Legal Rights in Computer Software

David C. Tunick

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

Recommended Citation
Available at: http://digitalcommons.law.scu.edu/chtlj/vol8/iss2/4

This Article 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.
LEGAL RIGHTS
IN
COMPUTER SOFTWARE

David C. Tunick†

There will come a time in the practice of many attorneys when for the first time they will represent a vendor or a user in the acquisition of computer software. The legal issues are complex, and the attorney uninitiated in the world of computers should tread carefully. This article will discuss some of the issues.

SOFTWARE DEVELOPMENT

In situations where a potential user of software contacts a vendor to write custom software (software developed especially for the user), it is typical for employees of both the vendor and user to work together to develop the software. The user will tell the vendor what is needed, and the vendor will write the programs (software). Several legal issues may arise with the writing of the software in these situations. These issues include: a) the potential disclosure of trade secrets from one side to the other, and the need for assurance that the secrets will not be disclosed further; b) ownership of the copyrights in the software; and c) ownership of new ideas developed during the discussions.

A. Disclosure of Trade Secrets

Trade secrecy is a matter of state law and is used to protect business ideas which are secret.¹ A common definition of trade secret is: "Any formula, pattern, device, plan, or compilation of information which is used in one's business, and which gives him an

---


407
opportunity to obtain an advantage over competitors who do not know it."® "As a creature of state statute or common law it differs somewhat from state to state."® Thus, those having trade secrets must guard them according to the laws of each state.® Typically, this means disclosing the secrets to employees or others on a need-to-know basis only. If the secrets are not guarded closely, they may be lost, and others could use them freely.®

One example of a trade secret which a user may disclose to a vendor during the software development is the user's customer list. An example of a disclosure that a vendor may make to the user is the vendor's creation of a tricky way for one of its programs to sort data. The vendor may need to disclose this to the user, but would like to keep the sort routine a secret.

So, each side needs to protect itself from any further disclosure of its secrets. It therefore is a good idea to include a confidentiality clause in the contract between the vendor and user. The clause also should require that each employee who is expected to learn the secrets of the other company also sign a secrecy statement. This will cause the employees to be aware of the secret nature of the work, and the need not to disclose any secrets.

A sample clause which protects secrets of both vendor and user is:

During performance of this contract, each party will provide the other with confidential information. Each party agrees to keep the information confidential, and to disclose it only to employees requiring the information while performing under this contract. Each such employee shall sign an agreement not to disclose the information except in performance of this contract. The confidentiality required by this clause shall not apply to information which is not confidential and proprietary. Any information already known to the other party, or which is learned other than in performance of this contract, or is known by members of the public, need not be kept in confidence.®

The agreement which the employee may sign is similar to one

---

® Restatement, Torts § 757 cmt. b (1939); Telex Corp. v. IBM Corp., 510 F.2d 894, 928 (10th Cir. 1975), cert. dismissed, 423 U.S. 802 (1975).
® CONTU' supra note 1, at 42-43 n.103.
® CLARENCE H. RIDLEY ET AL., COMPUTER SOFTWARE AGREEMENTS § 1.02(2) (1987).
® See Telex Corp. v. IBM Corp., 510 F.2d at 929.
given later in this article under the section, "Former Employees."\(^7\)

Of course, it also is possible that employees, in addition to those who are expected to learn the secrets of the other company, may inadvertently learn secrets. Therefore, it might be wise to require all employees of both companies to sign secrecy agreements.

B. Rights to New Ideas

During the discussion stages between the user and vendor, new ideas may be developed.\(^8\) These new ideas could involve many aspects of the software, such as the idea to sort data in some novel way, or the idea to write software to allow a user to generate greeting cards using a computer. While it is difficult to know what constitutes a new idea, it is better to own the rights to any potential developments than not. A sample clause which gives the user ownership rights is:

During the performance of this contract, new concepts and ideas may be developed. All of these shall belong to the user. Vendor shall use these only while performing under this contract.\(^9\)

WRITING THE PROGRAMS

Once the vendor and user have decided what the software is to do, the vendor can begin writing the programs. While each vendor and programmer will have his own style of writing programs, there are some programming practices which are common throughout the industry. There are legal concerns attendant to these common practices.

A. Flowcharts

Programmers often write flowcharts prior to writing the pro-

\(^7\) See infra note 78 and accompanying text.

