her particular expertise. Other libraries are vital to a company’s survival in a specific industry. Still others provide functionality for a particular type of application. For example, a graphics application can be written in any language, but it must utilize certain graphic libraries or the industry will never embrace the end product.⁴¹

In addition, publishers of many software applications and operating systems provide functional access to their products through APIs,⁴² eliminating the need for source code or language licensing. These APIs can be either language-independent or dependent. Many of Microsoft’s products, including the Office and BackOffice families and Windows95, as well as Apple’s Macintosh operating system⁴³

37. A class is an object-oriented term referring to a “useful grouping that is based on the behavior and state that [class members] have in common.” MICHAEL GUTTMAN & JASON R. MATTHEWS, *THE OBJECT TECHNOLOGY REVOLUTION* 171 (1995). A class contains data and the methods for operating on that data. One accesses a class in a manner similar to the way one accesses a library in a programming language.

38. Some libraries essential in the programming language C are `stdio.h` (input output functions), `time.h` (calculating time), `string.h` (manipulation on strings), and `malloc.h` (memory allocation).

39. These terms are roughly interchangeable. They are often called different names based on how they are implemented, but they all achieve much of the same means, which is to allow programmability through code a third party developer has created. *See supra* notes 16 and 19.

40. Some libraries can be accessed by multiple languages regardless of the language in which the libraries were originally written.

41. For example, a game developer might use the Microsoft Direct X libraries. The developer could use other graphic libraries, selected according to how the game operates, but his or her selection must be within a certain subset of graphics libraries because the operating system and hardware companies have only tested their software for optimization with the most popular libraries.

42. *See supra* note 19 and accompanying text.

43. *See, e.g., Apple – Design and Publishing*, (visited June 11, 1998) <<http://www.apple.com/publishing/>>.

follow this model. The publishers of these products not only distribute the APIs along with their products,⁴⁴ but they also publish development kits and books, hold huge developer conferences, and demonstrate at trade shows to make adoption of their technologies quite easy. Obtaining the support of these applications developers is critical to the success of a personal computer operating system product.⁴⁵

D. The Standardization Problem⁴⁶

The creators of these development technologies want developers to adopt and use their language libraries and software code libraries as standards in the industry. To achieve this end, they endlessly tout their products' benefits,⁴⁷ offer their products for download, and provide technical support. They often do all of this for free, without even asking for the downloader's name.⁴⁸

The developer market jumps at the best of these technologies, but for much of software to work, it needs standardization. This need for standardization in software explains Microsoft's ability to secure a vast majority of the operating systems in existence on the market.⁴⁹ Network protocols and operating systems require some high level of deployment and connectivity in order for developers to create

44. *See id.*

45. *See supra* note 2 and accompanying text. There are many tools, libraries, and other languages the industry does not consider to be standards or to be necessary for software product development. *See, e.g.* Rogue Wave Tools advertisement, DR. DOBBS JOURNAL, Apr. 1997, at 8-9. Vendors, such as Rogue Wave, provide various utility and other libraries for C++ and Java programmers. Although purchasing products from a company such as Rogue Wave may save a company a great deal of time and resources than it would otherwise have to expend to build the same utilities, any company with enough time and money could build them. That is, they are not so standardized, like Java, that a company would fail if it couldn't incorporate Rogue Wave's products in its own products.

46. For a comprehensive discussion of these problems as particularly applied to the Internet, see Mark. A. Lemley, *Antitrust and the Internet Standardization Problem*, 28 CONN. L. REV. 1041 (1996).

47. From the author's professional experience, most software companies spend over 50% of their budgets on marketing.

48. This happens frequently with software programs as well. The current browser wars between Netscape and Microsoft recently escalated when Netscape followed Microsoft's lead and decided to offer their browser for free, and also included the source code! Not even Microsoft is so bold. *See Netscape Press Release, Netscape Announces Plans to Make Next-Generation Communicator Source Code Available Free On The Internet* (visited Jan. 31, 1998) <<http://www.netscape.com/newsref/pr/newsrelease558.html>>.

49. It is estimated that Microsoft's operating systems are installed on as many as 95% of all computers. *See Will Rodger, Microsoft Hands Over Documents*, INTER@CTIVE WEEK (Jan. 29, 1999) available at

<<http://www.zdnet.com/pcweek/stories/news/0,4153,1013773,00.html>>.

applications and transmit data over networks.⁵⁰ A developer can write a program once for Microsoft's operating systems that will work for almost all existing users.⁵¹ An end user can buy one operating system from Microsoft and then easily add additional programs she knows will be compatible.

Herein lies the crux of the issue. If everyone could freely utilize the dominant market standard, then ownership of a standard is moot. Competitors would simply copy the standard and sell products based on that standard. Such is the case with TCP/IP, the standard communications protocol for the Internet.⁵² It belongs to no one – that is, it is in the public domain.

But the developer of the standard must get something for his creative efforts in developing a standard. The same factors that drive the software industry towards standards may also drive the industry towards monopoly of the industry.⁵³ This is to say, there is probably going to be a natural monopolist because usually a standard is an invention of a particular company. To solve this very problem in the past, industry competitors once cross-licensed their patent portfolios.⁵⁴ These pacts became an efficient way to avoid extensive licensing negotiations for technologies that were standards in an industry.⁵⁵ This is not a common practice today however, so new solutions must be developed.

Once companies develop software for commercial consumption, it is licensed, not sold.⁵⁶ Customers almost universally encounter the End User License Agreement (“EULA”) when purchasing software.⁵⁷ Software licenses often cover all forms of intellectual property.⁵⁸ Development tool agreements are often more complicated than other standard software agreements⁵⁹ and usually grant the user rights to

50. For example, the software on the Internet speaks on one protocol, TCP/IP. See DOUGLAS E. COMER, *THE INTERNET BOOK* 86 (1995).

51. See *id.*

52. See *id.*

53. See Lemley, *supra* note 46, at 1047-50.

54. See CHARLES H. FERGUSON & CHARLES R. MORRIS, *COMPUTER WARS: HOW THE WEST CAN WIN IN A POST-IBM WORLD* 154-57 (1993).

55. See *id.*

56. See Robert W. Gomulkiewicz & Mary L. Williamson, *A Brief Defense of Mass Market Software License Agreements*, 22 *RUTGERS COMPUTER & TECH L.J.* 335, 338-41 (1996).

57. See *id.* for a comprehensive discussion of EULAs.

58. See *infra* Part III, for a discussion of software intellectual property rights.

59. See Darren J. Carroll, *When More is Less: Controlling the Market for Computer Software Enhancements*, 43 *SYRACUSE L. REV.* 1321, 1345 (1992); Gomulkiewicz & Williamson, *supra* note 56, at 354.

copy and distribute derivative works developed from sample source code and certain binary files.⁶⁰ A software license may include a license to the copyrights, patents, and trademarks contained within the software.⁶¹ These intellectual property laws are triggered when an infringement suit is brought.

III. CURRENT INTELLECTUAL PROPERTY PROTECTION FOR SOFTWARE — THE RIGHT TO KEEP WHAT YOU OWN

A. Patent Protection for Software

Patents are an exclusively federal form of intellectual property protection.⁶² Even though a patent is at its very core a monopolization of an invention, patents are allowed to exist for a greater good. The framers of the Constitution believed that society has a larger interest in allowing these monopolies to exist because of the resulting improvements to society as a whole. Without patent protection, inventors might be afraid to create or to expose their inventions because others would use their inventions without compensating the creator. The constitutional protection provides an incentive for inventors to share their works with the world.

