First Principles:
Things Technology Transaction Attorneys Should Know

By: Paul H. Arne[1, 2, 3]

This article is about things I believe good technology transaction lawyers should know. Please consider this as just a good start.

1. Works Made for Hire.

There is an interesting distinction between the way business people talk about "works for hire," and the underlying law. The following rule seems to work reasonably well to translate business-speak into legal-speak.

If a businessperson says something to the effect of "that was developed as a work for hire," you can readily translate into correct legal terms by adding the word "not." In other words, "that was **not** developed as a work for hire."

The US Copyright Act, in Section 101, defines a work made for hire as follows:

> A "work made for hire" is—
>
> (1) a work prepared by an employee within the scope of his or her employment;
>
> or
>
> (2) a work specially ordered or commissioned for use as a contribution to a collective work, as a part of a motion picture or other audiovisual work, as a translation, as a supplementary work, as a compilation, as an instructional text, as a test, as answer material for a test, or as an atlas, if the parties expressly agree in a written instrument signed by them that the work shall be considered a work made for hire. …

When business people refer to a "work for hire," the particular work in question is almost never work developed by an employee of a company that is the recipient of the work. The reference is almost always to a third party developer or software provider who develops a particular piece of software and provides to another company. Therefore, subsection (1) of this definition, which is about employees, does not typically apply to what the businessperson is saying.

---

[1] Copyright © Paul H. Arne, 2022. All rights reserved.

[2] Paul is the chair of the Technology Transactions Group of Morris, Manning & Martin, L.L.P.

[3] This article does not create an attorney/client relationship with you and does not provide specific legal advice to you or your company. Certain legal concepts have not been fully developed and certain legal issues have been stated as fact for which arguments can be made to the contrary, due to space constraints. It is provided for educational purposes only.

Turning to subsection (2), it requires that each work be "specially ordered or commissioned." There is also a list of nine items that qualify as a work made for hire, assuming the remainder of the statute also applies. These nine items are not a list of examples that describe a more general concept. In order to apply the work for hire doctrine to a particular piece of software, the software must fit into one of these nine categories.

Software is not typically a "translation," an "instructional text," a "test," "answer material for a test," or an "atlas." This leaves works that are "a contribution to a collective work," "a part of a motion picture or other audiovisual work," a "supplementary work," and a "compilation." These categories will be examined separately below.

A. Collective Works

The Copyright Act defines a "collective work" as a

> work, such as a periodical issue, anthology, or encyclopedia, in which a number of contributions, constituting separate and independent works in themselves, are assembled into a collective whole.

Software does not seem to fit well with any of the examples in the statute's text. Even though modern software is usually developed as a series of executable objects, rather than a single, monolithic executable, typically the separately executable components, or libraries, are not designed to be usable outside of a software program that uses the library, suggesting that software libraries are not intended to be "separate and independent." Further, these additional executable objects are rarely "specially ordered or commissioned" for the specific software in question. Accordingly, software should typically not be considered a collective work.

B. Audiovisual Works

Outside of gaming software, most software is clearly not an "audiovisual" work.[4] Software is also considered a "literary work" under the Copyright Act, rather than an audiovisual work. Accordingly, neither the customary way business people identify software nor the definitions in the Copyright Act would classify software as an "audiovisual" work. Unless you are dealing with the development of a software game, it is very unlikely that software will be considered an audiovisual work.

C. Supplementary Work

"Supplementary Work" is defined in the definition of "work made for hire" itself.

> [A] "supplementary work" is a work prepared for publication as a secondary adjunct to a work by another author for the purpose of introducing, concluding, illustrating, explaining, revising, commenting upon, or assisting in the use of the other work, such as

[4] Sometimes, the output of software, most notably screen displays, may be separately copyrighted as audiovisual works.

> forewords, afterwords, pictorial illustrations, maps, charts, tables, editorial notes, musical arrangements, answer material for tests, bibliographies, appendixes, and indexes, and an "instructional text" is a literary, pictorial, or graphic work prepared for publication and with the purpose of use in systematic instructional activities.