\(^8\) It is possible that as a result of the contractual relationship between the parties new ideas or inventions will emerge, which will not clearly be owned by either party. It is desirable to have in the contract a clause which assigns such new ideas or inventions to one or the other party. Normally, the using organization, as the organization paying for services or products, is the organization which obtains the benefits of any new ideas or inventions generated as part of its contract. However, under certain conditions, such as in the case of a proprietary software package or a proprietary hardware alteration, the rights to the invention or the idea might belong to the vending organization. This is a negotiable point whose results would have to be assessed as part of the negotiation, probably by counsel for both parties.

grams in the language which will be usable by the computer. "A flowchart is a graphic representation for the definition, analysis or solution of a problem in which symbols are used to represent operations, data flow, or equipment."10

Flowcharts are protected by copyright law.11 Unlike the law of trade secrets, which may vary from state to state, copyright law is federal.12 Therefore, a copyright is enforceable throughout the United States.13 Copyright protection exists in literary works, and flowcharts are considered literary works.14 Although the vendor may draw, and thus be considered the creator of the flowcharts, the parties can make an agreement as to who owns the copyright.15 If the parties do not agree, the vendor will own the copyright by default.16

If the vendor retains the copyright to the flowcharts, the vendor should be aware that the user may have access to the flowcharts. Conceivably, the user could copy parts of the flowcharts. This would allow a user who is writing programs to incorporate parts of the vendor's flowcharts into his own programs. The vendor would then have a legal right to enforce the copyright against the user. If, on the other hand, the user gets the copyright to the flowcharts, the vendor cannot legally draw substantially similar flowcharts for a subsequent job.17

It is a copyright violation to copy the flowcharts even if they have not yet been registered with the Copyright Office.18 The copyright law has been changed in the last few years so that registration is not a requirement to obtain copyright protection. Rather, copy-

10. CONTU, supra note 1, at 53 n.126.
12. CONTU, supra note 1, at 40-47.
13. CONTU, supra note 1, at 45.
15. 17 U.S.C. § 201(d)(1) (1992) allows the copyright in a work to be conveyed. The statute provides: "The ownership of a copyright may be transferred in whole or in part by any means of conveyance. . . ."
18. Section 408 allows registration at any time during the subsistence of the copyright. However, one needs to register the copyright in order to bring an action for infringement. 17 U.S.C. § 411 (1992).
right protection exists from the moment of creation. Presumably, this means that copyright protection exists as the programmer begins drawing the flow chart, and continues to exist as more steps are added.

One word of caution: even though it violates the copyright law to copy flowcharts, legally one could view them and learn the underlying ideas. Then that person could draw a different flowchart based on the same ideas, and not violate copyright law. This is true, because ideas are not entitled to receive copyright protection; only expressions of ideas are protected. Therefore, it is useful to have a confidentiality agreement signed to protect against use of the ideas involved in a flowchart. The ideas could be trade secrets. To be certain the other side does not use or disclose the secrets, an agreement to that effect is wise.

Below is an example of what a flowchart looks like. It should be noted that copyright protection will not extend to the expression of an idea if that idea can be expressed in only a limited number of ways, since that would, in effect, give copyright protection to the underlying idea. Since the illustrated flowchart may be one of only a limited number of ways to accomplish its purpose, it is quite possible that it would not qualify for copyright protection. However, the most effective way to demonstrate a flowchart is a simplistic example.

This flowchart shows how a program might be written to cause the computer to do the simple task of inputting two numbers to the computer, adding them together, and outputting the result. The task will begin again, if the computer user so desires. Otherwise, the program will return to the operating system.

---


20. 17 U.S.C. § 101 (1992) defines "created": "A work is 'created' when it is fixed in a copy or phonorecord for the first time; where a work is prepared over a period of time, the portion of it that has been fixed at any particular time constitutes the work as of that time. . . ."

21. This could, however, violate trade secret rights.


23. Lotus, 740 F. Supp. at 59. Research has failed to locate the distinction between a "limited number of ways" and what constitutes more than a "limited number of ways." See 3 MELVIN B. NIMMER & DAVID NIMMER, NIMMER ON COPYRIGHT § 13.03[B][3] (Release No. 29, 1991), where the concept is mentioned but not defined.

B. Higher-Level Code

Programmers often draw the flowcharts first, and then write the programs in higher-level language based on the flowcharts. A higher-level language enables a programmer to write the program in a language which looks like English. Once the program is written in the higher-level language, it needs to be translated into binary to operate in the computer. A program called a "compiler" will translate the program into binary. These higher-level languages can be used for nearly any computer, since compilers exist for translating nearly any higher-level program into binary for nearly any

---

programs usually perform a specific task for the computer user, such as word processing, checkbook balancing, or playing a game. *Id.*


27. *Id.*
computer. Most programs are written in higher-level language.