At one time, software was not patentable. Although this is no longer the case,⁶³ the United States Patent and Trademark Office (“PTO”) formerly viewed software as comprised of mathematical equations and algorithms,⁶⁴ and equated it to mere exploitation of facts in existence rather than discoveries of new processes.⁶⁵ After a

60. See, e.g., EULA for Microsoft Visual Studio; Gomulkiewicz, *supra* note 56, at 354.

61. See *infra* Part III, on the development of intellectual property rights in software.

62. U.S. CONST., art. I, § 8, cl. 8 states: “The Congress shall have Power . . . to promote the Progress of Science and useful Arts, by securing for limited Times to Authors and Inventors the exclusive Right to their respective Writings and Discoveries”

63. In the well-known case of *Diamond v. Diehr*, 450 U.S. 175 (1981), the U.S. Supreme Court ruled that software does not constitute an inherently unpatentable subject matter. The Supreme Court held that so long as the mathematical algorithm inherent in the software is part of a larger process or apparatus that is patentable, the entire process may be granted patent protection, despite the presence of software. But the first widely recognized pure software patent came with *In re Pardo*, 684 F.2d 912 (C.C.P.A. 1982), awarded after twelve years of patent prosecution.

64. Software actually is comprised of mathematics on a low level. Computer programming consists mainly of operations on data; moving a series of 0s and 1s around to create events. This is done on either a high or low level; if a programmer uses a language such as Assembly, he or she is acutely aware of the mathematical operations on the data. But a high level language such as Java rarely if ever operates on the core building blocks (that is, 0s and 1s) of a program.

65. “Whoever invents or discovers any new and useful process, machine, manufacture, or

seminal Supreme Court case, *Diamond v. Diehr*,⁶⁶ the PTO began to accept patent applications for software. It is clear that software is now protectible under the patent laws.⁶⁷

B. *The Scope of Copyright Protection for Software*

Copyright protection is likewise based upon authority of the Constitution.⁶⁸ Pursuant to this authority, Congress passed the 1976 Copyright Act, which provides that all "original works of authorship fixed in any tangible medium of expression"⁶⁹ are protected against unauthorized copying. The primary purpose of the Copyright Act is to encourage authors to create and make available their works to the public. Like patents, the grant of a copyright yields the side effect of a constitutionally granted monopoly in these fixed works.⁷⁰

In 1980, the Copyright Act was amended to explicitly include software.⁷¹ In the case which gave birth to this field of jurisprudence, *Apple Computer, Inc. v. Franklin Computer Corp.*,⁷² the court ruled that a computer program is a form of literary work under 17 U.S.C. § 102 and thus receives protection from unauthorized copying. It is now commonly accepted that copyright protection extends to software.⁷³

But copyright law does not protect an idea – only its expression.⁷⁴ Anyone who independently develops the same

composition of matter, or any new and useful improvement thereof, may obtain a patent therefor . . ." 35 U.S.C. § 101 (1994).

66. *Diamond v. Diehr*, 450 U.S. 175 (1981).

67. *See, e.g.*, In re Iwahashi, 888 F.2d 1370, 1375 (Fed. Cir. 1989) (holding patent may be issued that is based on inter-operability algorithm); *Arrythmia Research Tech., Inc. v. Corazonix Corp.*, 958 F.2d 1053, 1058 (Fed. Cir. 1992) (holding claims where algorithm is based on process may be granted); In re Alappat, 33 F.3d 1526, 1541-1542 (Fed. Cir. 1994) (holding anti-aliasing technique algorithm is not included in mathematical algorithm bar to patent grant); In re Lowry, 32 F.3d 1579, 1583 (Fed. Cir. 1994) (holding data structure algorithm for computer storage memory is patentable); In re Beauregard, 53 F.3d 1583, 1584 (Fed. Cir. 1995) (holding computer programs in a tangible medium are patentable).

68. *See* U.S. CONST., art. I, § 8, cl. 8.

69. 17 U.S.C. § 102 (1994).

70. *See* *Mazer v. Stein*, 347 U.S. 201, 219 (1954).

71. Software is protected as a literary work, and is defined in the Copyright Act. "A computer program is a set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result." 17 U.S.C. § 101 (1994).

72. *Apple Computer, Inc. v. Franklin Computer Corp.*, 714 F.2d 1240 (3d Cir. 1983).

73. The Copyright Act was amended in 1980 to recognize the copyrightability in computer programs. *See* 17 U.S.C. §§ 101, 117 (1994). *See also* *Apple Computer, Inc. v. Franklin Computer, Inc.*, 714 F.2d 1240, 1248 (3d Cir. 1983) (upholding validity of software copyright).

74. *See* 17 U.S.C. § 102(b) (1994).

expression in a copyrighted work may freely use it: the copyright prevents only illicit *copying of the tangible expression*.⁷⁵ Literal copying of software obviously infringes a copyright, and courts have had no issue with finding liability in such cases.⁷⁶ But cases involving non-literal copying⁷⁷ are not so clear and courts have struggled: several evolutionary processes have emerged in modern software copyright jurisprudence.⁷⁸

C. The Use of Trademark Licenses as a Pretext for Technology Licenses

Some companies use an additional weapon beyond patent or copyright protection to protect their software: they use trademark licenses.⁷⁹ Trademark law gives protection to the commercial marks of a company. It was originally created by Congress to protect consumers from confusion. However, with the creation of the recent Federal Trademark Dilution Act of 1995,⁸⁰ trademark law also seeks to protect companies' interests as well.

For example, Microsoft does not require any special license to use the libraries found in its Windows 95 software, but in order for a software reseller to advertise that the program works with Windows 95 and to display the Microsoft logo and trademarks on its products, it

75. See *Mazer*, 347 U.S. at 217-18.

76. Literal copying means physical copying of the software code. One of the earliest cases was *Williams Elecs., Inc. v. Artic Int'l, Inc.*, 685 F.2d 870 (3d Cir. 1982).

77. Non-literal copying means copying the aspects of the computer program that are not reduced to written code. See *Computer Assocs. v. Altai*, 982 F.2d 693, 696 (2d Cir. 1992).

78. See *Apple Computer, Inc. v. Franklin Computer, Inc.*, 714 F.2d 1240, 1248 (3d Cir. 1983) (upholding validity of software copyright); *Whelan Assocs., Inc. v. Jaslow Dental Lab., Inc.*, 797 F.2d 1222, 1237-1238 (3d Cir. 1986) (holding copyright protection for software extends beyond literal code); *Computer Assocs. v. Altai*, 982 F.2d 693 706-11(2d Cir. 1992) (adopting the modern abstraction-filtration comparison doctrine.) The modernly accepted abstraction-filtration-comparison doctrine makes the following steps in determining whether a work is infringing: (1) abstraction, where a court "dissect[s] the allegedly copied program's structure and isolate each level of abstraction contained within it; (2) filtration, where separation of protectable expression from non-protectable material is accomplished; and (3) comparison, which occurs after a court has sifted out all elements of the allegedly infringed program which are "ideas," are dictated by efficiency or external factors, taken from the public domain, etc. *Id.* at 706-10. The idea-expression distinction is critical in the area of computer software; there is often merger in the expression of software programs, and until recently, patents for software technology have been difficult to obtain.

79. See, e.g., *Silicon Graphic's Web Site*, (visited June 11, 1998) <<http://www.sgi.com>>; *JavaSoft's Web Site* (a division of Sun Microsystems) (visited June 13, 1998) at <<http://java.sun.com>> (companies explain licensing policies of their trademarks).

