UIC John Marshall Journal of Information Technology & Privacy Law

Volume 5
Issue 2 *Computer/Law Journal - Fall 1984*

Article 5

Fall 1984

Computer Copyright Law: An Emerging Form of Protection for Object Code Software After Apple v. Franklin, 5 Computer L.J. 233 (1984)

Anderson L. Baldy III

COMPUTER COPYRIGHT LAW: AN EMERGING FORM OF PROTECTION FOR OBJECT CODE SOFTWARE AFTER *APPLE v. FRANKLIN*

The computer industry's phenomenal expansion in recent years has intensified the need for effective protection of software property rights.[1] A vast disparity exists between development costs required to produce marketable software and the cost to misappropriate or "pirate" a copy of existing software.[2] Thus, inadequate protection for this form of intellectual property serves as a disincentive to independent and innovative software development.[3]

Software producers have traditionally relied on trade secrecy laws as the primary method of enforcing and preserving intellectual property rights.[4] While trade secrecy laws provide adequate protection on an individual contractual basis,[5] the restrictive licensing and contractual agreements which form the basis of this protection artificially suppress reproduction, distribution, and ultimately, use of

1. *See* INTERNATIONAL BUREAU OF THE WORLD INTELLECTUAL PROPERTY ORGANIZATION, MODEL PROVISIONS ON THE PROTECTION OF COMPUTER SOFTWARE (Geneva 1978), *reprinted in* 11 LAW & COMPUTER TECH. 2, 3 (1978) (legal protection for computer software is desirable). *See also* Note, *Copyright Protection for Firmware: An International View*, 4 HASTINGS INT'L & COMP. L. REV. 473, 475 (1981). "The need for protecting the investment in software . . . is increasing due to . . . (1) the increasing percentage of software cost to the total cost of a computer system . . . , and (2) the trend toward standardized, mass marketed software, and away from single user custom made software." *Id.*

2. *See* NAT'L COMM'N ON NEW TECHNOLOGICAL USES OF COPYRIGHTED WORKS, FINAL REPORT at 10 (1978) (hereinafter cited as CONTU REPORT) (the cost of copying a computer program is small), *reprinted in* 3 COMPUTER L.J. 53, 57 (1981).

3. "[I]f the cost of duplicating information is small, then it is simple for a less than scrupulous person to duplicate it. This means that legal . . . protection for the information is a necessary incentive if such information is to be created and disseminated." *Id. See also infra* note 31 and accompanying text.

4. *See* MacGrady, *Protection of Computer Software - An Update and Practical Synthesis*, 20 HOUS. L. REV. 1033, 1050 (1983) (trade secrecy is overwhelmingly preferred by the software industry as a legal protective method).

5. *See* Nimtz, *Development of the Law of Computer Software Protection*, 61 J. PAT. OFF. SOC'Y 3, 20 (1979) (discussing mechanics of licensing and trade secrecy agreements).

the work.[6] In addition, there are risks inherent in the use of trade secrecy in a large market. Broad software dissemination exposes a vendor who relies solely on trade secrecy protection to the possible dissolution of proprietary rights should his product be deemed "lost to the public domain."[7] Copyright protection, on the other hand, by granting the copyright owner exclusive reproduction and distribution rights, would assure that the vendor's important proprietary interests would not be lost due to dissemination through a mass market. Copyright protection, therefore, encourages widespread marketing of the work.[8]

While scientific discoveries and technological advances have provided more efficient forms of expression and communication, the scope of works accorded protection under copyright law has only gradually evolved.[9] The law as applied to computer software is developing on an ad hoc basis. While source code programs[10] have consistently been accorded copyright protection, object code programs[11] have only recently been deemed copyrightable.[12] Further, the exact boundaries of the protection granted to object code programs as an alternate or supplement to trade secret protection[13]

---

6. *See id.* at 21.

7. *Id.* at 20-21 (discussing inherent risks accompanying trade secrecy protection).

8. *See* Mazer v. Stein, 347 U.S. 201, 219 (1954) (the grant of copyright encourages personal innovation and also provides for public dissemination of the fruits of an author's work).

9. *See* H.R. REP. No. 1476, 94th Cong., 2d Sess. 116, *reprinted in* 1976 U.S. CODE CONG. & AD. NEWS 5659, 5664-65 (the history of copyright law has been one of gradual expansion in the types of work accorded protection) (hereinafter cited as HOUSE REPORT). *See also* Goldstein v. California, 412 U.S. 546, 562 (1973) (the history of copyright law shows expansion of federal protection as technology expands). This expansion, however, has come about slowly. There were no major alterations in the 1909 Copyright Act until the General Revision of 1976, Pub. L. No. 94-553, 90 Stat. 2541 (1976) (codified as amended in scattered sections of 17 U.S.C.).

10. Source code programs usually perform a specific task for the computer user, such as balancing a checkbook or playing a game. For a more extensive definition and explanation, see *infra* notes 19-21 and accompanying text.

11. Object code programs manage the internal operations of the computer and facilitate use of the source code programs. For elaboration, see *infra* notes 22-24 and accompanying text.

12. *See* Data Cash Sys., Inc. v. JS & A Group, Inc., 480 F. Supp. 1063, 1069 (N.D. Ill. 1979) (while source code is a copyrightable writing, object code is an uncopyrightable mechanical tool or machine part), *aff'd on other grounds*, 628 F.2d 1038 (7th Cir. 1980).

13. *See* MacGrady, *supra* note 4, at 1040 (trade secrecy is often chosen over copyright law to protect software property interests). *See also* Note, *Trade Secrets and the Skilled Employee in the Computer Industry*, 61 WASH. U.L.Q. 823, 838 (1983) ("The uncertainty of traditional statutory protection for computer software has caused many firms to seek trade secret protection.").

have not been delineated by the federal judiciary.[14] The recent decision in *Apple Computer, Inc. v. Franklin Computer Corp.*[15] merely provides that some object code is copyrightable subject matter regardless of its embodiment medium, in that case Read Only Memory (ROM) chips. Thus, *Apple* stands for the proposition that the embodiment of object code in a ROM does not preclude copyright protection, but does not provide guidelines for determining what constitutes a copyrightable ROM-embedded program.

This Note compares the effectiveness of currently-applied software protection practices with that of the copyright law. The objections to object code copyrightability are presented, and the justification for object code protection is analyzed in light of recent congressional modifications of the Copyright Act. Section I provides a brief orientation to the terms and phrases necessary for a cognizant discussion of object code copyrightability. Section II argues that increased reliance on copyright protection for object code programs is necessary to alleviate risks associated with current protective practices. Section III sets forth the statutory framework and case history relevant to object code copyright protection. Section IV concludes that while ROM-embedded object code falls within this framework of copyrightable subject matter, there exists a distinction between copyrightable ROM-embedded object code and uncopyrightable uses of ROM-embedded hardware.