Software simply doesn't fit the definition of something that is akin to "forewords, afterwords, pictorial illustrations, maps, charts, tables, editorial notes, musical arrangements, answer material for tests, bibliographies, appendixes, and indexes."

D. Compilation

"Compilation" is also defined in the Copyright Act.

> A "compilation" is a work formed by the collection and assembling of preexisting materials or of data that are selected, coordinated, or arranged in such a way that the resulting work as a whole constitutes an original work of authorship. The term "compilation" includes collective works.

Typically, the main part of software, sometimes called a "kernel," is written by programmers for the specific functionality that the software is intended to achieve. Because software rarely consists only of "preexisting materials," software does not fit the definition of "compilation."

And that's it. If software does not fit into one of these categories, then subsection (2) of the definition of "work make for hire" doesn't fit. Accordingly, in the software context, except in unusual situations, only subsection (1) of the definition is applicable to software, related to employees only.

In addition to the conclusions of business people being exactly the opposite of the proper legal conclusion, there is a drafting concern here as well. Reliance on work for hire language in a purported transfer of rights in copyright is usually misplaced except in the context of the work of an employee. The operative language that transfers the copyright to software in a typical independent contractor or service provider context is the language of assignment, not the language of work made for hire. Work for hire language doesn't usually hurt, but don't assume that this language is what drives the transfer of copyright.

2. Owning and Assigning Intellectual Property Rights

In some ways, the concept in this section flows form the logical consequences of the work made for hire doctrine.

If a customer asks a developer, who is not an employee, to develop some software, and the customer pays the developer 100% of the cost of its development, who owns the copyright, the customer or the developer? The answer is the developer. Unless the work made for hire doctrine applies, the person who does the creating gets the copyright to the software.

Any assignment of ownership of copyrights must be in writing. No writing, no assignment.[5] At times, however, even a writing doesn't actually transfer the copyright.

This issue has to do with the way we normally draft agreements. When a party has a contractual obligation to do something, we typically use "shall" or "will" to express that obligation. Both "shall" and "will" are future tense verbs. If you tell your spouse, roommate, or significant other that you "*will* go to the store," just after you finish that sentence, you haven't gone to the store yet. Stating in an agreement that a party "will (or shall) own all rights in and to the software," begs the question of "when." This issue is further exacerbated because, frequently, the contract with the assignment language is executed before the software is actually created.

This problem tends to crop up when a company is sold, resulting in the potential problem that the company doesn't actually own the copyright to the software it purports to own.

The way to avoid this issue is to use present tense language of assignment. In almost all cases I've read or had researched on this issue, "hereby" is also added to the language. The following example language seems to work, after adjustment for the actual terms used in the agreement. "Developer hereby transfers and assigns all right, title and interest, including all Intellectual Property Rights, in and to the software to Company."

3. Derivative Works

Many of you may have been involved in negotiations where the licensee of software ("customer") wants to "own" all derivative works. A lawyer for the licensor or an owner of the software ("developer") may be troubled by the possibility that "ownership" of a derivative work means that ownership of the main software of the developer would also be transferred to the customer. Working through this particular issue can be helped significantly by understanding what ownership of a derivative work actually means.

Section 103(b) of the Copyright Act addresses derivative works (as well as compilations).

> The copyright in a compilation or derivative work **extends only to the material contributed by the author of such work, as distinguished from the preexisting material employed in the work, and does not imply any exclusive right in the preexisting Material**. The copyright in such work is independent of, and does not affect or enlarge the scope, duration, ownership, or subsistence of, any copyright protection in the preexisting material. (emphasis supplied)

As can be seen, one can have a copyright in a derivative work, but those rights do not extend to the pre-existing material.

In negotiations where a customer wants to "own" derivative works, and assuming that ownership of the modifications by the customer is otherwise acceptable to the developer, adding the

[5] However, implied license rights may exist.

following phrase (or similar) at the end of the ownership language is usually sufficient to resolve the issue: "subject to developer's underlying rights in the original work."

4. Idea-Expression

There is a split in the circuits in connection with the idea-expression dichotomy, relating to certain aspects of computer software, specifically APIs. Personally, I had hoped that the Supreme Court would have resolved this issue in its recent ruling in Google v. Oracle,[6] but it didn't happen.