80. Federal Trademark Dilution Act of 1995, Pub. L. No. 104-98, 109 Stat.985 (codified at 15 U.S.C.A. § 1125, 1127 (1998)).

must obtain permission to do so.⁸¹ Without the Microsoft Windows 95 trademark, the software publisher can not claim that its product works with the Windows 95 operating system. Even though the trademark provides no protection for Microsoft's software, sales of a third party's compatible product are commercially infeasible without use of the Windows 95 compatibility logo. Sun Microsystems achieves the same end by restricting any use of its Java logo.⁸²

IV. ANTITRUST THREAT TO SOFTWARE STANDARDIZATION

A. *Modern Jurisprudence and its Relationship to Standardized Development Tools*

Perhaps the most worrisome problem for a creator of standardized software technology in protecting its standardized intellectual property is compliance with the antitrust laws. It is assumed that the creator will have some measure of intellectual property rights, but the enforcement of such rights can bring liability under another family of law, antitrust. Antitrust laws prevent various types of conduct, which include agreeing on the price competitors will charge,⁸³ predatory practices designed to exclude competitors from the market,⁸⁴ and certain limits on the behavior of firms with market power.⁸⁵ Refusal to license an industry standard to a party may harm competition, thus bringing antitrust law into play.

1. Section 2 Liability for Monopoly Conduct

Congress passed Section 2 of the 1890 Sherman Act⁸⁶ [hereinafter Section 2] in response to the public's outrage at giant "trusts." It condemns "every person who shall monopolize."⁸⁷ Congress chiefly intended the Sherman Act to protect consumers from monopolists and cartels.⁸⁸ Monopolization may be proven a

81. See (visited Apr. 1, 1999) <<http://www.microsoft.com/permission/t-mark/logos.htm>>.

82. See *supra* note 29, and accompanying text.

83. See 15 U.S.C. § 1 (1994).

84. See *Brooke Group v. Brown & Williamson Tobacco*, 509 U.S. 209, 224 (1993) (holding no predatory price discrimination under Robinson-Patman Act by defendant cigarette manufacturer).

85. See 15 U.S.C. § 2 (1994).

86. See *id.* at §§ 1-2.

87. *Id.*

88. The Supreme Court in *United States v. Grinnell Corp.*, 384 U.S. 563, 604-05 (1966) set forth the modern formulation of the monopoly offense, which contains two elements. The court enumerated them as: "1) the possession of monopoly power in the relevant market; and 2)

variety of ways, including proving another antitrust law has been violated in an effort to monopolize.⁸⁹

Further, Section 2 is judicially defined as prohibiting acquisition or the maintenance of a monopoly through anticompetitive conduct.⁹⁰ Mere appearance of monopoly by market power or market share is not enough.⁹¹

Specifically, Section 2 makes it illegal for the monopolist of a properly defined essential facility to deny access to a competitor.⁹² Under the essential facility doctrine, the owners of facilities that are essential for a company to compete must make them available to anyone, even competitors, on non-discriminatory terms.⁹³ A local electricity monopoly must share its power with others,⁹⁴ and a telephone monopoly must interconnect all long distance carriers.⁹⁵ But the essential facilities doctrine is not without criticism,⁹⁶ and the cases in which a court finds liability are few and far between.⁹⁷

An essential facility can be a natural monopoly, a subsidized facility such as an athletic stadium, or sometimes, a regulated monopoly.⁹⁸ The monopolist of a properly defined essential facility is often a natural monopolist. A natural monopoly is one that is generally characterized by extremely high start-up costs and relatively low marginal costs.⁹⁹ The end result of competition in a market characterized by a natural monopoly is a single victorious firm that drives out its competitors.¹⁰⁰ Low marginal costs give natural

the willful acquisition or maintenance of that power as distinguished from growth or development as a consequence of a superior product, business acumen, or historic accident." *Id.* at 570-71. A more recent case, *Aspen Skiing Co. v. Aspen Highlands Skiing Corp.*, 472 U.S. 585 (1985), has reaffirmed this test.

89. See generally, *Standard Oil Co. v. United States*, 221 U.S. 1 (1911) (holding anticompetitive conduct can be defined as conduct which violates antitrust laws).

90. See *Aspen Skiing Co. v. Aspen Highlands Skiing Corp.*, 472 U.S. 585 (1985); *United States v. Aluminum Co. of Am.*, 148 F.2d 416, 429-30 (2d Cir. 1945).

91. See *Aluminum Co. of Am.*, 148 F.2d at 429-30.

92. See 15 U.S.C. § 2 (1994).

93. See *Otter Tail Power Co. v. United States*, 410 U.S. 366, 378 (1973).

94. See *id.*

95. *MCI Communications Corp. v. American Tel. & Telegraph, Co.*, 708 F.2d 1081 (7th Cir. 1983).

96. See, e.g., PHILIP E. AREEDA & HERBERT HOVENKAMP, *ANTITRUST LAW* 771 (1996).

97. See Lemley, *supra* note 46, at 1084, citing *Alaska Airlines v. United Airlines*, 948 F.2d 536 (9th Cir. 1991); *Illinois ex rel. Burriss v. Panhandle E. Pipe Line Co.*, 935 F.2d 1469, 1482 (7th Cir. 1991).

98. See, e.g., *Otter Tail Power Co. v. United States*, 410 U.S. 366 (1973). See also, *Aspen Skiing Co. v. Aspen Highlands*, 472 U.S. 585 (1985).

99. See HERBERT HOVENKAMP, *FEDERAL ANTITRUST POLICY* 30 (1994).

100. See *id.*

monopoly firms an incentive to cut prices in order to expand their market share over the entire range of the market because doing so reduces their average total costs.¹⁰¹ Competition in natural monopoly markets can be somewhat wasteful. While the lure of the market may cause several companies to enter, only one will succeed and the others will struggle.

In the 1940s, the Supreme Court held that Section 2 of the Sherman Act forbids the use of a monopoly power in one market to acquire a competitive advantage in a second market.¹⁰² This is true even if there has not been an attempt to monopolize the second market. The liability comes from the “abuse” of economic power already held in the first market, or “leveraging.”¹⁰³

The leveraging theory has been applied to claims that a monopolist used restrictive agreements to obtain an unfair advantage over rivals.¹⁰⁴ However, leveraging that creates a second monopoly generally does not result in higher market prices or lower output. This economic theory has led recent courts to now reject it outright.¹⁰⁵

2. Particular Issues With Software Standardization

Antitrust law today is based on enhancing competition and not on protecting competitors. But in the software industry, like many others, the need for standards is essential.¹⁰⁶ A party holding the rights to a standard is indeed a monopolist – everyone holding a license to Java got it from Sun – but having a standard is absolutely necessary because of the unique way software works.¹⁰⁷ Software

101. *See id.*

102. *See United States v. Griffith*, 334 U.S. 100, 107-08 (1948).

103. In *United States v. Griffith*, 334 U.S. 100 (1948), a firm used dominant position in some movie theaters to obtain a competitive advantage in other locations. The Supreme Court stated that a firm monopolizes illegally when it 1) has the power to exclude competition and 2) has exercised it, or has the purpose or intent to exercise it. *Id.* at 107. Other cases have recognized that a firm is forbidden to use monopoly power in one market to acquire a competitive advantage in a second market “even if there has not been an attempt to monopolize the second market.” *Berkey Photo, Inc. v. Eastman Kodak Co.*, 603 F.2d 263, 276 (2d Cir. 1979). Liability flows from the abuse of economic power already held in the first market - not from any threat that economic power will be created in the second market. *See id.*

104. *See, e.g., Kerasotes Michigan Theaters, Inc. v. National Amusements, Inc.* 854 F.2d 135, 138 (6th Cir. 1988) (reversing dismissal of leverage claim).

105. *See Spectrum Sports v. McQuillan*, 506 U.S. 447, 453 (1993) (holding absent proof of dangerous probability of monopolizing relevant market, no liability for attempted monopolization under Section 2 of the Sherman Act).

106. *See Lemley, supra note 46*, at 1072.