I. TERMINOLOGY

Familiarity with the terms and phrases used in the computer software industry is a prerequisite to a cognizant discussion of computer copyright law. "Hardware" consists of the physical electrical circuits, tape drives, motors, readers, printers, relays and memory which comprise the computer's mechanical and electrical components.[16] "Software," often called a computer program, is the set of instructions used directly or indirectly in a computer to bring about a certain result.[17] Software manipulates and instructs the various

14. *See* Tandy Corp. v. Personal Micro Computers, Inc., 524 F. Supp. 171 (N.D. Cal. 1981) (object code can be considered a "copy" of a copyrighted computer program). *See also* Apple Computer, Inc. v. Franklin Computer Corp., 714 F.2d 1240 (3d Cir. 1983) (a computer program, whether in object or source code form is a copyrightable "literary work"), *cert. dismissed*, 104 S. Ct. 690 (1984); Williams Elec., Inc. v. Artic Int'l, Inc., 685 F.2d 870 (3d Cir. 1982) (plaintiff had the right to protect through copyright law object code deemed "artistic expressions").

15. 714 F.2d 1240.

16. *See* Gemignani, *Legal Protection for Computer Software: The View from '79*, 7 RUTGERS J. COMPUTERS, TECH. & L. 269, 273 (1980) (definition of "hardware").

17. 17 U.S.C. § 101 (1982). The 1980 Amendment to the 1976 Copyright Act in-

hardware components to perform desired functional tasks. The instructions, typically prepared by a highly skilled programmer, may have considerable market value if developed and organized in an innovative and efficient manner.

There exist various software types, or "levels," classified by the degree of sophistication of the instructions utilized.[18] The most sophisticated type of software, the "source code" program, features instructions comprised of English language commands. BASIC and FORTRAN are examples of source code languages.[19] Source code indirectly instructs the computer hardware to perform a variety of specific functional tasks such as word and data processing, or arithmetic tasks such as mathematical calculations.[20] Source code is an independent and distinct product and is typically marketed separately from the computer hardware itself.[21]

The mechanical hardware by itself cannot comprehend the highly sophisticated English-based source code commands. To process a source code program, a less sophisticated program capable of directly instructing the hardware's internal functions is required.[22] This processing program is called "object code."[23] Most microcomputers feature object code software permanently stored on "Read Only Memory" (ROM) silicon chips which are built into the hard-

---

cluded this definition of "computer program." Pub. L. No. 96-517, § 10(a), 94 Stat. 3015, 3028 (1980). *See also* A. TANENBAUM, STRUCTURED COMPUTER ORGANIZATION 1 (1976) ("A sequence of instructions describing how to perform a certain task is called a program").

18. TANENBAUM,*supra* note 17, at 5.

19. *Id.* at 7. Some examples of FORTRAN source code instructions are: "READ I" instructs the computer to read in the variable "I"; "IF (I.GT.5) GO to 70" instructs the computer to go to instruction number 70 if variable "I" is greater than five (5).

20. Source code languages may be referred to as "problem-oriented languages" because they are employed by the programmer to carry out and solve basic tasks. *See id.*

21. *See* Popper, *Software: The New Driving Force*, BUS. WK., Feb. 27, 1984, at 74. The nationwide demand for more effective and efficient programs has spurred the growth of independent software sales, which in 1984 alone are expected to exceed $10 billion. *Id.*

22. Hardware can only comprehend instructions in binary form; therefore, an unsophisticated program comprised of instructions which feature clusters of "1's" and "0's" is necessary to enable the hardware to perform the functional tasks ordered by the source code commands. *See* TANENBAUM, *supra* note 17, at 1 (these primitive instructions are necessary to allow people to communicate with the computer).

23. *See* Note, *Copyright Protection of Computer Program Object Code*, 96 HARV. L. REV. 1723, 1724 (1983). Clusters of "0's" and "1's" are the only symbols recognized by binary, or digital computers. *Id.* A program comprised of a series of these clustered symbols is called "object code." *Id.* at 1725.

ware.[24] Object code is electromechanically or photoelectrically embedded on the ROM silicon chip.[25] If the ROM were viewed under a microscope, the viewer could observe an electronic circuitry pattern which is the object code. The object code, if transcribed onto paper, would appear as a series of "1's" and "O's," representing the ROM chip's circuitry pattern.[26] While the ROM-embedded object code is susceptible to misappropriation by exact duplication of the ROM chip, object code transcribed onto paper is an unintelligible cluster of symbols, without value even to an expert.

II. THE INADEQUACY OF CURRENT PROPRIETARY PRACTICES AND THE NEED FOR COPYRIGHT PROTECTION

The sale of hardware embedded with ROM-resident object code dominates the microcomputer market. The remaining market share lies in peripheral source code software sales.[27] Major producers such as IBM, Burroughs, Apple and Tandy compete in both phases of this market.[28] Microcomputer hardware marketed by a major producer features permanently affixed object code specifically designed to be compatible with the producer's separately marketed source code.[29] The smaller competitors tend to concentrate on only one aspect of the market, selling either source code compatible with the object code of a major producer's hardware, or hardware which features object code compatible with a major producer's source code.[30]

---

24. *See* Apple Computer, Inc. v. Franklin Computer Corp., 545 F. Supp. 812, 813 (E.D. Pa. 1982).

25. *Id.* Information to be permanently stored on a ROM silicon chip is implanted by inscribing the equivalent of "on/off" switches, arranged according to the "1's" and "O's" of the object code program. This circuitry pattern of "on/off" switches determines the sequence of electrical events that occur on the chip, and when in operation constitute the physical realization of the object code program. *Id. See generally* Boraiko, *The Chip*, 162 NAT'L GEOGRAPHIC 421, 426-31 (1982) (providing a description of the process required to create a silicon chip).

26. *See* Gemignani, *supra* note 16, at 273.

27. *See* Popper, *supra* note 21, at 75. The total estimated value of U.S. microcomputer hardware sales in 1984 is nearly $12 billion. Peripheral source code software sales for use with these microcomputers is estimated at around $2 billion. *Id.*

28. *See generally* Popper, *supra* note 21; Gemignani, *supra* note 16, at 274. A major computer hardware manufacturer, such as IBM, is in the best position to develop and market software capable of properly operating on its independently marketed hardware. IBM no longer supplies the software as an integral part of its hardware sales package. *Id.* at 274 n.21.

29. Gemignani, *supra* note 16, at 274 n. 21.

30. *See* Frank, *The New Software Economics, Part 2*, COMPUTERWORLD, Jan. 15, 1979, at 5. The significance of the independent software industry's impact is increasing. The total volume of the industry's sales in 1977 was only $.5 billion, roughly

In seeking to market microcomputers which feature object code compatible with a major producer's widely marketed source code, small competitors are faced with significant disincentives to independent development. The substantial research and development costs[31] and time requirements for independent development, as well as the uncertain availability of copyright protection for object code, serve as incentives for misappropriation.[32]

A. Attempts to Physically Prevent Misappropriation—The "Black Box"

Because of the uncertainty of copyright protection for object code, many producers have pursued less secure protective methods.[33] As noted, ROM-embedded object code is unintelligible to the human eye, and object code transcribed onto paper is a cluster of symbols meaningless to anyone not involved with its original development.[34] Dissemination in either of these forms may provide an effective "black box" barrier to misappropriation.[35]

There have been attempts to thwart copying by protecting the ROM-embedded object code in an epoxy or cemented encasement, but such methods have not eradicated the possibility of exact reproduction by determined misappropriators. ROM-embedded object

---

equal to IBM's total software sales. *Id.* The over 3,000 independent software producers now in existence are expected to generate $30 billion in annual sales by 1988. Popper, *supra* note 21, at 75. *See also* CONTU Report, *supra* note 2, at 11 ("it is all but certain that programs written by non-machine manufacturers will gain an increasing share of the market.").