Section 102 of the Copyright Act has two subsections. Subsection (a) describes what is protected under copyright.

> (a) 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;
> (2) musical works, including any accompanying words;
> (3) dramatic works, including any accompanying music;
> (4) pantomimes and choreographic works;
> (5) pictorial, graphic, and sculptural works;
> (6) motion pictures and other audiovisual works;
> (7) sound recordings; and
> (8) architectural works.

As can be seen from subsection (a), copyright protection exists in something that is (i) "original" (ii) a "work of authorship," and that is (iii) "fixed in any tangible medium of expression." A nonexclusive list of works of authorship are included.

Subsection (b) of Section 102 describes what is not protected under copyright.

> (b) In no case does copyright protection for an original work of authorship extend to any idea, procedure, process, system, method of operation, concept, principle, or discovery, regardless of the form in which it is described, explained, illustrated, or embodied in such work.

The line drawn between subsections (a) and (b) is known as the "idea-expression dichotomy." Generally speaking, ideas are not protected under copyright, but expressions of ideas are protected under copyright.

The split in the circuits in connection with software relate to things that are expressive while also being described in Section 102(b), most notably a "process" or a "method of operation." Menu

[6] Google LLC v. Oracle Am., Inc., 141 S. Ct. 1183, 209 L. Ed. 2d 311 (2021).

structures and APIs are two examples of parts of computer software that may be considered both expressive and a process or method of operation.

Here are the arguments on both sides of this issue.

A. First Circuit

Before *Oracle v. Google*, the most important case in this area was *Lotus Development Corp. v. Borland International, Inc.*[7] Lotus was the developer of Lotus 1-2-3, a very popular spreadsheet program. Borland was the developer of Quattro Pro,[8] which was competitive replacement for Lotus 1-2-3. Because of the pervasiveness of Lotus 1-2-3, Borland attempted to ease user's transition from Lotus 1-2-3 to Quattro Pro (i) by allowing Quattro Pro users to substitute Quattro Pro's menu structure with Lotus 1-2-3's menu structure and (ii) by allowing macros written for use with Lotus 1-2-3 to run on Quattro Pro without modification, which also required the use of Lotus 1-2-3's menu structure. In multiple decisions, the District Court held that Borland's use of Lotus 1-2-3's menu structure was a copyright infringement.[9]

Reversing the District Court, the First Circuit held that "the Lotus menu command hierarchy was an uncopyrightable 'method of operation.'"[10] It noted that "[t]he Lotus menu command hierarchy does not merely explain and present Lotus 1-2-3's functional capabilities to the user; it also serves as the method by which the program is operated and controlled."[11]

The Court also addressed the relation between expressive elements and methods of operation under the Copyright Act.

> Accepting the district court's finding that the Lotus developers made some expressive choices in choosing and arranging the Lotus command terms, we nonetheless hold that ***expression is not copyrightable*** *because it is part of Lotus 1-2-3's "method of operation."* We do not think that "methods of operation" are limited to abstractions; rather, they are the means by which a user operates something. If specific words are essential to operating something, then they are part of a "method of operation" and, as such, are unprotectable.[12] (emphasis supplied)
>
> The fact that Lotus developers could have designed the Lotus menu command hierarchy differently is immaterial to the question of whether it is a "method of operation." In other words, *our initial*

---

[7] Lotus Dev. Corp. v. Borland Int'l, Inc., 49 F.3d 807 (1st Cir. 1995), *aff'd,* 516 U.S. 233, 116 S. Ct. 804, 133 L. Ed. 2d 610 (1996).

[8] An earlier version of Quattro Pro, Quattro, was also involved in the litigation.

[9] Lotus Dev. Corp. v. Borland Int'l, Inc., 799 F.Supp. 203 (D.Mass. 1992) and Lotus Dev. Corp. v. Borland Int'l, Inc., 831 F.Supp. 223 (D.Mass. 1993).

[10] Lotus Dev. Corp., 40 F.3d 807, 815 (1st Cir. 1995).