107. *See id.*

must be with a certain number of users, or else it becomes commercially infeasible. End-users of software need to have programs that work on their computers with few to no custom alterations; this is only possible if standards are adopted and adhered to by programmers with languages, operating systems, and programming methods. Because of the factors that drive developers and end users to standardization, there will always be some dominant actors in the markets and a monopoly holder if the standard is proprietary.¹⁰⁸

Application of Section 2 to a standardized software market with a dominant standard does not have any prior precedent, with the possible exception of the present trial in the case of *U.S. Department of Justice v. Microsoft*.¹⁰⁹ Microsoft's best legal argument is that its current monopoly on the market was not caused by its monopolistic behavior, but rather, that it gained its position through good competitive practices and ingenuity.¹¹⁰

The tendency to standardize in the software industry may create a natural monopoly with control over an essential facility.¹¹¹ The market for programming languages or libraries generally has and will converge on standards over time.¹¹² Standards in the software development market tend to live for long periods, due to developer's learning patterns, market inertia, and backwards compatibility needs for prior products.¹¹³ And once a particular company has a monopoly, the monopoly is likely to be long lasting as well. A company in such a position will not only keep market power with its current standard, but it will have a far easier time obtaining support from loyal software developers with its future products. Software engineers will already understand how the company's other products work, and will be more comfortable with the company's new products than with ones from a third party.

A current example of leveraging can be seen in the *U.S. Department of Justice v. Microsoft* matter, currently in trial. What is

108. *See id.*

109. *See* Michael Moeller, *Microsoft, DOJ Hit the Courts*, PC WEEK ONLINE, (visited Oct. 19, 1998) <<http://www.zdnet.com/pcweek/stories/news/0,4153,361415,00.html>>.

110. *See* *United States v. E. I. du Pont de Nemours & Co.*, 351 U.S. 377 (1955).

111. *See* Lemley, *supra* note 46, at 1056.

112. *See* David S. Evans & Bernard Reddy, *Some Economic Aspects of Standards in Network Industries and their Relevance to Antitrust and Intellectual Property Law*, 1 INTELLECTUAL PROPERTY ANTITRUST, 1996 at 177, 189 (PLI Patents, Copyrights, Trademarks, and Literary Property Course Handbook Series, No. G-448, 1996).

113. *See* Lemley, *supra* note 46, at 1056.

at issue is Microsoft's ability to require that its browser be installed on the desktop of every version of Windows 95. Because Microsoft has a monopoly in the operating system market,¹¹⁴ the argument is that Microsoft is using its monopoly in one market to leverage its monopoly of the other market. It remains to be seen what control the government is going to ask for in future releases of Microsoft's market. Many academics and industry commentators believe the government has gone far enough.¹¹⁵

V. THE COUSINS OF ANTITRUST: DEFENSES AVAILABLE TO INTELLECTUAL PROPERTY INFRINGEMENT

A user of a proprietary standard may have defenses if it uses a patented software technology although it has no contractual or other right to otherwise do so. Generally, the patent owner has complete control over issuing licenses for his patents.¹¹⁶ Only one case has ever held otherwise: *Allied Research Prod, Inc. v. Heatbath Corp.*¹¹⁷ where the court held that a patent owner could not discriminatorily refuse a license to another company solely because of a "personal dispute" between two individuals. This case has never been followed by any court.

The generally followed law states instead that patent law requires no compulsory licensing.¹¹⁸ Another leading case is *Minnesota Mining and Manufacturing Co. v. Research Medical, Inc.*,¹¹⁹ which held that a patentee's refusal to grant a license to defendant, after granting a license to defendant's competitor, does not constitute patent misuse.¹²⁰

114. It is estimated that Microsoft's operating systems are installed on as many as 95% of all computers. Will Rodger, *Microsoft Hands Over Documents*, INTER@CTIVE WEEK (Jan. 29, 1999) available at <<http://www.zdnet.com/pcweek/stories/news/0,4153,1013773,00.html>>. Although there are no set numbers as to what constitutes a monopoly, this would certainly be considered a monopoly by any court. See *United States v. Aluminum Co. of Am.*, 148 F.2d 416, 427 (2d Cir. 1945).

115. See Jason Pontin, *Out of Step: The Suits Against Microsoft are in Conflict with Antitrust Policy*, RED HERRING, Jul. 1998, at 144.

116. See DONALD S. CHISUM, CHISUM ON PATENTS, § 19.04[3][f] (1997). See also, *Minnesota Mining & Mfg Co. v. Research Med., Inc.*, 679 F. Supp. 1037 1065-66(D. Utah 1988); *Cygnus Therapeutic Sys. v. ALZA Corp.*, 92 F.2d 1153, 1160 (Fed. Cir. 1996).

117. *Allied Research Prods., Inc. v. Heatbath Corp.*, 300 F. Supp. 656 (N.D. Ill. 1969).

118. *Dawson Chem. Co. v. Rohm & Haas Co.*, 448 U.S. 176, 215 (1980).

119. *Minnesota Mining and Mfg. Co. v. Research Med., Inc.*, 679 F. Supp. 1037 (D. Utah 1988).

120. See CHISUM, *supra* note 116, at § 19.04[3][f], at n.55.1, *citing* *Data Gen. Corp v. Grumman Sys. Support Corp.*, 36 F.3d 1147, 1186 (1st Cir. 1994); CHISUM, *supra* note 116, at § 19.04[3][f], at n.54, *citing* *Tricom Inc. v. Electronic Data Sys. Corp.*, 902 F. Supp. 741, 743

However, certain conduct by a patentee can prohibit the patentee from upholding or enforcing his patent rights. Some of these defenses might be available to someone who is defending against claims of unauthorized use of a standardized product. In the patent arena, patent misuse, implied license by equitable estoppel, and equitable estoppel might be available to a defendant. Copyright has its counterpart in the copyright misuse defense, which is drawn chiefly from patent misuse.

A. Patent Misuse

The courts are quite wary of a patentee abusing his patent grant. As a response to this fear, patent misuse as an equitable defense to an infringement action, is made available where a patent owner has exploited his patent in an improper manner either by violating the antitrust laws or by extending the patent beyond its lawful scope.¹²¹ If misuse is found, the courts will not provide a remedy for infringement or breach of a license agreement – even if the infringer is not harmed by the abusive practice.¹²²

Morton Salt Co. v. G.S. Suppiger Co.,¹²³ provides a classic example of patent misuse. Here, the defendant attempted to force purchasing of salt with the licensing of its patented salt dispensing machines. The Court held that Morton's creation of a tying arrangement was contrary to patent public policy.¹²⁴

(E.D. Mich. 1995) (holding no compulsory licensing for software).

121. Early lower court decisions refused to recognize as a defense to infringement an allegation that the plaintiff was engaged in conduct in violation of the antitrust laws. *See, e.g., Strait v. National Harrow*, 51 F. 819, 820 (N.D.N.Y. 1892). However, the Supreme Court recognized the patent misuse defense in connection with tying cases. *See, e.g. Mercoird Corp. v. Mid-Continent Inv. Co.*, 320 U.S. 661, 664-65 (1943); *Dawson Chem. Co. v. Rohm & Haas*, 448 U.S. 176, 200-01 (1980).

122. *See* CHISUM, *supra* note 116, at § 19.04, (discussing patent misuse). *See also, e.g., United States Gypsum Co. v. National Gypsum Co.*, 352 U.S. 457, 465 (1957); *Morton Salt Co. v. G.S. Suppiger Co.*, 314 U.S. 488, 492-94 (1942); *National Patent Dev. Corp. v. T.J. Smith & Nephew, Ltd.*, 877 F. 2d 1003, 1008 (D.C. Cir. 1989); *Riker Lab., Inc. v. Gist Brocades, N.V.*, 636 F.2d 772, 777 (D.C. Cir. 1980) (holding misuse conduct must relate to patent being litigated, but no harm to patentee is necessary).