31. *See* Note, *supra* note 1, at 475 ("The *creation* of software and firmware requires large investments of both time and money.") (emphasis added). *See also* CONTU Report, *supra* note 2, at 10 (software *duplication* costs are often not substantial) (emphasis added).

32. *See* Telex Corp. v. International Business Mach. Corp., 367 F. Supp. 258 (N.D. Okla. 1973) (Former IBM employees revealed to Telex a new source code program which they assisted in developing for IBM. By hiring away key IBM employees, Telex saved roughly $10 million in software development costs.), *rev'd on other grounds*, 510 F.2d 894 (10th Cir. 1975), *cert. denied*, 423 U.S. 802 (1975).

33. *See* MacGrady, *supra* note 4, at 1045.

34. Article, *Legal Proprietary Interests in Computer Programs: The American Experience*, 21 Jurimetrics J. 345, 352 (1981).

35. A "black box" refers to a product which is deliberately marketed in an intricate and unintelligible form. The purpose is to prevent competitors from discovering the product's design, successfully reproducing the product, and reaping a profit by circumventing research and development costs incurred by the original developer.

Object code's unintelligibility to misappropriator's is viewed as a benefit by its developers. Software developers often lease their product in object code form in an attempt to prevent the lessee from reading, understanding and reproducing the software. *See* Note, *Microcomputer Emulation: Protecting Manufacturers From Computer Copying*, 17 Suffolk U. L. Rev. 656, 659 (1983).

code may be "reverse-engineered" and deciphered by ingenious competitors capable of reproducing a silicon chip with moderately sophisticated facilities and resources.[36] Thus, reliance on a physical "black box" barrier as the sole method for protecting widely marketed object code leaves the producer susceptible to a complete loss of his proprietary rights.

B. Legal Methods of Preventing Misappropriation

The three legal mechanisms applicable to protect software proprietary interests are patent, trade secrecy and copyright law.[37] Of these, patent protection is the most encompassing, providing the owner with an infringement remedy against those who copy or merely use the patented idea, process or device.[38] Due to the rigorous standards for patentability, however, software is generally considered inappropriate subject matter. Patentable subject matter must meet the standards of "usefulness," "novelty"[39] and "nonobviousness."[40] While a mathematic or functional principle incorporated in a program may satisfy these requirements, the software itself is patentable only if the manner in which the principle is utilized is also new and useful.[41] As applied by most federal courts, this analysis has operated to disqualify most claims of software patentability.[42]

---

36. *See* Durham Indus., Inc. v. Tomy Corp., 630 F.2d 905, 908 (2d Cir. 1980) (original works disseminated in the public domain may be copied unless copyright, patent, trademark or trade secrecy protection is secured).

37. *See generally* Rose, *Protection of Intellectual Property Rights in Computers and Computer Programs: Recent Developments*, 9 PEPPERDINE L. REV. 547, 549 (1982) (discussing the three legal avenues of protection available to the applicant seeking to protect intellectual property interests in software).

38. The Federal Patent Act provides that "whoever invents or discovers any new or useful process, machine, manufacture of composition of matter, or any new and useful improvement thereof, may obtain a patent therefore. . . ." 35 U.S.C. § 101 (1982).

39. A patent will be barred for lack of novelty if the invention had previously been described in a prior patent or printed publication, or if it had been "known or used by others." 35 U.S.C. § 102 (a) (1982).

40. The test of obviousness is whether "the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains." 35 U.S.C. § 103 (1982).

41. *See* Parker v. Flook, 437 U.S. 584 (1978) (a process embodying a scientific principle or mathematical formula held not patentable under § 101 of the Patent Act).

42. *See, e.g.*, Gottschalk v. Benson, 409 U.S. 63 (1972) (program which converted coded decimals to pure binary numbers not patentable). *But see*, Diamond v. Diehr, 450 U.S. 175 (1981) (computer-controlled rubber curing process utilized patentable program). *Diehr* has been interpreted to stand for the proposition that an invention is not unpatentable merely because it includes a computer program. Computer

The legal protection form most widely utilized by the software industry has been trade secrecy law.[43] Available for any marketed information of content and expression, the trade secret must not be generally known in the trade and must provide the recipient with "a competitive advantage."[44] The broad scope of trade secrecy protection has contributed to its popularity within the software industry. Trade secret law protects the most valuable aspect of computer software—its detailed design and logic, including the underlying concepts, techniques, methods and processes.[45] In contrast, copyrighted software is only protected against substantially similar[46] reproductions of the coded instructions. Thus, copyright protection forces a competitor to develop and market a substantially different version of the software, but does not provide protection as encompassing as trade secrecy.[47] Several factors may nevertheless justify attempts to secure copyright protection as an alternative or supplement to trade secrecy protection.

The major disadvantage of trade secrecy as a protective device arises in its application to a mass-marketed product. The essence of trade secret protection consists of individual license or contractual arrangements between buyers and sellers.[48] With software, these arrangements would provide that the secret program will not be

---

software, however, has not yet been explicitly ruled patentable subject matter. *See* MacGrady, *supra* note 4, at 1039. The standard for software patentability is apparently whether it "transforms or reduces an article to a different state or thing." *Gottschalk*, 409 U.S. at 70.

43. *See* MacGrady, *supra* note 4, at 1050 (of the legal methods of software protection, most producers prefer trade secrecy).

44. *See* Imperial Chem. Indus. v. National Distilleries & Chem. Corp., 342 F.2d 737, 742 (2d Cir. 1965) (a trade secret must be protectable and must provide a competitive advantage); E.I. DuPont de Nemours & Co. v. United States, 288 F.2d 904, 911 (Ct. Cl. 1961) ("A trade secret is any information not generally known in a trade.").

45. Trade secrecy offers broader protection than copyright because it can protect what copyright, by statutory definition, cannot—the design. *DuPont*, 288 F.2d at 911.

46. For a discussion of the "substantial similarity" standard, see P. GOLDSTEIN, COPYRIGHT, PATENT, TRADEMARK AND RELATED STATE DOCTRINES 849-50 (1980).

47. *See* Pope, *Protection of Proprietary Interests in Computer Software*, 30 ALA. L. REV. 527, 531 (1979). Because widespread dissemination of software is deemed desirable, copyright protection would appear to be the most effective protective device. Copyright law, however, only protects the physical embodiment of the program, not the idea underlying the program. *Id.*

48. Nimtz, *supra* note 5, at 21. An example of such a restriction might appear as follows:

> The customer shall not in any fashion transfer or make available to any third party the programs or manuals. The customer agrees that the programs and manuals contain information which is proprietary to [vendor] and the customer will not reproduce or disclose this information, in whole or in part, to any third party.