Copyright protection extends to programs written in higher-level language. This is called "source code" and is protected as a literary work. Assuming the vendor owns the copyright to the software, if a programmer leaves the higher-level language coding sheets on her desk and they are seen by another and copied, the copying would be a copyright violation. However, as with flowcharts, one could read the higher-level language and learn the underlying ideas. Then, using the underlying ideas, the person could write a different program and not infringe the copyright. Therefore, if the only protection for the program were copyright, the ideas could not be protected. However, a confidentiality agreement relating to the underlying ideas can be used to prevent the potentially expensive loss.

The underlying ideas would be considered trade secrets, and protected as trade secrets if the proper precautions are taken, such as confidentiality agreements. As was true earlier with flowcharts, if the vendor and user agree that the user gets the ownership of the copyright to the higher level code, the vendor cannot legally write substantially similar code. However, unless the user has rights to the underlying ideas, the vendor could write different programs using these same ideas.

As an example of what higher-level code looks like, here is a program written in BASIC. This program is written for the IBM Personal Computer and is the BASIC program corresponding to the flowchart to add two numbers, and display the result.

10 CLS: REM Clear screen
20 C0=0: REM For summing entries
30 FOR A=0 TO 1
40 INPUT "NUMBER=", TEMP: REM Get input
50 C0=C0+TEMP: REM Place running total into C0
60 NEXT A: REM End loop
70 PRINT "Answer="; C0: REM Print result

28. Id.
29. Id.
31. 17 U.S.C. § 106 (1992) allows copying only by the owner of the copyright; except that under 17 U.S.C. § 117 (1992), one is allowed to read a computer program into a computer, and also make a backup copy. Reading the program into the computer is considered making a copy. CONTU, supra note 1, at 31.
33. BASIC is an acronym for Beginner's All-Purpose Symbolic Instruction Code.
C. File Structures

When programmers write programs, they need to use some of the memory space in the computer, or some other storage media to save data, such as disks. A file structure is "a storage place for data, and it's really no different in a computer than it is in a file drawer; it's like a manila folder that contains all the data on a particular subject category in a computer." One court has said: "[a]nother analogy, particularly accessible to lawyers, is to a very complex cataloging structure like the structure of Lexis or Westlaw without any entries yet made." As an example, a program may allocate 1,000 memory locations for names, the next 1,000 for addresses, the next 1,000 for birth dates, and so on.

In Whelan Associates, Inc. v. Jaslow Dental Laboratories, the court dealt with whether file structures are protectable by copyright. In Whelan, the developer of computer software for dental laboratory record-keeping brought a copyright infringement suit against the dental laboratory for whom the software was created. Whelan claimed the laboratory wrote infringing software. The court concluded that file structures were copyrightable.

Recall that copyright protection for any copyrightable material, including file structures, will not be available for the expression of an idea if that idea can be expressed in only a few ways. However, if there are many ways in which the file structures can be organized, they may receive copyright protection.

---

35. Id. at 1242.
37. Id. at 1224-27.
38. Id. at 1242-43.
40. Whelan, 797 F.2d at 1242-43.
INFRINGEMENTS AFTER SOFTWARE ACQUISITION

Until this point in the article, the infringements discussed parties working together and involve the precautionary steps for the owner of the copyright and trade secrets may take in relation to the other party. When working together, each could learn the secrets of the other. Also, the user could view the flowcharts and higher-level code of the vendor; and, assuming the vendor kept the copyrights, the user could infringe by copying the work in progress. Of course, there also can be infringements when software is purchased or licensed, and the parties have not worked together.

A. Copying the Binary Program

Once the programming vendor has written a program in higher-level language, the program is translated by a compiler program into binary. It is the binary version of the program which is understood by, and operates, the computer. Frequently, the binary program is copied onto disks\textsuperscript{41} by the vendor, and then sold or licensed. The binary version of the program, as well as the higher-level source code, is copyrightable.\textsuperscript{42}

Limited copying of software is authorized by the copyright law.\textsuperscript{43} Copying is allowed by someone who owns a copy of the program, so that it may be read into the computer in order to be used and also so that an archival copy may be made.\textsuperscript{44} If there is a license to use software, it would be advisable to have the license specify exactly to what extent the licensee may copy. For example, since the copyright law literally allows only the owner of the software to make an archival copy, the licensee of software is advised to have an agreement allowing the making of an archival copy. An archival copy is needed in case the disk delivered to the user becomes unusable. It should be noted that one who obtains a copy of software does not thereby acquire the copyright to the software any more than one who obtains a copy of a book or movie. Therefore, the copying that is permitted must be authorized by the