123. *Morton Salt Co. v. G.S. Suppiger Co.*, 314 U.S. 488 (1942).

124. *See Philip Abromats, Comment: Copyright Misuse and Anticompetitive Software Licensing Restrictions: Lasercomb America, Inc. v. Reynolds* 52 U. PITT. L. REV. 629, 637 (1991). *Morton Salt* did not involve the analysis of antitrust laws. For a salt case that did, see *International Salt Co. v. United States*, 332 U.S. 392 (1947). International owned patents for two salt-processing machines, and had the distinction of being the largest producer of salt. The court ruled that it deserved a limited monopoly of what it invented, but that it could not monopolize the sales of salt when the effect was to foreclose competitors from the salt market. *See id* at 395-96.

1. Patent Use Reform Statute

Congress wanted to expressly protect patentees from any obligation to license their patents. The "Patent Misuse Reform Act" was passed in 1988,¹²⁵ and lists acts by patent owners that shall not be deemed misuse. 35 U.S.C. 271(d) reads:

No patent owner otherwise entitled to relief for infringement or contributory infringement shall be . . . deemed guilty of misuse or illegal extension of the patent right by reason of having done one or more of the following:

(4) refused to license or use any rights to the patent. . .

Although Section 271(d)(4) was said to be an attempt to codify existing patent law,¹²⁶ at least one court has stated that a refusal to license in itself is not a form of patent misuse, but a refusal that violates the antitrust laws or otherwise seeks to extend the power of the patent beyond its scope is a form of patent misuse.¹²⁷ This law is quite helpful when the licensing of a patent not commonly used as a standard is called into question. But, for the standardized software creator, Section 271(d)(4) helps little when a standard may constitute an "essential facility,"¹²⁸ and thus is deemed a natural monopoly, which triggers liability under antitrust law.¹²⁹

The doctrine of patent misuse is a doctrine closely intertwined with antitrust law.¹³⁰ The key question in determining if there has been an antitrust violation is whether the patentee has unlawfully attempted to misuse or increase its market power in the relevant markets; any such violation of the antitrust laws will automatically constitute patent misuse.¹³¹ But even conduct that does not quite constitute an antitrust violation can still be considered patent misuse.¹³²

125. Patent Misuse Reform Act, 35 U.S.C. § 271(d) (1994). See also CHISUM, *supra* note 116, at § 19.04[1](f).

126. See *supra* Part V.A. See also, Remarks of Representative Kastenmeier, 134 CONG. REC., H10648, H10648 (Oct. 20, 1988).

127. See *Braun v. Abbott*, 124 F.3d 1419, 1426 (Fed. Cir. 1997). See also 35 U.S.C. § 271(d)(5) (1994) (tying allowed unless there is market power in the relevant market for the patent or patented product, the standard for tying under *Jefferson Parish Hosp. Dist. No. 2 v. Hyde*, 466 U.S. 2 (1984)).

128. See *supra* Part IV.A.2.b.

129. See *infra* Part V.A.1.c. for discussion of the interplay between antitrust and patent misuse laws.

130. See CHISUM, *supra* note 116, at § 19.04[2].

131. See *id.*

132. Chisum states that the "three classic acts of misuse are (1) requiring the purchase of

2. *Braun v. Abbott*: The Law of Patent Misuse in the Federal Circuit

In *Braun v. Abbott*, the court outlined the patent misuse defense.¹³³ The court stated that the patent misuse doctrine is a method of limiting abuse of patent rights *separate from the antitrust laws*.¹³⁴ That is, no separate antitrust violation is needed; a lesser test of liability will suffice. The Federal Circuit held that the key inquiry in utilizing a patent misuse defense is “whether, by imposing the condition, the patentee has impermissibly broadened the ‘physical or temporal scope’ of the patent grant with anticompetitive effect.”¹³⁵ This could entail using a patent that enjoys market power in the relevant market,¹³⁶ or restraining competition in an unpatented product.¹³⁷ However, field of use restrictions have generally been upheld¹³⁸ and any anticompetitive effects they may cause are reviewed in accordance with the rule of reason.¹³⁹ Again, this application of the laws may not help the standard owner in its prosecution, because of the potential crippling effects on a party without access to a standardized patent.

Braun was not a Section 271(d)(4) case, rather, it involved post-sale restrictions on goods. The District Court had allowed an instruction to the jury stating that a patentee “is not allowed to place restrictions on customers that prohibit resale of the patented product, or allow the customer to resell the patented product only in connection with certain products.”¹⁴⁰ Braun objected to this instruction, stating that it created *per se* liability for any condition that Braun placed on its sales.¹⁴¹ The Federal Circuit agreed, affirming its decision in *Mallinckrodt, Inc. v. Medipart, Inc.*,¹⁴² where the court looked very closely at the legality of restrictions placed upon the post-sale use of patented goods. *Mallinckrodt* held that the unconditional

unpatented goods for use with patented apparatus or processes, (2) prohibiting production or sale of competing goods, and (3) conditioning the granting of a license under one patent upon the acceptance of another and different license.” *Id.* at § 19.04[3].

133. *Braun v. Abbott*, 124 F.3d 1419, 1426 (Fed. Cir. 1997).

134. *See id.*

135. *Id.* citing *Windsurfing Int'l, Inc. v. AMF, Inc.*, 782 F.2d 995, 1001-02 (Fed. Cir. 1986); *see also* *Mallinckrodt Inc. v. Medipart Inc.*, 976 F. 2d 700, 704 (Fed. Cir. 1992).

136. *See* 35 U.S.C. § 271(d)(5) (1994).

137. *Braun*, 124 F.3d at 1426.

138. *See* *General Talking Pictures Corp. v. Western Elec. Co.*, 305 U.S. 124, 127 (1938).

139. *See* *Mallinckrodt*, 976 F.2d at 708.

140. *Id.*

141. *Id.*

142. *Mallinckrodt Inc. v. Medipart Inc.*, 976 F.2d 700 (Fed. Cir. 1992).

sale of a patented device will generally exhaust the patentee's right to control the purchaser's use of the device.¹⁴³ This is known as the exhaustion doctrine.

Most importantly for software producers, however, is that the exhaustion doctrine does not apply to an *expressly conditional* sale or license¹⁴⁴ — in software law, this is an EULA.¹⁴⁵ The *Braun* court explained that it is more reasonable to infer that the parties negotiated a price that reflects only the value of the “use” rights conferred by the patentee.¹⁴⁶ As a result, express conditions accompanying the sale or license of a patented product are generally upheld.¹⁴⁷ A strict reading of this would imply that if the owner of a patent, used as a standard in an industry, refused to license a technology, then it would be upheld by the terms of the EULA and the end-user would have no recourse. However, the court noted that such express conditions are contractual in nature and are subject to antitrust, patent, contract, and any other applicable law, as well as equitable considerations such as patent misuse. Conditions that violate another law or equitable consideration are unenforceable, but violation of valid conditions entitles the patentee to a remedy for patent infringement or breach of contract.¹⁴⁸

3. Interplay Between Antitrust and Patent Misuse

But what if the patentee's refusal to license deprives a competitor of something it needs to compete in the relevant market? This action appears to be an antitrust violation despite Congress' mandate that refusal to license is not patent misuse, thereby creating the complex relationship between the misuse doctrine and the federal antitrust laws.¹⁴⁹ As the court outlined in *Braun*, use of a patent to violate the antitrust laws will constitute misuse.¹⁵⁰ However, conduct that falls short of an antitrust violation may still constitute misuse.¹⁵¹

143. See *id.*, at 706; *Braun*, 124 F.3d at 1426.

144. See *Braun*, 124 F.3d at 1426.

145. See *supra* note 57 and accompanying text.

146. But see *infra* notes 160-162 and accompanying text for a discussion of *Wang Laboratories, Inc. v. Mitsubishi Electronics*, 103 F.3d 1571 (Fed. Cir. 1997), where the court found that Wang did get their asking price: standardization of their product.