Note, *supra* note 1, at 484.

used or appropriated in a manner not contemplated by both parties. Theoretically, the protection accorded the software could remain in perpetuity provided the secret is properly preserved.[49] Secrecy preservation on a small scale can be achieved with moderate effort, but when applied to mass market situations, the trade secrecy benefits may break down in the face of prohibitive enforcement costs. The larger the market, the more difficult and expensive it is to protect and maintain secrecy. In addition, the developer risks loss of trade secrecy status should the software's details or logic become generally known to the trade or industry.[50]

In contrast, copyright law as a proprietary device for software is neither expensive nor difficult to maintain. Regardless of the extent of a program's distribution, the copyright owner has the right to bring an infringement action upon misappropriation.[51] Assuming the software meets copyright requirements, the owner needs only to affix a copyright registration notice to preserve rights against misappropriation.[52] Reliance on copyright protection would not necessitate a system of cumbersome and expensive individual license or contract agreements. Thus, for a mass marketed product such as computer software, copyright law is a more effective protective device than trade secrecy.

The preferred protective method should also encourage development and dissemination of new and innovative software. Copyright protection fosters such development and thereby promotes the technological progress which benefits the public.[53] While trade secrecy helps to assure the financial interests of the developer, copyright protection can secure both the public and private benefit.[54] The grant of copyright protection is often justified by the premise that it will encourage widespread communication and transfer of information.[55] A trade secret, in contrast, is inherently hostile to the free

49. *See* Nimtz, *supra* note 5, at 19-20.

50. *Id.* at 20-21.

51. Neither 17 U.S.C. § 101 (1982) nor 17 U.S.C. § 102 (1982) provide any limitation on the distribution of a copyrighted work.

52. *See* Peter Pan Fabrics, Inc. v. Martin Weiner Corp., 274 F.2d 487 (2d Cir. 1960) (to comply with § 10 of the Federal Copyright Act, notice must be "affixed to each copy . . . offered for sale").

53. "The economic philosophy behind the clause empowering Congress to grant patents and copyrights is the conviction that encouragement of individual effort by personal gain is the best way to advance the public welfare through the talents of authors and inventors in 'Science and the useful Arts.' " Mazer v. Stein, 347 U.S. 201, 219 (1954).

54. *Id.*

55. *Id. See also* Goldstein v. California, 412 U.S. 546, 555 (1947) (to promote science and art, Congress may reward authors by granting them control over commer-

flow of ideas.[56]

III. SCOPE OF PROTECTION—LIMITING PRINCIPLES AND STATUTORY FRAMEWORK

Copyright law is now considered by commentators and scholars[57] to provide the most effective and appropriate apparatus for assuring proprietary protection of mass-marketed software. Unfortunately, the technology of software has outpaced legislative and judicial initiative. Copyright law's application as a method of protecting software property interests has not been consistently extended to all software forms. Of the existing forms, only source code is uniformly considered copyrightable.[58] The courts and Congress have accorded ROM-embedded object code only tenuous and sporadic protection.[59]

The industry's reluctance to rely on copyright protection is due in part to the copyright law's limited scope and applicability.[60] The copyright monopoly is only available for "original works of authorship" which are "fixed in any tangible medium of expression."[61] While software typically originates as a written "work of authorship," its existence as a program in a computer consists of electrical and mechanical forms, arguably more similar to a patentable process or idea than a copyrightable written form of expression.[62] If the "fixed in any tangible medium of expression" standard is interpreted to require that a copyrightable work must be in human-intelligible form, ROM-resident object code will not qualify for protection.[63] These elementary objections, coupled with judicial adherence to traditional interpretation of the primary copyright law

---

cial use of work); CONTU REPORT, *supra* note 2, at 11 (to encourage the creation and broad distribution of computer programs in a competitive market, copyright protection is necessary).

56. *See* Nimtz, *supra* note 5, at 19-20.

57. The most prestigious authority typically cited as advocating copyright protection is the Software Subcommittee to the National Commission on New Technological Uses of Copyrighted Works (CONTU). *See* CONTU REPORT, *supra* note 2, at 11. *But see* MacGrady, *supra* note 4, at 1040 ("As recently as fifteen years ago, copyright seemed to be the best available form of legal protection for computer programs. However, such has not been the case.").

58. *See* 1 M. NIMMER, NIMMER ON COPYRIGHT § 2.04[C] nn. 25.1, 25.2 (rev. ed.) (courts are split on object code copyrightability).

59. While the Copyright Office will accept for registration transcribed versions of object code programs, ROM chips cannot be registered.

60. *See* Note, *supra* note 13, at 838 (statistics depicting lack of reliance on copyright as a protective device).

61. 17 U.S.C. § 102(a) (1982).

62. *See*, CONTU REPORT, *supra* note 2, at 31.

63. *See* Article, *supra* note 34, at 352 and text accompanying note 34.

standards noted above, have contributed to the continued uncertainty regarding the copyrightability of ROM-resident object code.

A. Original Work of Authorship

In *Baker v. Selden*,[64] the United States Supreme Court first interpreted the "original work of authorship" standard and in so doing significantly limited copyright law's protective scope.[65] The Court ruled that copyright law did not provide protection against appropriation of an idea or process utilized in an author's work, but rather protected only the particular expression adopted by the author to convey his idea or process.[66] Thus, a book containing explanations and examples of an author's independently developed accounting procedure was copyrightable, but the copyright did not preclude others from publishing substantially different explanations and examples which utilized the author's procedure.[67] Essentially, copyright prevents misappropriation of an idea's particular expression rather than the use of the idea itself.[68] This qualification of copyright's protective scope has become significant in object code infringement disputes, fostering claims that a program operating in a computer is an idea not subject to copyright protection.[69]

B. Fixation in a Tangible Medium of Expression

The Copyright Act of 1976 protects "original works of authorship"[70] including "literary works . . . expressed in words or numbers . . . or numerical symbols or indicia"[71] which are "fixed in any tangible medium of expression . . . from which they can be perceived, reproduced or otherwise communicated, either directly or *with the aid of a machine or device*."[72] If the Copyright Act is inter-

64. 101 U.S. 99 (1879).

65. *Id.* at 104.

66. *Id.*

67. *Id.* at 103. The "substantially different" standard stems from the idea-expression dichotomy. *See, e.g.*, Stern Elecs., Inc. v. Kaufman, 669 F.2d 852, 855 (2d Cir. 1982) (registering videogame programs would not prevent duplicating game by writing different program which produced the same resulting video display).

68. The most important limitation upon copyright law's protective scope is the distinction between an "idea" and a particular "expression" of that idea. Copyright law protects only the latter. *Baker* is typically cited as the earliest statement of this principle.

69. *See, e.g.*, Apple Computer, Inc. v. Franklin Computer Corp., 545 F. Supp. 812 (E.D. Pa. 1982) (defendant claimed ROM-resident object code was an uncopyrightable machine part).