\textsuperscript{41} The binary program also can also be transmitted on telephone lines. Commerce Union Bank v. Tidwell, 538 S.W.2d 405, 408 (Tenn. 1976).
\textsuperscript{44} "[I]t is not an infringement for the owner of a copy of a computer program to make... another copy provided: (1) that such a new copy... is created as an essential step in the utilization of the computer program in conjunction with a machine..., or (2) that such a new copy... is for archival purposes only..." Id. CONTU suggests that rightful possessors of programs be allowed to input them into the computer. CONTU, supra note 1, at 31.
copyright law or by the owner of the copyright in a license or other agreement.

One interesting aspect of the copyright law which relates to copying a binary program from a disk or some other form of storage device, and that is not discussed in the cases in much depth, is what must be actually seen or perceived.\textsuperscript{45} Section 102(a) of the Copyright Act provides:

Copyright protection subsists, in accordance with this title, 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.

The binary code resides on the storage media. There are computer programs which can cause this binary code to be printed on the printer so that it can be perceived in binary or another number system. However, the programmer did not write in that system, but rather in a higher-level language. So, the higher-level language probably would be considered the original work of authorship. It appears from a literal reading of § 102(a) that it is the original work of authorship which needs to be perceived.\textsuperscript{46} However, § 103 says that the “subject matter of copyright as specified by § 102 includes ... derivative works ....” A “derivative work” includes translations.\textsuperscript{47} It then could be argued that the derivative work, the binary code, is a translation of the higher-level language and copyrightable as an original work of authorship under § 102. Therefore, when the binary code is printed and perceived, § 102 is satisfied, because a derivative work is perceived.

B. Screen Displays

Screen displays\textsuperscript{48} are copyrightable as audio-visual works.\textsuperscript{49}

\textsuperscript{45} For example, Apple Computer, 714 F.2d at 1247 quotes § 102(a), but uses the section only to show that the work must be original and fixed in a tangible medium of expression. It does not discuss perceiving what is fixed (see id.), and § 102(a) requires that the copyrighted work be perceived.

\textsuperscript{46} Without getting into all of the technicalities, even using a “reverse compiler” to generate the higher-level language from the binary code may not reproduce the original higher-level language in all details. For example, while reverse compilers can regenerate the syntax of the higher-level language, most will be unable to regenerate the original names of items, tables, arrays, and statement labels.


However, there are several aspects of screen displays that need to be discussed.

i) Typically, copyright infringement can exist even without identical copying; substantial similarity is enough. However, if the underlying idea can be expressed in only a limited number of ways, substantial similarity might not cause infringement. In a case involving the PAC-MAN game, the 7th Circuit said:

Certain expressive matter in the PAC-MAN work . . . should . . . receive protection only from virtual identical copying. The maze and scoring table are standard game devices, and the tunnel exits are nothing more than the commonly used "wrap-around" concept adapted to a maze/chase game. Similarly, the use of dots provides a means by which a player's performance can be gauged and rewarded with the appropriate number of points, and by which to inform the player of his or her progress. Given their close connection with the underlying game, K.C. Munchkin's maze design, scoring table, and "dots" are sufficiently different to preclude a finding of infringement on that basis alone.

And in a case involving spreadsheets, the United States District Court in Massachusetts said:

In both Lotus 1-2-3 and VP Planner, as in many other electronic spreadsheet programs, a highlighted element of the basic screen display resembles an "L" rotated ninety degrees clockwise with letters across the top to designate columns, and numbers down the left side to designate rows.

[If a previous programmer's idea can be expressed in only one of a limited number of ways — such as the rotating "L" screen display . . . then the expressions too may be copied . . . where, in contrast, the idea is capable of countless ways of being expressed, only inexpensive cloning, and not innovation, would be advanced by allowing programmers to copy the particular way

the ideas have been expressed by others.  

Thus, if a company acquires software on a binary storage medium, such as a disk, and the company wants to write its own software to perform similar functions, the company can have similar screen displays if those displays are common or are one of only a few ways of expressing the ideas.

ii) Screen displays which are uncommon receive broader copyright protection, and will be protected from infringement by displays which are substantially similar. In the PAC-MAN case, the court said:

[I]t is the substantial appropriation of the PAC-MAN characters that requires [a probable finding of infringement]. The expression of the central figure as a “gobbler” and the pursuit figures as “ghost monsters” distinguish PAC-MAN from conceptually similar video games.