[11] *Id.*

[12] *Id.* at 816.

> *inquiry is not whether the Lotus menu command hierarchy incorporates any expression. Rather, our initial inquiry is whether the Lotus menu command hierarchy is a "method of operation."* Concluding, as we do, that users operate Lotus 1-2-3 by using the Lotus menu command hierarchy, and that the entire Lotus menu command hierarchy is essential to operating Lotus 1-2-3, we do not inquire further whether that method of operation could have been designed differently. *The "expressive" choices of what to name the command terms and how to arrange them do not magically change the uncopyrightable menu command hierarchy into copyrightable subject matter*.[13] (emphasis supplied)

As can be seen, the First Circuit held that it did not matter whether the Lotus 1-2-3 menu structure was expressive or not. If the menu structure constituted a method of operation, then it was not protected by copyright. Whether or not the menu structure was expressive was irrelevant.

Part of the First Circuit's rationale for this decision is also telling.

> That the Lotus menu command hierarchy is a "method of operation" becomes clearer when one considers program compatibility. *Under Lotus's theory, if a user uses several different programs, he or she must learn how to perform the same operation in a different way for each program used. … We find this absurd*.[14]

The reasoning here is fairly straightforward. Looking at Section 102(b), it states. "In no case does copyright protection for an original work of authorship extend to any idea, … method of operation…." By using the phrase "In no case does copyright protection … extend," Section 102(b) seems to state that whether protection is otherwise available or not under Section 102(a), if it is an idea or method of operation, then copyright protection does not apply.

B. Federal Circuit, interpreting Ninth Circuit Law

The Federal Circuit's first opinion in *Oracle v. Google* demonstrates the opposite approach.

This litigation was over Google's use of the Java platform in the creation of its Android mobile device software. Oracle owns Java as a result of its purchase of Sun Microsystems, the original developer of Java.[15]

In the creation of Android, Google believed it was important to allow the development of Android applications without requiring programmers to learn a new programming language. Google wanted to allow programmers already schooled in Java to write programs using Java

---

[13] *Id.*

[14] *Id.* at 817-818 (emphasis supplied).

[15] Oracle Am., Inc. v. Google Inc., 872 F. Supp. 2d 974, 975 (N.D. Cal. 2012), *rev'd and remanded*, 750 F.3d 1339 (Fed. Cir. 2014).

standards of programming. However, Google completely rewrote the functional code in all of the Java "methods," which is where the actual functional code resides in Java.[16] Oracle and Google agreed that the rewritten code was not subject to a copyright claim.[17] Therefore, for the most part,[18] Google only copied the names and syntax of a number of Java methods, classes, and packages, in other words, the API. A programmer could therefore program an Android application using the same commands and structures used to program in Java.

Google ultimately used the names and syntax of 37 Java packages, 600 classes, and over 6,000 methods. Even though the underlying functional code was different, the names and syntax of the methods and classes remained the same. It was this identical information that was the subject of Oracle's copyright claim.

The District Court followed *Lotus v. Borland*.

> This order holds that, under the Copyright Act, no matter how creative or imaginative a Java method specification may be, the entire world is entitled to use the same method specification (inputs, outputs, parameters) so long as the line-by-line implementations are different. … The method specification is the *idea*. The method implementation is the *expression*. No one may monopolize the *idea*.[19]

The District Court therefore held that expressive material is not protected under copyright if the material is a method of operation or system.

The Federal Circuit reversed the District Court almost in its entirety.

> [W]e conclude that a set of commands to instruct a computer to carry out desired operations may contain expression that is eligible of copyright protection. … [A]n original work — even one that serves a function — is entitled to copyright protection as long as the author had multiple ways to express the underlying idea.[20]

The Federal Circuit seems to have adopted the concept that the limitation on copyright as described in § 102(b) only applies if the item is not expressive in the first place.

The Federal Circuit seems to be saying that an idea, process, method of operation, etc., can never be expressive. In other words, if something can be expressed in multiple ways, the way something is expressed is necessarily different from the something itself. Accordingly, something that is expressive will only be treated as an "idea" in very limited circumstances, to

---

[16] Other than "range-Check," which is not addressed in this article.