70. 17 U.S.C. § 102(a) (1982).

71. *Id.*

72. 17 U.S.C. § 101 (emphasis added).

preted literally, object code would satisfy the fixation requirements, because it may be perceived indirectly by the user with the aid of the source code. Literal interpretation of the "literary works" definition also suggests that the communication need not be human-intelligible. Judicial analysis of the fixation requirement, however, has hindered the availability of copyright protection for object code.[73]

The Supreme Court first interpreted this requirement in *White-Smith Music Publishing Co. v. Appollo Co.*[74] The Court determined an author's work would satisfy the fixation requirement only when expressed in human-intelligible form.[75] Thus a player-piano roll was not an illegal "copy" of the copyrighted sheet music melody which it played. A "copy," observed the Court, must be readable by the human eye.[76] Copying of source code software, which consists of sophisticated English-based instructions, would therefore be prohibited. Replication of ROM-embedded object code, however, because it is comprised of unintelligible commands, would not be a copyright law violation under the *White-Smith* intelligibility standard.[77]

C. Statutory Framework

Some courts have allowed the *White-Smith* intelligibility requirement to prevail over the literal application of the current Copyright Act.[78] Congress, when enacting the current Act's relevant provision, felt that further deliberation as to the exact scope of software copyright protection was needed before a definitive and explicit legislative solution could be offered.[79] In lieu of extending protection to object code under the 1976 Act, Congress added section 117 which declared that the new Act extended no new rights to works used "in conjunction with automatic systems capable of storing, processing, retrieving or transferring information, or [to works used] in conjunction with any similar device, machine or process."[80] This provision effectively maintained the status quo with respect to computer uses of copyrighted works.[81]

---

73. *See, e.g.*, Data Cash Sys., Inc. v. J S & A Group, Inc., 480 F. Supp. 1063 (N.D. Ill. 1979) (copyrightable work must be in human-intelligible form), *aff'd*, 628 F.2d 1038 (7th Cir. 1980).

74. 209 U.S. 1 (1908).

75. *Id.* at 16.

76. *Id.* at 14.

77. *See supra* notes 19-26 and accompanying text.

78. *See, e.g., Data Cash*, 480 F. Supp. at 1068-69.

79. *See* HOUSE REPORT, *supra* note 9, at 5731 (Congress felt it was premature to change existing law regarding computer uses because the problems were not sufficiently developed for definitive legislative solution).

80. 17 U.S.C. § 117 (1976).

81. *See* Apple Computer, Inc. v. Formula Int'l, Inc., 562 F. Supp. 775, 779 (C.D. Cal.

Thus, in *Data Cash Systems v. J S & A Group*,[82] the earliest decision to address object code copyrightability, the district court determined that the effect of the 1976 Act was to leave the object code developer with the same rights as existed under previous federal copyright law.[83] In that case, the ROM silicon chip in which the object code was embedded was held to be an unintelligible medium of expression. The duplicated ROMs, therefore, could not be "copies" under the copyright law.[84] This being the case, the court determined that no copyright infringement could exist, even though the defendant duplicated and distributed replicas of the plaintiff's ROM-resident object code program.[85]

In 1978, the congressionally created National Committee on New Technological Uses of Copyrighted Works (CONTU) presented to Congress its study regarding software copyrightability, and made three specific legislative recommendations.[86] First, CONTU recommended that the 1976 Act be amended to explicitly state that "computer programs" are copyrightable subject matter; second, the stopgap section 117, which restricted copyright protection of software, should be deleted; and third, a provision allowing consumer adaptations of copyrighted programs should be adopted.[87]

In 1980, Congress amended the Copyright Act, incorporating the second and third CONTU recommendations.[88] Although the 1980

1983) (§ 117 preserved status quo pending study and congressional decision), *aff'd*, 725 F.2d 521 (9th Cir. 1984).

82. 480 F. Supp. 1063 (N.D. Ill. 1979), *aff'd*, 628 F.2d 1038 (7th Cir. 1980).

83. *Id.* at 1067.

84. *Id.* at 1069. *Cf.* DeSilva Constr. Corp. v. Herrald, 213 F. Supp. 184 (M.D. Fla. 1962) (a structure is not a copy of the architectual plans upon which the structure is based). Similarly, the *Data Cash* court determined that a replicated ROM-embedded object code program was not a copy of the original object code program. 480 F. Supp. at 1069. On appeal, *Data Cash* was affirmed on the substantially different ground that the plaintiffs had not properly included a copyright notice on the ROM. 628 F.2d at 1038. Commentators have suggested that the affirmance on alternate grounds, and the refusal by the Seventh Circuit to adopt the district court's conclusion that human intelligibility is a prerequisite to copyrightability, serve as an implicit rejection of the lower court's determination that ROM-embedded object code is uncopyrightable. *See, e.g.*, Boorstyn, *Copyrights, Computers, and Confusion*, 63 J. PAT. OFF. SOC'Y 276, 284 (1981) ("Implicit in the [Seventh Circuit's] decision is the recognition of copyrightability of the computer program as fixed in the medium of a ROM.").

85. 480 F. Supp. at 1068-69.

86. *See* CONTU REPORT, *supra* note 2. The CONTU Commission was created in 1974 to make recommendations to Congress regarding computer software copyrightability.

87. *Id.* at 1.

88. Pub. L. No. 96-517, § 10(b), 94 Stat. 3015, 3028 (1980) (codified at 17 U.S.C. § 117 (1982)). This provision provides that owners of copies of computer programs may use or adapt these copies for their own personal use.

amendment included a definition of the term "computer program,"[89] it did not explicitly provide that software was copyrightable. The legislative history, however, indicates an intent to include software as copyrightable subject matter.[90] Whether these discussions and references to software extend the scope of copyright protection to ROM-embedded object code, however, is unclear.

The 1980 amendment provided no new elaboration regarding the scope of the terms "work of authorship" and "fixed in any tangible medium of expression." Moreover, the CONTU commission itself was divided upon the precise issue of ROM-embedded object code copyrightability.[91] While the majority felt object code was a "work of authorship" qualifying for copyright protection, the dissent cited the lack of "communicative purpose" as a reason to exclude object code from the scope of protection under this standard.[92] The CONTU dissent believed that programmed ROM silicon chips were not intended to communicate with the computer user, and were therefore uncopyrightable mechanical devices, no different from "solid-state circuits of television sets."[93]

D. Object Code Litigation

While CONTU vacillated, Congress deliberately dodged the issue of object code protection, and the matter has ultimately been referred to the federal judiciary. Very few courts have directly addressed the issue of ROM-embedded object code copyrightability. While these courts have enunciated uniform tests to determine

---

89. *Id.* at § 10(a)(codified at 17 U.S.C. § 101 (1982)) ("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.").

90. *See* House Report, *supra* note 9, at 5664:

> The history of copyright law has been one of gradual expansion. . . . [s]cientific discoveries and technological developments have made possible new forms of creative expression that never existed before. In some of these cases the new expressive forms . . . computer programs, for example—could be regarded as an extension of copyrightable subject matter Congress had already intended to protect, and were thus considered copyrightable from the outset without the need of new legislation.