North American not only adopted the same basic characters but also portrayed them in a manner which made K.C. Munchkins appear substantially similar to PAC-MAN. The K.C. Munchkin gobbler has several blatantly similar features, including the size and shape of the “body,” the V-shaped “mouth,” its distinctive gobbling action (with appropriate sounds), and especially the way it disappears upon being captured. An examination of the K.C. Munchkin ghost monsters reveals even more significant visual similarities. In size, shape and manner of movement, they are virtually identical to their PAC-MAN counterparts. K.C. Munchkin’s monsters, for example, exhibit the same peculiar “eye” and “leg” movement. Both games, moreover, express the role reversal and “regeneration” process with such great similarity that an ordinary observer could conclude only that North American copied ... PAC-MAN.

And in the spreadsheet case, the court said:

[A] menu command structure is capable of being expressed in many if not an unlimited number of ways and, ... the command structure of [Lotus] 1-2-3 is an original and nonobvious way of expressing a command structure ... Accordingly, the menu structure, taken as a whole — including the choice of command terms, the structure and order of those terms, their presentations on the screen and long prompts — is an aspect of 1-2-3 that is not present in every expression of an electronic spreadsheet. It

54. Id. at 78.
55. Atari, 672 F.2d at 616-18.
56. Id. at 617-18.
meets [one of the requirements] of the legal test of copyrightability.\textsuperscript{57}

Therefore, one who acquires software and intends to sell other software to perform similar functions should not design screen displays to be similar, except for those portions of screens which are common methods, or one of only a few methods, of expressing the idea.

iii) One further note on this subject: the copyright law requires that protection exist only on original works of authorship.\textsuperscript{58} There is an argument that because of user interaction with the computer, the screen displays change each time. Therefore, they are not original to the software vendor.\textsuperscript{59} However, this has been rejected in cases involving video games.\textsuperscript{60} The cases have held that all available screen displays have been preprogrammed and merely are selected by the user;\textsuperscript{61} the user merely selects the preprogrammed images, just as a television viewer selects what to watch by changing channels.\textsuperscript{62} In both cases, what appears on the screen exists before the user selects. Thus, the original work rests with the computer programmer, and not the user.

So, if a company is writing software, whether it be games,\textsuperscript{63} business software,\textsuperscript{64} greeting cards,\textsuperscript{65} or other software, the company needs to be certain it is not infringing the screen displays of another. Displays which are common methods of accomplishing the desired result will not receive copyright protection. However, displays representing uncommon methods will receive copyright protection.

C. Sequence of Screens

The sequence of the screens also can receive copyright protection.\textsuperscript{66} In a case between competing software vendors which in-

\textsuperscript{57} Lotus, 740 F. Supp. at 68.  
\textsuperscript{58} 17 U.S.C. § 102(a) (1992).  
\textsuperscript{59} Stern Elecs., Inc. v. Kaufman, 669 F.2d 852, 856 (2d Cir. 1981).  
\textsuperscript{61} Stern, 669 F.2d at 856.  
\textsuperscript{62} Midway, 704 F.2d at 1011-12.  
\textsuperscript{63} E.g., id.  
\textsuperscript{64} E.g., Whelan Assoc., Inc. v. Jaslow Dental Lab., Inc., 797 F.2d 1222, 1231-32 (3d Cir. 1986), cert. denied, 479 U.S. 1031 (1987).  
\textsuperscript{66} Id. at 1133.
volved software for printing greeting cards, signs, banners, and posters, the Second Circuit explained:

Defendant . . . argues that the idea underlying the menu screens, input formats, and sequencing of screens in "Print Shop" is indistinguishable from its expression. Any menu-driven computer program that allows its users to print greeting cards, signs, banners, and posters will have a user interface substantially similar to that in "Print Shop," defendant contends, because there is no other conceivable way to structure such a program. The evidence at trial disproved defendant's contention. Plaintiff introduced a program titled "Stickybear Printers," . . . that allows its users to print greeting cards, signs, banners, and posters with variable combinations of user-dictated text, graphics, and borders. The functions of "Stickybear Printer" are substantially the same as "Print Shop," thus it can be said that the ideas underlying "Stickybear Printer" and "Print Shop" are the same. Yet the expressions of those ideas are very different. The menu screens and sequence of screens in the two programs are very different. The entire structure and organization of the user interface are different. In short, the existence of "Stickybear Printer" proves that there do exist other, quite different ways of expressing the ideas embodied in "Print Shop." 6