147. See *Braun*, 124 F.3d at 1426; *Mallinckrodt*, 976 F.2d at 708; cf. *General Talking Pictures Corp. v. Western Elec. Co.*, 305 U.S. 124, 127 (1938) (holding that under the facts of the case, a restrictive license is legal).

148. See *Braun*, 124 F.3d, at 1426; *Mallinckrodt*, 976 F.2d at 703.

149. See CHISUM, *supra* note 116, at § 19.04[2].

150. See *Braun*, 124 F.3d at 1426.

151. See *id.*

This would seem to indicate that antitrust has a higher standard.

In the antitrust laws, unilateral refusals to deal are illegal when they constitute monopolization or an attempt to monopolize the market.¹⁵² The plaintiff must show that the defendant is a monopolist and that the refusal to deal is an anti-competitive exclusionary practice,¹⁵³ since a monopolist may refuse to deal provided the refusal is not anti-competitive.¹⁵⁴ If this is the antitrust standard,¹⁵⁵ then antitrust may provide grounds for a plaintiff to assert rights to a development technology. This would circumvent the patent misuse statute's directive and create a loophole for liability. If it is assumed that the product in question is a standard for the type of application that the plaintiff wants to create, unavailability of the standard will have devastating anti-competitive effects.¹⁵⁶

A patent owner regains the right to enforce a patent when improper patent misuse practice has been fully abandoned, and the consequences of the act have fully been dissipated.¹⁵⁷ As such, no award of damages is available to the accused infringer — only the right to use the patent. But if misuse were found to be that of leveraging or monopoly conduct, when would the company be able to re-assert its patent rights? When the standards have changed? One wonders what would happen to the user's existing products that have incorporated the standardized technologies?

152. See *Lorain Journal Co. v. United States*, 342 U.S. 143, 149-50 (1951). Defendant newspaper's intent in refusing to deal with those who had bought advertising on the radio was found to reserve an advertising monopoly to itself. See also *Aspen Skiing Co. v. Aspen Highlands*, 472 U.S. 585, 604-04 (1985), where the Supreme Court reaffirmed *Lorain Journal*, and held that a refusal to deal by a monopolist, together with a showing of anticompetitive motive or intent violated section 2 of the Sherman Act, 15 U.S.C. § 2 (1994).

153. See *Aspen Skiing Co. v. Aspen Highlands*, 472 U.S. 585, 596, 604 (finding jury instructions proper).

154. Such unilateral refusals to deal are illegal under section 2 of the Sherman Act, 15 U.S.C. § 2 (1994) and under the "unfair competition" provision of section 5 of the Federal Trade Commission Act, 15 U.S.C. § 20 (1994).

155. See *Braun*, 124 F.3d at 1426.

156. An example of this might be found in a dispute over the GIF graphic file exchange format. Invented by CompuServe in 1987, CompuServe released the standard for everyone to develop and incorporate in their products. A patent had been obtained in 1986 for the compression algorithm used in the standard, and the owner of the patent, Unisys, failed to inform anyone of its existence. The file format gained a huge acceptance and a large market share, and was wide spread in graphics programs. In 1994 Unisys began to assert its patent rights, after having remained completely silent while the standard ensued. The case has never been litigated. See Jimmy Guterman, *Are your GIFs legal?* CHICAGO TRIBUNE, June 18, 1998 at 1.

157. See CHISUM, *supra* note 116, at § 19.04[3] (1997); *Morton Salt Co. v. G.S. Suppiger Co.*, 314 U.S. 488, 492-93 (1942); *B.B. Chem. Co. v. Ellis*, 314 U.S. 495-98 (1942).

B. Implied License as Defense

Another theory available to one refused a patent license for a development tool is that of implied license, also known as implied license by equitable estoppel.¹⁵⁸ The doctrine of implied license by equitable estoppel is related to that of equitable estoppel, discussed *infra* in Part V.C. The primary difference is that in implied license cases courts look for an affirmative grant of consent or permission to make, use, or sell the embodiment of a patented technology, while equitable estoppel cases focus on examining “misleading” conduct that suggested the patentee would not enforce patent rights.¹⁵⁹ Judicially implied licenses are rare under any doctrine, but courts will grant them if a specific course of conduct is exhibited by the patentee.¹⁶⁰

In *Wang v. Mitsubishi*, the Federal Circuit upheld a finding of noninfringement by implied license based on the lower court’s examination of equitable estoppel principles. Wang set out to make its memory module the standard in the industry, and successfully convinced the relevant national standards committee to adopt the technology. Rather than produce the chips themselves, Wang encouraged manufacturers to begin producing and marketing the modules. Several manufacturers, including Mitsubishi, cooperated with Wang to begin producing and marketing the modules. Meanwhile, without disclosing to anyone it was pursuing a patent application, Wang applied for and was awarded several patents on the module design. When Wang sued Mitsubishi for infringement of the patent, Mitsubishi raised the defense of implied license, and the court allowed the defense.¹⁶¹

The court held that an implied license merely signifies a patentee’s waiver of the statutory right to exclude others from making, using, or selling the patented invention.¹⁶² It explained that

158. This is not the same defense as equitable estoppel, discussed *infra* Section V.C.

159. See *Wang Lab. Inc. v. Mitsubishi Elecs.*, 103 F.3d 1571, 1581 (Fed. Cir. 1997).

160. See *id.* at 1580.

161. See *id.*

162. See *id.* at 1579, citing *Spindelfabrik Suessen-Schurr Stahlecker & Grill GmbH v. Schubert & Salzer Maschinenfabrik Akiengesellschaft*, 829 F.2d 1075, 1081 (Fed. Cir. 1987). See also *DeForest Radio Tel. Co. v. United States*, 273 U.S. 236, 241 (1927), where the Supreme Court stated:

No formal granting of a license is necessary in order to give it effect. Any language used by the owner of the patent, or any conduct on his part exhibited to another from which that other may properly infer that the owner consents to his use of the patent in making or using it, or selling it, upon which the other acts, constitutes a license and a defense to an action for a tort.

an implied license could not arise out of the unilateral expectations or even reasonable hopes of one party; one must have been led to take action by the conduct of the other party.¹⁶³ “Accused infringer’s acts of going forward to the point where it became committed to utilizing an inventor’s patented inventions in its new lines cannot in itself create [an implied license of equitable] estoppel.”¹⁶⁴ Implied licenses arise by acquiescence, by conduct, by equitable estoppel (estoppel in pais), or by legal estoppel.¹⁶⁵ Therefore, the courts will look at the entire course of conduct for facts giving rise to estoppel, and at how the conduct of the patentee caused the other to act,¹⁶⁶ in order to conclude whether or not an implied license exists.¹⁶⁷

However, neither the Federal Circuit nor the Supreme Court requires a formal finding of equitable estoppel for a legal conclusion of implied license.¹⁶⁸ The primary difference is that implied license cases look for an affirmative grant of consent or permission to make, use, or sell the patented technology,¹⁶⁹ where equitable estoppel cases focus on “misleading” conduct suggesting that the patentee will not enforce patent rights.¹⁷⁰

In *Wang*, the court took particular note that 1) the parties had been in meetings for over six years; 2) Wang coaxed Mitsubishi into the SIMM market; 3) Wang provided designs, suggestions, and samples to Mitsubishi; and 4) Wang purchased the resulting SIMMs from Mitsubishi, years before accusing Mitsubishi of infringement.¹⁷¹ But *Wang* has facts on the extreme side of the scenario: long duration, and intimacy of the parties. While these circumstances may be of help to parties who are familiar with each other, it is probably doubtful that an implied license could ever be found among

See also, *RSA Data Sec., Inc. v. Cylink Corp.*, No. 96-20094 SW, 1996 WL 266201, (N.D. Cal May 17, 1996).