91. *See* CONTU Report, *supra* note 2, at 21. The CONTU majority stated: "Flow charts, source codes, and object codes are works of authorship in which copyright subsists, provided they are the product of sufficient intellectual labor. . . ." The minority countered that object code should not be considered copyrightable because it lacks communicative purpose and is functionally a utilitarian device. For the dissent's view, see *id.* at 32 (statement of Commissioner Hersey).

92. *Id. See also* Koenig, *Software Copyright: The Conflict Within CONTU*, 27 Bull. Copyright Soc'y 340 (1980) (The single area of discord within the CONTU commission was the scope of copyright protection to be given computer programs.).

93. CONTU Report, *supra* note 2, at 32.

copyrightability, divergent results have been reached. Typically, alleged infringers try to avoid copyright protection of source code by exactly duplicating a software developer's corresponding ROM-resident object code version of the program, claiming in court that the ROM is not copyrightable subject matter.[94]

The United States District Court for the Northern District of California first accorded object code copyright protection in *Tandy Corp. v. Personal Micro Computers, Inc.*[95] The plaintiff had developed and registered with the copyright office a ROM-resident object code program for use in its TRS-80 home computer. Defendant produced exact replicas of the plaintiff's ROM silicon chip and marketed them in its PMC-80 computer. When charged with misappropriation of the plaintiff's copyrighted work, the defendant contended that under the *Data Cash* interpretation of federal copyright law, object code was not copyrightable and the duplicated ROMs were not "copies" because the ROM was an unintelligible medium.[96] The district court, however, found that the history of the 1976 Act and Congress' subsequent repeal of section 117 in 1980 clearly indicated a congressional intent to include software within the scope of copyright law.[97] Citing the legislative history, the court declined to apply the *White-Smith* standard and noted that the work's form or fixation medium was not a relevant factor in analyzing the work's copyrightability. Thus, fixation of the object code program on the ROM chip did not preclude copyright protection of the program.[98]

In *Williams Electronics, Inc. v. Artic International, Inc.*,[99] the Third Circuit held the human-intelligibility standard should not be applied to determine a work's copyrightability.[100] Under facts substantially similar to those of *Tandy*, the *Williams* court determined that object code warranted copyright protection.[101] Explicitly re-

94. *See, e.g.*, GCA Corp. v. Chance, 2 COPYRIGHT L. REP. (CCH) ¶ 25,464 (N.D. Cal. Aug. 31, 1982).

95. 524 F. Supp. 171 (N.D. Cal. 1981).

96. *Id.* at 173-74.

97. *Id.* at 174.

98. *Id.* at 173. For a more extensive discussion of *Tandy* and its implications, see Potenza, *Copyright Protection in the Object Code of a Computer Program*, 38 BULL. L. SCI. & TECH. 2, March 1982.

99. 685 F. 2d 870 (3d Cir. 1982).

100. *Id.* at 874.

101. Plaintiff sought an injunction preventing defendant from infringing plaintiff's copyright on a ROM-resident object code program which was used in an electronic video game. *Id.* at 870. The court held that the fixation requirement was met through embodiment of the program in an electronic device, although there was no copyright protection for the electronic devices themselves. *Id.* at 874.

jecting the contention that the 1976 Act only protects works embodied in a medium directly communicable to human beings, the Third Circuit noted that the statutory "fixation" requirement is satisfied if the work can be "reproduced, or otherwise communicated."[102] The court noted that ROM-resident object code programs may be indirectly perceived by the computer user "with the aid of a machine or device."[103] Thus, applying a literal interpretation of the 1976 Act, the court held that ROM-resident object code constituted copyrightable subject matter.[104]

In *Apple Computer, Inc. v. Franklin Computer Corp.*,[105] however, the district court reached the opposite result.[106] The court denied the plaintiff's request for a preliminary injunction restraining the defendant from marketing its "Apple compatible" ACE 100 microcomputer.[107] The ACE 100 featured replicas of the ROM-embedded object code marketed by the plaintiff in its Apple II microcomputer.

The defendant challenged the copyright protection of the plaintiff's ROM-embedded program on two grounds. It was first alleged that copyright protection is limited to material that has underlying expressive or communicative purpose.[108] The district court agreed with this interpretation of the Copyright Act and characterized the ROM-embedded object code as a functional hardware piece for the purpose of operating and synchronizing the microcomputer's processing of the source code.[109]

The defendant's second contention was that ROM-resident object code is a process or idea, uncopyrightable under *Baker* and section 102(b) of the Copyright Act.[110] Alternatively, the defendant contended that object code is a utilitarian or mechanical device not qualifying for copyright protection because it is not a fixed expression or "work of authorship."[111] Accepting most of the defendant's argument against object code copyrightability, the district court denied the plaintiff's motion for injunctive relief.[112] In support of its decision, the court noted that ROM-embedded object code can only

102. *Id.* at 873.
103. *Id.*
104. *Id.* at 878.
105. 545 F. Supp. 812.
106. *Id.* at 825.
107. *Id.*
108. *Id.* at 823.
109. *Id.* In his dissenting opinion in the CONTU Report, Commissioner Hersey agreed with this characterization. *See* CONTU REPORT, *supra* note 2, at 32.
110. 545 F. Supp. at 823.
111. *Id.* at 821-23.
112. *Id.* at 825.

be directly read by an expert using a microscope and is designed and used to control the operation of the computer rather than to convey expressions to humans.[113] Thus, ROM-embedded object code was more accurately described as "an essential part of the machine that makes it work," rather than a copyrightable "explanation" or "expression."[114]

On appeal however, the Third Circuit provided the most comprehensive justification to date for object code copyrightability.[115] In reversing the district court, the Third Circuit determined that a computer program, whether in source or object code form, is a "literary work" protected by the copyright law from unauthorized duplication.[116] The court rejected the defendant's argument that the ROM-embedded object code was an uncopyrightable idea, noting that since other object code programs could be written to perform the same functions, Apple's object code was merely one expression of the idea, and was therefore subject to copyright protection.[117]

The court also rejected the defendant's contention that object code was a utilitarian or mechanical device, holding that the fact that object code was etched on a ROM silicon chip did not make the program a machine part. In the view of the court, the defendant's argument mistakenly focused on the physical characteristics of the object code instructions rather than the message conveyed by the instructions themselves.[118] The Third Circuit reaffirmed its holding in *Williams* that the medium of embodiment is an insufficient basis for disqualifying a work from copyright protection. Thus, the defendant's claim that a ROM-resident object code program's medium rendered it an uncopyrightable idea was inconsistent with its concession that source code was a copyrightable expression.[119]

---

113. *Id.* at 821.

114. *Id.*

115. 714 F.2d 1240. For an enlightened discussion of the *Apple* appeal and its implications for computer copyright law, see Raysman & Brown, *Major Computer Software Issues Resolved in Programming Ruling*, NAT. L.J., Sept. 19, 1983, at 50, col. 1.

116. 714 F.2d at 1240.

117. *Id.* at 1253.

118. *Id.* at 1251.