When the court discusses "sequence of screens," it apparently refers to the order and choices being presented on the screens. 67 Thus, even if the screen displays were not substantially similar, the sequence in which similar questions were asked and similar information was provided on the screens could show copyright infringement. 68 Once again, it should be noted that if two programs are similar, there will be no copyright infringement unless actual copying has occurred. A programmer who writes a program to do the same tasks as other software in substantially similar ways, but without copying, does not infringe. 69

**Former Employees**

**A. Non-Competition Agreements**

Software vendors, of course, have programmers designing and writing the software. It is not unusual for these programmers to start their own competing businesses. A company may wish to pre-

---

67. *Id.* at 1132.
68. *Id.* at 1136-37.
70. *Id.* at 1135-36.
prevent its programmers from quitting for this purpose. However, it may be difficult to enforce an agreement not to compete. In California, with certain limited exceptions, an agreement under which a person is prevented from engaging in a profession, trade, or business is void. One example of a limited exception relates to the selling of a business and its goodwill where the seller agrees not to compete. Usually, however "[a] former employee has the right to engage in a competitive business for himself and to enter into competition with his former employer, even for the business of those who had formerly been the customers of his former employer. . . ."

B. Trade Secrets

While it may be difficult for a software vendor to prevent its programmers from opening competing businesses, the vendor can prevent the programmers from using the trade secrets of the vendor. These secrets could include customer lists and technical information.

Ideally, the vendor should have the employee sign a confidentiality agreement upon employment. A sample agreement which the vendor may wish to have with the employee programmer reads:

Employee agrees, during employment and thereafter, not to disclose, except as required in performance of employee's duties, or with employer's permission, information provided by employer or otherwise learned in connection with employee's duties. However, employee may disclose information known to the general public or learned other than in performance of employee's duties for employer. Employee understands that wrongful disclosure may result in irreparable harm to employer or another and that money damages may not be an adequate remedy. Therefore, employer may obtain injunction(s) to prevent disclosures, in addition to any other available remedies.

---

72. Monogram, 64 Cal. App. 3d at 697.
74. Id. at 633-35.
75. Id. at 634-36.
76. Telex Corp. v. IBM Corp., 510 F.2d 894, 929 (10th Cir. 1975), cert dismissed, 423 U.S. 802 (1975).
77. BRANDON & SEGELSTEIN, supra note 9, at 154-155, 416.
78. Modeled after a clause in BRANDON AND SEGELSTEIN, supra note 9, at 416.
C. Copyright

If an employee should quit to start her own programming company, that does not mean she can copy the programs she previously wrote. In the absence of an agreement to the contrary, the copyrights in the programs written by an employee within the scope of employment belong to the employer. A programmer would need to change the code, file structures, sequence of operation, and screen displays to be safe from an infringement suit, even though she may have written the original program. And, recall that the trade secrets used in the programs also may not be used by the programmer in her new company.

One note of caution for software companies: if a programmer writes software as an independent contractor, rather than an employee, the programmer owns the copyrights unless otherwise agreed. There are a variety of factors that assist in determining whether a person is an employee or an independent contractor. These factors include: where the work is performed, whether social security or income tax is withheld from pay, whether the pay is per program, whether the programmer works exclusively for the vendor, whether the vendor retains authority to assign additional projects, whether the programmer is hired through the channels the vendor customarily uses for hiring employees, and whether the programmer obtains from the vendor all benefits customarily extended to its regular employees.

COPYRIGHT REGISTRATION AND NOTICE

Copyright protection exists from the moment of creation of the work. Therefore, the software producer does not need to register

82. Id.
83. Dumas v. Gommerman, 865 F.2d 1093, 1105 (9th Cir. 1989).

Prior to the passage of the 1976 Act, there existed a dual system of federal and state copyright law. "Unpublished" works - those in limited distribution and unavailable to the general public - were protected by state common-law copyright. The common-law protection was for an indefinite period, ending only upon publication of the work. Upon publication, a work lost its common-law copyright protection and entered the public domain unless the copyright owner complied with the requirements for federal statutory copyright under the federal copyright act. See Vacheron & Constantin-Le Coultre Watches, Inc. v. Benrus Watch Co., 260 F.2d 637 (2d Cir. 1958), superseded by statute in Nova Stylings, Inc. v. Ladd, 695 F.2d 1179 (9th Cir. 1983). Federal copyright protection was afforded to a work for a period of twenty-eight years, renewable once.
with the Copyright Office in order to secure protection. Registration is required in some situations, such as when pursuing an action for infringement.\textsuperscript{85} The registration must be done within three months of first publication of the work in order to recover attorney’s fees and statutory damages in an infringement action.\textsuperscript{86}