163. *See Stickle v. Heublein, Inc.*, 716 F.2d 1550, 1559 (Fed. Cir. 1983). The court affirmed that no implied license exists unless there is a nexus between the patentee’s conduct and the infringing actions.

164. *Id.*

165. *See Wang Lab., Inc. v. Mitsubishi Elecs.*, 103 F.3d 1571, 1579 (Fed. Cir. 1997); *AMP, Inc. v. United States*, 389 F.2d 448, 452 nn. 4-5.

166. *See Bandag, Inc. v. Al Bolser’s Tire Stores, Inc.*, 750 F.2d 903, 925-6 (Fed. Cir. 1984); *Stickle*, 716 F.2d at 1559.

167. *See Wang*, 103 F.3d, at 1579; *AMP*, 389 F.2d at 452.

168. *See AMP*, 389 F.2d at 453-54. An effect otherwise would remove all distinction between the doctrines.

169. *See Wang*, 103 F.3d at 1580; *Stickle*, 716 F.2d at 1559.

170. *See Wang*, 103 F.3d at 1580; *A.C. Aukerman Co. v. R.L. Chaides Constr. Co.*, 960 F.2d 1020, 1041-44 (Fed. Cir. 1992).

171. *See Wang* 103 F.3d at 1582.

anonymous end users of a standard product.

Perhaps the most interesting and relevant part of *Wang* for development tool creators was the court's finding that Wang received the compensation it desired for "licensing" its technology. Wang's patents became industry standard, and remain the standard today.¹⁷² Although no financial remuneration changed hands, Wang got what it wanted with the help of others. The court held that having a standard product is good enough for Wang.¹⁷³ This may be what most standardized software developers want.¹⁷⁴ The granting of a standard may take away the patentee's right to enforce the patent on the particular standard, but it does give the patentee name recognition, a broad user base, and general industry acceptance of the company's technologies.

C. Equitable Estoppel

Another defense available in patent infringement is equitable estoppel. If a software company places its patented technology into the marketplace for all to use, actively promotes it to be used as the standard in the industry, and assures software and hardware developers that they can use it freely, a user of patented technology may assert the defense of equitable estoppel. Elements of an equitable estoppel defense are:

(1) the patentee, through misleading conduct, leads the alleged infringer to reasonably infer that he does not intend to enforce the patent against the alleged infringer. The conduct may include specific statements, action, inaction, or silence where there is a duty to speak;

(2) the alleged infringer relies on the patentee's conduct; and

(3) due to the reliance, the alleged infringer will be materially prejudiced if the patentee is permitted to proceed with its infringement suit.¹⁷⁵

The Federal Circuit describes a typical equitable estoppel situation as follows: (1) the infringer knows of the patent; (2) the patentee objects to the infringer's activities; (3) the patentee does not seek relief until much later; and (4) the infringer is misled to believe

172. *See id.*

173. *See id.*

174. *See supra* Part II.

175. *ABB Robotics, Inc. v. Robotics Corp.*, 52 F.3d 1062 (Fed. Cir. 1995); *accord*, *A.C. Aukerman Co. v. R.L. Chaides Const. Co.*, 960 F.2d 1020 (Fed. Cir. 1992) (en banc); *B. Braun Med. Inc. v. Abbott Lab.*, 124 F.3d 1419, 1425 (1997).

the patentee will not act on its patents.¹⁷⁶

*Braun v. Abbott*¹⁷⁷ discusses both equitable estoppel and patent misuse.¹⁷⁸ Braun invented and manufactured a valve and licensed the patent to Abbott, with the restriction that it not be used in “extension sets” that permit the delivery of additional fluids and drugs.¹⁷⁹ Abbott instead asked NP Medical, its co-defendant, to manufacture similar valves for it.¹⁸⁰ In finding an equitable estoppel defense inapplicable, the court stated the standard equitable estoppel definition,¹⁸¹ and held that there was no equitable estoppel defense available to Abbott because

[T]here exists no evidence that Braun communicated anything to Abbott or NP Medical that would lead them to believe that Braun would not sue them for infringement if they made, used, or sold competing valves . . . although Braun did have relations with Abbott, those relations concerned [other matters].¹⁸²

With standard development technologies, the availability of the equitable estoppel defense would be a very fact intensive inquiry. A court would have to consider whether the standard creator attempted to assert its patents and then ceased doing so, whether the parties were at arms length, and what the expectations of the parties were in granting and using the development tool technology. This is probably the most difficult argument for an anonymous party who downloads software from the Internet to make. Although the company must know that its technology is being mass deployed and mass used, the company often is not on specific notice of who exactly is using it and for what purpose.

D. The Child of Patent Misuse: Copyright Misuse

Much of software is protected under copyright laws because until recently, it was not patentable. Additionally, some software developers chose not to patent their technologies due to the time and expense involved.¹⁸³ As previously stated, copyright laws confer a

176. *A.C. Aukerman Co. v. R.L. Chaides Const. Co.*, 960 F. 2d 1020 (Fed. Cir. 1992) (en banc); *Wang*, 103 F.3d at 1579.

177. *Braun v. Abbott*, 124 F.3d 1419 (Fed. Cir. 1997).

178. *See supra* Part V.A.3.

179. *See Braun*, 124 F.3d at 1422.

180. *See id.*

181. *See id.* *See also supra* note 176 and accompanying text.

182. *Braun*, 124 F.3d at 1425.

183. *See supra* Part III.B.

limited monopoly on the copyright holder.¹⁸⁴ Monopolies can be abused in an attempt to restrain competition in a situation analogous to antitrust law restrictions. Copyright misuse can be offered as a defense to prevent recovery when the plaintiff has engaged in misconduct analogous to that found in patent misuse, or anticompetitive behavior.

The misuse doctrine exists primarily in patent law,¹⁸⁵ but was adopted most notably as a viable defense to copyright infringement in *Lasercomb America, Inc. v. Reynolds*.¹⁸⁶ The Fourth Circuit held that "a misuse of copyright defense is inherent in the law of copyright just as a misuse of patent defense is inherent in patent law."¹⁸⁷ But *Lasercomb* is tenuous: prior to 1990,¹⁸⁸ there had been only one case that applied the doctrine to bar recovery on a copyright infringement claim.¹⁸⁹ Indeed, commentators have suggested that this is a bizarre holding and is not in accordance with standing law.¹⁹⁰

But a copyright misuse defense might be attractive to litigants: the copyright misuse defense is a per se violation, since it avoids the uncertainties and difficulties involved in proving that a restraint of trade is unreasonable under the antitrust laws.¹⁹¹ And, like patent misuse, an antitrust violation may not be necessary for the defense to be upheld.¹⁹²

*Lasercomb*¹⁹³ is the exceptional case. Other courts have completely denied the existence of the copyright misuse defense on similar facts.¹⁹⁴ From the rulings, one might think that the Fourth

184. See *supra* Part III.B.

185. See *supra* Part V.A.1.

186. *Lasercomb Am., Inc. v. Reynolds*, 911 F.2d 970, 976 (4th Cir. 1990).

187. *Id.* at 973.

188. *Lasercomb Am., Inc. v. Reynolds*, 911 F.2d 970 (4th Cir. 1990) was decided in 1990.