119. *Id.* at 1252. The Third Circuit noted:

> Perhaps the most convincing item leading us to reject Franklin's argument is that the statutory definition of a computer program as a set of instructions to be used in a computer in order to bring about a certain result . . . makes no distinction between application [source code] programs and operating [object code] programs. Franklin can point to no decision which adopts the distinction it seeks to make.

IV. THE SCOPE OF COPYRIGHT PROTECTION FOR ROM-EMBEDDED OBJECT CODE

Beginning with *Tandy* and continuing with *Williams* and *Apple*, an emerging trend affirming object code's status as copyrightable subject matter is evident.[120] There are, however, two recurring objections to object code copyrightability. The first is the argument that ROM-embedded object code is a "utilitarian" or mechanical device, rather than a product of authorship.[121] A utilitarian device, like a machine part or any other useful article, performs a functional task and is not copyrightable because its physical embodiment does not convey information.[122] The second is the contention that ROM-embedded object code lacks "communicative purpose."[123] Whereas source code is designed to communicate directly with the user, object code interacts with the user only indirectly through the source code.[124] Thus, courts deciding on this contention must focus on whether the Copyright Act requires that a work be created for the purpose of direct rather than indirect communication with humans.

A. Copyright Protection of an Idea?

The scope of works protected by the Copyright Act is expressly enunciated in section 102(a).[125] More significant for the purposes of determining the scope of software protection, however, is section 102(b), which sets forth the scope of unprotected works. Section 102(b) codifies the principles of *Baker v. Selden* and provides that the scope of "authorship" does not extend to an idea, process, procedure, system or operation method regardless of the embodiment form.[126]

This longstanding *Baker* prohibition is typically invoked to at-

120. *See* MacGrady, *supra* note 4, at 1044 (these cases may signal a new trend toward copyright protection of object code programs. *Apple* should provide security to software developers in an area where there has been little guidance from the judiciary.).

121. *See* Stern, *Another Look at Copyright Protection of Software: Did the 1980 Act Do Anything for Object Code?*, 3 Computer L.J. 1 (1981). A mechanical device may not be characterized as a "writing" or an "original work of authorship," and therefore may not be copyrightable. *Id.* at 4-5.

122. The Copyright Act denies copyright protection to utilitarian aspects of articles. *See* 17 U.S.C. §§ 101, 102(b) (1982). *See also* Muller v. Triborough Bridge Auth., 43 F. Supp. 298 (S.D.N.Y. 1942) (bridge not protected by copyright law merely because it was depicted in copyrighted plans and drawings).

123. *See* Stern, *supra* note 121, at 13 n. 49 ("human intelligibility is a predicate for finding something to be a 'copy' ").

124. *See supra* notes 19-23 and accompanying text.

125. 17 U.S.C. § 102(a) (1982).

126. 17 U.S.C. § 102(b) (1982). *See* Nimmer, *supra* note 58, § 2.18[D] at 2-207 (it is

tack copyright protection where the alleged infringer claims object code is an idea, mathematical procedure or process which facilitates the functional operation of the hardware.[127] The test enunciated by the Third Circuit in *Apple* to determine if a particular object code computer program constituted a process or idea was whether other programs could be written which performed the same function.[128] Copyright protection does not preclude alternate expression of the idea used by a program if there are other ways to organize a set of object code commands to bring about a result similar to that of the protected program. If a competitor can create a substantially different ROM-embedded object code program which achieves the same result as the copyrighted ROM-embedded object code program, copyright protection does not monopolize the idea.

With few exceptions, a programmer has several alternatives in developing a program.[129] The probability that copyright protection will serve to monopolize the idea is inversely proportional to the intricacy of the program itself. The more complicated a program becomes, the more avenues there are available to structure the program to achieve a certain result.[130] There exist a virtually unlimited number of instruction sequences that would enable a programmer to construct a program which performs even the more basic algorithmic or mathematical procedures.[131]

In *Apple*, the defendant failed to establish that copyright protection of the plaintiff's ROM-resident object code program precluded the defendant from creating a substantially different yet compatible computer program. Indeed, the record reveals that the defendant did not even attempt to formulate its own version of the plaintiff's object code.[132]

---

apparent that § 102(b) codifies a substantial part of the holding and dicta of Baker v. Selden, 101 U.S. 99 (1879)).

127. *See, e.g., Apple*, 545 F. Supp. 812 (defendant claimed ROM-resident object code was an uncopyrightable idea under *Baker*).

128. "If other programs can be written or created which perform the same function as Apple's operating system program, then that program is an expression of the idea and hence copyrightable." 714 F.2d at 1253.

129. *See* MacGrady, *supra* note 4, at 1035 ("Except in no-logic cases such as those involving mathematical formulae—where there is only 'one way to do it'—a coder typically faces several alternatives in implementing a given series of design steps.").

130. *See* Stern Elecs., Inc. v. Kaufman, 669 F.2d 852, 855 (2d Cir. 1982) ("many different computer programs can produce the same 'results' ").

131. *See* CONTU REPORT, *supra* note 2, at 20 n. 106 ("The availability of alternative non-infringing language is the rule rather than the exception.").

132. 714 F.2d at 1253. In the district court, Franklin justified its actions by claiming that in order to give non-Apple owners the opportunity to take advantage of the Apple-compatible material that exists in the marketplace, Franklin created an object code program "which must of necessity share a great deal of the . . . structure of Ap-

B. Copyright Protection of a Machine Part?

In defending its exact duplication and use of Apple's product, Franklin also argued that a grant of copyright protection to Apple's ROM would constitute an extension of monopolistic protection over a machine part.[133] According to the defendant, copyright was being applied to provide the plaintiff with the equivalent of a patent monopoly over its ROM chip without subjecting the ROM to the more rigorous standards of patentability.

This argument misconstrued the scope of protection sought by the plaintiff. The plaintiff did not seek to impair its competitors' use of ROM silicon chips, but merely sought to preclude misappropriation of its own object code programs embedded on the chips.[134] Arguments that ROM-resident object code is not copyrightable because it is a utilitarian or mechanical device are frequently based upon the *Baker* doctrine.[135] In *Williams*, the defendant claimed there could be no copyright protection for ROM silicon chips because they were essentially mechanical hardware parts.[136] In rejecting this argument, the *Williams* court noted that mere etching of object code on a ROM chip which forms part of the computer hardware does not make the object code itself a machine part.[137] In other words, the medium - the ROM silicon chip - must be distinguished from the message - the object code program.[138]

It is essential that courts recognize that the medium is not copyrightable, and that utilization of a ROM in a computer does not render the ROM itself copyrightable.[139] A ROM may be utilized to house instructions which do not constitute a "computer program."[140] For example, a microcomputer may utilize ROM-embedded microcode or firmware, which is essentially hardwired circuitry converted to ROM. In such instances, the ROM is a utilitarian, mechanical part of the computer hardware, without any programmed

---

ple's operating system." Franklin contended that it had "designed" an Apple-compatible microcomputer, whereas Apple claimed Franklin had "stolen" the logic and structure of its system. 545 F. Supp. at 814-15.