Since the copyright registration rules can change at the discretion of the Registrar of Copyrights,\textsuperscript{87} it is advisable to confer with the Copyright Office in Washington, D.C. to acquire the latest forms and information. For example, presently the Copyright Office requires receipt of the first twenty-five and last twenty-five pages of computer program source code.\textsuperscript{88} This can be on paper or microfilm.\textsuperscript{89} For a program less than fifty pages in length, a visually perceptible copy of the entire program must be sent.\textsuperscript{90} The Copyright Office also can give information on other matters, such as filing fees, and whether and where to place copyright notices when distributing the software.\textsuperscript{91} Although copyright notice no longer is necessary when distributing copyrightable materials,\textsuperscript{92} such as software, it still may be a good idea to include appropriate notice in order to dis-

---

\textsuperscript{85} The 1976 Copyright Act abolished the distinction between statutory and common-law copyright and made the concept of “publication” irrelevant. Section 301(a) of the 1976 Act provides:

On and after January 1, 1978, all legal or equitable rights that are equivalent to any of the exclusive rights within the general scope of copyright as specified by section 106 in works of authorship that are fixed in a tangible medium of expression and come within the subject matter of copyright as specified by sections 102 and 103, \textit{whether created before or after that date and whether published or unpublished}, are governed exclusively by this title. Thereafter, no person is entitled to any such right or equivalent right in any such work under the common law or statutes of any State.

(Emphasis supplied.) The plain meaning of § 301 is that the 1976 Act abolished common-law copyright by preemption.


\textsuperscript{88} Id. Statutory damages between $500 and $20,000 can be awarded as the court deems just, and apparently do not necessarily need to be related to actual losses. 17 U.S.C. § 504(c)(1) (1992).

\textsuperscript{89} 17 U.S.C. § 408(c) (1992).


\textsuperscript{91} Id. at 2, 4.

courage copying as a way to prevent the inconvenience of an infringement lawsuit.

**SOURCE CODE ESCROW**

**A. Delivered Software is in Binary**

Typically, when the software is delivered, it is in binary (computer readable) form on a tape or disk. In order to make corrections, the programmer goes back to the "source code," the language that looks something like English. This is the higher level code. A sample of source code appears earlier in this article under the heading **HIGHER LEVEL CODE**. Frequently, corrections are made to the source code. Then another program, called a compiler, translates the source code into binary.

If the vendor is unable to maintain the software, possibly because its better programmers have quit, or even because the vendor is bankrupt, the user may need to get the source code together with programmer's manuals describing the logic of the programs, and compilers in order to compile the programs from source code into binary. But vendors are reluctant to release source code, because their programming methods could be learned and copied by those seeing the source code. Not only may copying be difficult to prove, but if the user does copy the source code, the vendor might lose its competitive advantage should the user compete with the vendor by selling software. Negotiating with the vendor in order to obtain source code could be a major hurdle.

**B. Vendor's Failure to Repair Software**

The user will want to be able to obtain the source code in order to attempt to make any necessary corrections itself. A user should not be misled into thinking the software will be easy to change just because she gets a copy of the source code. Programmers must understand the language in which the programs are written, and then spend an enormous amount of time figuring out the program logic and making the necessary changes.

---

93. *See supra* text accompanying notes 25-32.
94. Actually, the source code will be needed both on paper, so it can be read, and also on tape or disk. If the user does not receive the source code on tape or disk, the user will need to enter all of the possibly thousands of lines of code from the keyboard. This already has been done by the vendor, so that the source code can be compiled each time from tape or disk, plus the few changes probably being entered from the keyboard.
95. Such as **BASIC** (Beginner's All-Purpose Symbolic Instruction Code), which is used in the sample higher level program, *supra* text accompanying note 33.
A problem arises in drafting the contract language clearly enough so that it is understood when the user shall receive the source code and other required material (e.g., compiler, programmers' manuals). It is not enough to state that the user shall receive these materials if the programs contain a "serious" malfunction which is not repaired within seven days after the user notifies the vendor of the malfunction. Defining "serious" is difficult. Some malfunctions are of greater concern than others, and require quick attention. Specifying which are serious can be difficult, if not impossible. For example, if the date is incorrect on a printout, is this a serious error? If January is abbreviated "Jan" (without a period) rather than "Jan." (with a period), it would not seem to be serious. But, if January is abbreviated "*/,", a serious error probably exists.

A user may not wish to rely on the vendor to supply the necessary materials if the vendor fails to correct errors in a timely fashion. The user and vendor may disagree as to whether the materials should be turned over to the user. For this reason, it may be advisable to have the vendor deposit the materials with a third party, depositing updates as they occur. This is the creation of an escrow. When a predetermined triggering event occurs, such as the failure to correct the software, the holder of the materials releases the materials to the user.