[17] Oracle Am., Inc. v. Google Inc., 872 F. Supp. 2d at 978.

[18] Other than "range-Check."

[19] Oracle Am., Inc. v. Google Inc., 872 F. Supp. 2d at 998.

[20] Oracle Am., Inc. v. Google Inc., 750 F.3d 1339, 1367 (Fed. Cir. 2014).

wit: the merger doctrine, scènes à faire doctrine, and related concepts. The concept of merger and scènes à faire is when certain expressions actually become the idea itself.

The Supreme Court ultimately held for Google, but the Court held for Google on the grounds of fair use. Accordingly the Supreme Court did not resolve the split in the circuits.

5. Rights of Contractors

In a software license grant to a company, what does "company" mean? Will employees of the company have the right to use the software for the benefit of the company? What about independent contractors who use the software for the benefit of the company?

It goes without saying that the principle purpose of contract construction is to determine what the parties intended based on the language in the agreement. Accordingly, there may be language in multiple parts of the agreement that weigh on whether use by independent contractors is included in a license grant to a "company." However, if there is no other language in the agreement pointing to the contrary, a license grant to a company might not be sufficient to allow independent contractors to use the software.

The principal case for this proposition seems to be *MAI Systems Corp. v. Peak Computer, Inc.*[21] This case involved computers that were popular before the personal computer was introduced and before they became so powerful: minicomputers. One of the features of minicomputers was that the hardware manufacturer also provided the operating system. MAI Systems was the provider of a brand of those minicomputers. A few employees of MAI joined Peak Computer, which performed hardware maintenance on MAI Systems minicomputers.

When one performs hardware maintenance, frequently the first action is to turn on the computer and run diagnostic software that is provided as a part of the operating system. MAI Systems sued Peak Computer claiming copyright infringement. MAI Systems prevailed.

Many practitioners were surprised by this result, so much so that Congress got involved. Congress amended the Copyright Act by adding a new Section 117(c), effectively reversing *MAI v. Peak*. It states, in part:

> [I]t is not an infringement for the owner or lessee of a machine to make or authorize the making of a copy of a computer program if such copy is made solely by virtue of the activation of a machine that lawfully contains an authorized copy of the computer program, for purposes only of maintenance or repair of that machine….

However, this amendment is very narrowly drafted, to include only making copies for maintenance or repair of a machine, and not otherwise. The general principle—and the one we are usually most worried about—was not modified by this amendment.

[21] 991 F.2d 511 (9th Cir 1993).

The practical point is fairly simple. A license grant made to a particular entity may not include use by contractors unless the license grant specifically includes them.

6. Who is an employee?

As can be seen from the discussions about work made for hire and whether licenses to a company includes contactors, determining who is an employee and who is a contractor can be quite important. So how do you figure out whether someone is an employee or not?

The Supreme Court seems to apply Federal common law to this question.

> In determining whether a hired party is an employee under the general common law of agency, we consider the hiring party's right to control the manner and means by which the product is accomplished. Among the other factors relevant to this inquiry are the skill required; the source of the instrumentalities and tools; the location of the work; the duration of the relationship between the parties; whether the hiring party has the right to assign additional projects to the hired party; the extent of the hired party's discretion over when and how long to work; the method of payment; the hired party's role in hiring and paying assistants; whether the work is part of the regular business of the hiring party; whether the hiring party is in business; the provision of employee benefits; and the tax treatment of the hired party. … No one of these factors is determinative.[22]

The Supreme Court was fairly clear that this was not an inclusive list of factors. This case involved the work made for hire doctrine. At the time this came out, some practitioners were surprised by this ruling, which is never a good thing for those who assumed the decision would go the other way.

Conclusion

In many ways, the items described in this article are what one might consider some of the "first principles" in the copyright aspects of a technology transactions practice. Interestingly, the older I get, the more I rely on these first principles. Relying on first principles allows one to evaluate fact patterns and legal issues in ways that free you from only those things you have seen or done in the past.

[22] *Cmty. For Creative Non-Violence v. Reid*, 490 U.S. 730, 751 – 752 (1985).