133. 714 F.2d at 1250.

134. *Id.* at 1253.

135. *See supra* notes 121-122 and accompanying text.

136. Williams Elecs., Inc. v. Artic Int'l, Inc., 685 F.2d 870, 874 (3d Cir. 1982).

137. *Id. See also* Tanenbaum, *supra* note 17, at 10 ("[T]he essence of software is the set of instructions that make up the programs, not the physical media on which they are recorded.").

138. 685 F.2d at 876.

139. *But cf.* 17 U.S.C. § 115 (1982) (extends copyright protection to "phonorecords").

140. *See* Harris, *Apple Computer, Inc. v. Franklin Computer Corp.—Does A ROM A Computer Program Make?*, 24 Jurimetrics J. 248, 250 (1984).

software instructions, and thus void of any copyrightable "authorship" characteristics. Arguably, when the ROM is used for such purposes, the most appropriate type of protection is patent law, which protects physical devices, rather than copyright law, which protects works of authorship.

The question becomes where to draw the line between a copyrightable computer program and a patentable utilitarian device. The answer approaches an almost philosophical discussion of where the physical and abstract meet. Courts, most recently the Third Circuit in *Apple*, have, understandably, been unwilling to resolve this dilemma. The unfortunate result of this judicial avoidance of the issue is the lack of meaningful guidelines for distinguishing between a copyrightable ROM-resident computer program and an uncopyrightable, but possibly patentable, ROM-resident hardware circuitry pattern.[141]

C. Purpose of Communicating with Humans Required?

Opponents of object code copyrightability argue that the purpose of a copyrightable work's creation must be direct communication with humans.[142] Rather than engaging in a direct dialogue with the user, ROM-resident object code directly instructs and manages the internal hardware functions and facilitates source code operation. Thus, the argument is made that ROM-embedded object code lacks the requisite communicative purpose because it is fixed in an unintelligible medium and is created for the purpose of communicating directly with the hardware rather than with the user.[143]

Litigants have not hesitated to challenge object code copyrightability under the *White-Smith* human-intelligibility standard despite enactment of superseding modifications in the 1976 Act.[144] In *Tandy*, however, the district court refused to accept a misappropriator's claim that a duplicated ROM silicon chip was not an infringing "copy."[145] The defendant argued that the ROM chip was not copy-

141. *Id.*

142. *See supra* text accompanying notes 123-24. *See also* Note, *supra* note 35, at 659 ("The major legal vice of object code is that a human being cannot read the language, and therefore the code lacks the 'communicative purpose' deemed essential to copyrightability."); Synercom Technology, Inc. v. University Computing Co., 462 F. Supp. 1003, 1011-12 (N.D. Tex. 1978) (material must express ideas to the user rather than simply facilitate his task in order to be copyrightable). *But see*, *Apple*, 714 F.2d 1240 (rejecting the requirement of intelligibility to humans); *Williams*, 685 F.2d 877 (software need not be intended as a medium of communication).

143. *See supra* text accompanying notes 84, 93, 113.

144. *See supra* text accompanying notes 93, 113.

145. Tandy Corp. v. Personal Microcomputers, Inc., 524 F. Supp. 171 (N.D. Cal. 1981).

rightable subject matter because it was not perceptible without the aid of a machine.[146] Basing its decision on the language of the 1976 Act, the court decided that a silicon chip is a "tangible medium of expression" so that a program fixed in this form is copyrightable.[147] The court stressed that imprinting a computer program on a silicon chip, which allows the computer hardware to comprehend the program and act upon its instructions, met the requirement of fixation in any "stable form."[148]

Both the language and the legislative history of the 1976 Act indicate that the communicative requirement should not be interpreted to mandate direct communication with the user as a prerequisite to copyrightability.[149] The statute itself allows object code programs to be communicated directly or "with the aid of a machine or device."[150] The legislative history provides that a work's medium of fixation is irrelevant and merely requires that the work be embodied in a "stable form" to be eligible for copyright protection.[151]

CONCLUSION

The practices currently utilized to secure proprietary protection

---

146. *Id.* at 174.

147. *Id.* at 173. The House Report accompanying the 1976 Copyright Act amendments also supports this view. *See* HOUSE REPORT, *supra* note 9 and *infra* note 149. *But cf.*, Stern, *supra* note 121, at 12, where the author states:

> If anything, the 1980 Act merely reinforces the view that copies of programs must be in intelligible form for them to be protected under the copyright laws. . . . [N]othing in the 1980 statute states that all computer programs are potentially copyrightable. There is no express reference in the statutes or legislative history to the copyrightability of object code as distinguished from that of computer programs generally.

148. 524 F. Supp. at 173. *See also infra* note 153.

149. *See* HOUSE REPORT, *supra* note 9, at 5667. The Report provides:

> The term "literary works" does not connote any criterion of literary merit or qualitative value: it includes . . . computer data bases and computer programs to the extent that they incorporate authorship in the programmer's expression of original ideas, as distinguished from the ideas themselves.

150. 17 U.S.C. § 102(a) (1982) provides:

> Copyright protection subsists . . . in original works of authorship fixed in any tangible medium of expression, now known or later developed, from which they can be perceived, reproduced, or otherwise communicated, either directly or with the aid of a machine or device. Works of authorship include the following categories: (1) literary works. . . .

See also supra notes 70-72 and accompanying text.

151. *See* HOUSE REPORT, *supra* note 9, at 5665. The Report provides:

> Under the bill it makes no difference what the form, manner, or medium of fixation may be - whether it is in words, numbers, notes, sounds, pictures or any other graphic or symbolic indicia, whether embodied in a physical object . . . or in any other stable form, and whether it is capable of perception directly or by means of any machine or device. . . .

of computer programs are insufficiently suited for the rapidly expanding software market.[152] Trade secrecy agreements must be procured on an individual basis, and when used to protect a mass marketed product are difficult and expensive to enforce.[153] Physical "black box" barriers to misappropriation of ROM-resident object code do not always prevent "reverse engineering," deciphering and exact reproduction by misappropriators.[154] Copyright law provides a narrower scope of protection against substantially similar reproduction, but fosters development of creative works by assuring lasting and inexpensive protection despite widespread dissemination in a mass market.[155]

Recent expansion of the scope of copyrightable subject matter has made more effective avenues of protection available for all computer software forms. Attempts to distinguish a copyrightable source code program from its object code counterpart contradict the plain meaning and legislative history of the current Copyright Act.[156] The developing body of object code litigation has determined object code to be copyrightable subject matter. While recognition of the propriety of such protection is welcomed, the federal judiciary still must act to provide meaningful guidelines for distinguishing copyrightable ROM-embedded object code from uncopyrightable utilitarian uses of ROM-embedded hardware.

*Anderson L. Baldy III**

152. *See supra* notes 33-50 and accompanying text.
153. *See supra* notes 43-50 and accompanying text.
154. *See supra* notes 35-36 and accompanying text.
155. *See supra* notes 46-56 and accompanying text.
156. *See supra* notes 147-49 and accompanying text.

* Student, University of Florida.