C. Vendor in Bankruptcy

A fear that users often have is that the vendor will be unable to maintain the software because of the vendor's bankruptcy.

1. License Agreement for Software

If the vendor actually is a licensor, rather than a seller, and the user is licensed to use the software, two possibilities exist when bankruptcy occurs.

i) The bankruptcy trustee may reject the license as an executory contract. In this case, under a 1988 Bankruptcy Code revision, it appears the user (licensee) can retain rights to use the

---

97. Id.; RIDLEY, supra note 4, at 2-145-46; BRANDON & HALVEY, supra note 6, at 122.
98. If the trustee rejects an executory contract under which the debtor is a licensor of a right to intellectual property, the licensee under such circumstances may elect — (B) to retain its rights . . . under such contract and under any agreement supplementary to such contract to such intellectual property . . . as such rights existed immediately before the case commenced.
software (delivered originally in binary) and also get the materials from the escrow agent. The new law gives the user/licensee the rights which existed just prior to the commencement of the bankruptcy case.

ii) If the bankruptcy trustee does not reject the licensee’s usage of the software in binary, the result seems to be the same. The licensee can use the software and get the materials out of escrow. Actually, an escrow arrangement need not be created. The licensor and user can agree that if the licensor fails to maintain the software, the licensor will turn over to the user the source code, compilers, and other materials. It appears this will be enforced in bankruptcy.

One word of caution: because the Copyright Code allows only owners of a copy of the software to make needed adaptations, the licensee/user should receive contractual permission to make the adaptations.

2. Sale of Software

If the user had purchased the software, rather than acquired rights under a license, a literal reading of the Bankruptcy Code would not allow the user to get the source code out of escrow. This is true, because the Bankruptcy Code protects licensees, not purchasers. However, this problem may be more imaginary than real. When a user purchases custom software, it is common that the user acquires the copyrights to all software, as well as source code, programmer’s manuals, and user’s manuals. Thus, the user will have the necessary materials to attempt to make corrections.

In the unusual situation where the user purchases custom software, and the vendor wants to deliver only the binary for fear

99. *Id.* See also *Bigelow*, supra note 96, § 10.07(3)[b].
101. Unless and until the trustee rejects such contract, on written request of the licensee, the trustee shall . . . (B) not interfere with the rights of the licensee as provided in such contract, or any agreement supplementary to such contract, to such intellectual property (including such embodiment), including any right to obtain such intellectual property (or such embodiment) from a third entity. 11 U.S.C. § 365(n) (1992).
102. The Bankruptcy Trustee must turn over to the licensee (user) intellectual property specified in the license agreement. 11 U.S.C. §§ 365(n)(1)(B), (3)(A)(B), (4)(A)(B). The intellectual property would be the source code, compilers, and other necessary materials.
104. See * supra* notes 98 & 101.
the user may learn the vendor's programming secrets if the user sees the source code, an escrow may be established. The user would purchase the source code, and have it delivered to a third party, who releases it to the user when the vendor fails to maintain the software. Since the user already owns the source code and other materials, these do not automatically become the property of the estate in bankruptcy.\textsuperscript{106}

D. \textit{Vendor Not in Bankruptcy}

It cannot be predicted why a vendor would be unable to correct errors in the software. It might be because of the vendor's bankruptcy, or because the vendor's skilled programmers have quit. It should be remembered that the user may want the necessary materials if the vendor is not in bankruptcy, but merely has been unable to fix the software. In this case, a user may fear that a disagreement between he and the vendor may cause the vendor to refuse to provide the materials. A user may feel safer if an outside party has the materials and therefore may want to establish an escrow.

CONCLUSION

Those involved in the development and acquisition of computer software need to be aware that protectable rights exist in software. Software companies, their programmers, those hiring software companies to write custom software, and users acquiring canned software all have rights in the product. This article has attempted to show that all parties need to protect their own rights, while not infringing the rights of others.

POSSIBLE COMPUTER CONTRACTING REFERENCES

For those attorneys who will be drafting or reviewing contracts for the acquisition of computer products, the following books may be useful.


Data Processing Contracts: Structure, Contents and Negotiation by George Brandon and John K. Halvey (3d ed. 1990)


\textsuperscript{106} \textit{Bigelow, supra} note 96, § 10.07[3][b].
Computer Contract Negotiations by Auer and Harris (VanNostrand Reinhold Co. 1981)