189. See Abromats, *supra* note 124, at 634. In a case involving ASCAP, *M. Witmark & Sons v. Jensen*, 80 F. Supp 843 (D. Minn. 1948), *aff'd sub nom. M. Witmark & Sons v. Berger Amusement Co.*, 177 F.2d 515 (8th Cir. 1949). ASCAP licensees were forced to purchase a blanket license for every song that ASCAP owned. The court held that ASCAP was using the copyright laws impermissibly and had exceeded the constitutionally granted monopoly, under public policy grounds.

190. See generally Abromats, *supra* note 124.

191. See Marshall Leaffer, *Engineering Competitive Policy and Copyright Misuse*, 19 DAYTON LAW REV. 1087, 1088-89 (1994). Antitrust regulates nearly every industry.

192. See *Research, Inc. v. Computer Assocs. Int'l Inc.*, 793 F. Supp. 68, 69 (E.D.N.Y. 1992).

193. *Lasercomb Am., Inc. v. Reynolds*, 911 F.2d 970, 976 (4th Cir. 1990).

194. *Bellsouth Adver. & Publ'g. Corp. v. Donnelley Info. Publ'g., Inc.*, 933 F.2d 952, 951 (11th Cir. 1991), *rev'd en banc*, 999 F.2d 1436 (11th Cir. 1993); *Practice Management Info. Corp. v. American Med. Ass'n*, 877 F. Supp. 1386, 1393-94 (C.D. Cal. 1994); *Microsoft Corp.*

Circuit stands alone with *Lasercomb*. However, the Fifth Circuit recently upheld an injunction based on anticipated success of a misuse defense in a software dispute.¹⁹⁵

But the concerns for the software technology creator are the same; any further extension of the doctrine can give rise to similar fears as with patent misuse and antitrust concerns.¹⁹⁶ A lesser standard than that for antitrust liability is acceptable to find the defense; any refusal to license can implicate anti-competitive effects that antitrust so desperately seeks to prohibit.

VI. COMPETING POLICY CONCERNS AND THE CONSTITUTIONALLY GRANTED MONOPOLIES

A. *Juxtapositions of Contradictions*

There is a terrible contradiction of policies at work here. The software development community demands, needs, and requests decent standards. But spending the time, money, and extensive resources to create and develop a standard creates a relationship of love and hate. Developers are absolutely dependent on the company's standard proprietary creations: they wait anxiously for the next bug-fixed release to come out, for improvements in the tools or the language, and for the standard to evolve in ways that they see fit. Companies usually do what is best for business, which usually does not involve giving charitable donations of their inventions to the software community.

The relationship between a company and the software developers who incorporate its technology has its juxtaposition in the law as well. Our antitrust laws were designed to foster competition, and the Constitution gives inventors and creators protection for their works. But in antitrust, copyright, and patent law, there are major offensive and defensive tactics that an opponent or intellectual property infringer could take against the standard development tool creator to try to strip it of full rights to its works. What may appear as anti-competitive behavior by the software developer may just be the result of the natural forces working in the market place, especially in an industry such as software.

The tension has to break. A decision to apply antitrust and the

v. BEC Computer Co., 818 F. Supp. 1313, 1316-17 (C.D. Cal. 1992).

195. DSC Communications Corp. v. DGI Techs, Inc., 81 F.3d 597 (5th Cir. 1996).

196. See *supra* Part V.A.1.

intellectual property defenses related to antitrust doctrines could dramatically extend the scope of antitrust intervention in the market.¹⁹⁷ Many commentators have argued that *sua generis* legislation is needed¹⁹⁸ for a field that is grasping for anything it can take from normal common law principles.

Others argue that if a company is to create a standard, it must donate it to the world and allow the standard to be intellectual-property free.¹⁹⁹ And still others propose (and in the case of some standards setting organizations, actually require) that there be a mandatory reasonable license fee for a standard to find harmony with the law.²⁰⁰

As it stands now, any extension of the law that finds a requirement to license or donate to the public could go directly against the Constitution, which granted authors and inventors the protection of their property in the first place.²⁰¹ That is, courts refuse to create a practice of forced licensing.²⁰²

B. A Balance Must Be Struck

A medium should be struck. With such a legal arsenal in the technology user's position, it is likely that a standard owner would be forced to give up her intellectual property rights. A company that invested so much to create, could have its rights stripped away from it, and face years of legal fees.

One possible solution is this. *Sua generis* legislation should be enacted to exempt standard creators from liability for antitrust offenses. Furthermore, the patent misuse act should likewise be modified to disallow the use of such defense when it implicates standardized technologies because of the inappropriateness of the application of 35 U.S.C. § 271(b)(4) to standards.²⁰³ This would allow the standard creator to keep his constitutionally protected rights

197. See Lemley, *supra* note 46, at 1085. Perhaps the scope has already been extended with the Department of Justice's prosecution of Microsoft and Intel. See Michael Moeller, *Microsoft, DOJ Hit the Courts* (Oct. 19, 1999) <<http://www.zdnet.com/pcweek/stories/news/0,4153,361415,00.html>>.

198. See, e.g., Pamela Samuelson et al., *A Manifesto Concerning The Legal Protection Of Computer Programs*, 94 COLUM. L. REV. 2308, 2312 (1994).

199. See Lemley, *supra* note 46, at 1090.

200. See *id.* (discussing the practices of ANSI).

201. See U.S. CONST., art. 1, § 8, cl. 8.

202. See Robert P. Merges, *Contracting Into Liability Rules: Intellectual Property Rights and Collective Rights Organizations*, 84 CAL. L. REV. 1293 (1996).

203. See *supra* notes 124-131 and accompanying text.

and be compensated for them.

The creator should be required to license the standards to parties to whom it would otherwise refuse, but not under principles of existing law where no compensation would be given (e.g., if a misuse defense is successful the patentee cannot assert his patent rights at all). The compulsory licensing requirements should require: 1) the developer of the technology actively to encourage the use of its product as a standard; 2) an independent body of industry persons to determine the appropriate royalty or licensing fee for the licensing of the standard; and 3) such licensing to occur under the same terms that other parties receive – i.e., what is fair in the marketplace.

This would also leave the standard creation companies with adequate recourse. Suits might be permitted if the licensee breached the contract terms. The parties would also be able to set prices and negotiate separately in their customary licensing practices. A governing body separate from the courts would only step in when a dispute arose between a development tool creator and a proposed licensee. And again, this solution would only occur if the party actively took steps to make its product a standard.

This solution, although taking some rights away from all parties involved, would resolve the tension that currently exists in the software development tool industry. Such a solution would cut short many years of litigation and disputes over standardized technologies in a field where rapid evolutionary changes are the norm. It would also protect the standard developer companies from current legal principles that attempt to take away their constitutionally granted rights. In addition, it would provide the companies needing standards with the resources they need at a reasonable fee.²⁰⁴ Holding companies liable for antitrust liability and providing proposed licensees with a variety of defenses would hold these providers subject to liability and would contravene the Constitution, which gives them full rights to ownership of their intellectual property.

VII. CONCLUSION

Companies currently have much to gain if they develop standard software development tool technology. At the same time, these entities have much to give to the software industry that desperately needs these standards. But with the privilege of owning a standard comes a plethora of legal issues about which to be concerned. With a

204. See *supra* Part IV.A.1.b for a discussion of the essential facility doctrine.

technology's acceptance comes the dependence of software developers on the resulting standard. Without access to the standard, product development efforts by software developers could be crippled.

This Comment proposes that Congress should enact legislation to create a safe harbor for standard creating entities. Such legislation would allow these entities to own their software along with a respective bundle of rights, and would create separate rules to protect their software. At the same time, this legislation would protect users of the standards through mandatory licensing of the development tool technology at rates and on terms determined by the contracting parties and an independent tribunal. Without such proposed legislation, prosecution of standard creation entities could be wrongly prosecuted through the misapplication of modern legal principles, and the restriction of inventive and expressive rights will occur.