



(/)

HOME (/)
ABOUT
BLOG (/BLOG)
ONLINE ORIGINALS (/ONLINE-ORIGINALS)
SYMPOSIUM
SUBMISSIONS

Should We Take a Closer Look at the Copyright Implications of Translating Computer Code?

MARCH 24, 2023



Blog Post | 111 KY. L. J. ONLINE | March 24, 2023

Should we take a closer look at the Copyright Implications of Translating Computer Code?

By: Zachary Dean Russell, Staff Editor, Vol. 111

In copyright law, computer codes are treated in the same vein as literary works.[1] The definition of a literary work in 17 U.S.C. §101 describes a literary work as one made of words, numbers, or similar symbols that does not otherwise classify as an audiovisual work, which helps to rationalize the convention of treating computer code as a literary work.[2] A computer code falls squarely into that definition as a work comprised of nothing more than letters, numbers, and various associated symbols. Additionally, a computer code is the unique expression of a programmer's creative endeavors which confers copyright protection upon the code, much like a writer would have copyright protection for a short story or novel.[3] Based on these parallels, the rationale makes sense, but what about other aspects of copyright law? Does this comparison make sense for other aspects of copyright law such as derivative works and fair use? After thinking it through, I think it may be worth taking a closer look at the copyright implications of translating computer codes, specifically source codes. The current test courts use for comparing computer codes may provide too narrow of protection, and there is a risk of programmers translating codes into new languages to prevent reinventing the wheel.

To begin, I can see someone asking, "You can translate a computer code?" Yes! The source code of a computer program contains lines of commands written by a programmer. This source code is run through a compiling program to convert the human-readable lines of code into the binary, the "ones and zeros," that the computer can process to perform different functions.[4] Similar to how conventional literary works may be written English, Greek, or Japanese, source codes can be written in different programming languages.[5] There are many different programming languages, and some may have

close similarities between them, but each language has its own nuances. Copying a segment of code written in one language may not function properly when inserted in a code written in another language. It is similar to how certain phrases or slang in one language may not exist or may lose its meaning when directly translated into another language.[6]

Follow-up questions may include, “how could this possibly lead to a common issue?” Well, upon searching for a list of programming languages on Google, one can find a Wikipedia page that lists dozens languages.[7] Additionally, it seems new programming languages are constantly being developed.[8] It’s not hard to imagine early adopters of these new programming languages wanting to test the waters by trying to translate preexisting source codes to make them function exactly the same in these new languages. So, ultimately, what are the copyright implications if someone were to take a preexisting code, or a major section of one, and translate it into a new programming language? Should the copyright owner of the original code be able to claim copyright infringement? Or should this be chalked up to fair use under 17 U.S.C. § 107?[9] Afterall, it doesn’t make much sense to keep “reinventing the wheel,” and it’s only natural some programmers may wish to use preexisting codes as a template to achieve certain functions via new programming languages.

If we rigidly adhere to the parallels between literary works and computer codes, one likely path involves derivative works under 17 U.S.C. § 103.[10] Translations of an original copyrighted work falls under the definition of a derivative work.[11] Permission from the original author is required to create a derivative work, otherwise the translator would risk infringing on the original work.[12] But a potentially infringing translator can call upon the defense of fair use, which, according to some, is an almost all-consuming doctrine.[13] There is also the need to consider case law’s influence. One case that provides some guidance is the 1992 case, *Computer Associates International, Inc. v. Altai, Inc.*, which brought about the abstraction-filtration-comparison (AFC) test.[14] That said, working backwards, *Whelan Associates, Inc. v. Jaslow Dental Laboratory, Inc.* may provide a better, more similar fact pattern to the general concern of translating computer codes despite the test—the *Whelan* test—used within being replaced with the AFC test from *Altai*. [15] Both tests raised concerns. Where the *Whelan* test was deemed too broad, at the time of the opinion, the AFC test brought concerns of being too narrow with respect to the scope of copyright protection.[16] Back when *Altai* was decided thirty years ago, scope of protection went from being criticized as overly broad, based on the *Whelan* test, to being too narrow. Additionally, the field of computer science is constantly moving forward with new coding languages being developed each year. I think there is value in taking a new look at how we should evaluate issues involving the translation of copyrighted computer codes in a way that best benefits the opposing interests of allowing the field to grow while protecting programmers’ work.

[1] 17 U.S.C. § 101 (2022).

[2] *Id.* (“Literary works’ are works, other than audiovisual works, expressed in words, numbers, or other verbal or numerical symbols or indicia, regardless of the nature of the material objects, such as books, periodicals, manuscripts, phonorecords, film, tapes, disks, or cards, in which they are embodied.”).

[3] 17 U.S.C. § 102 (2022).

[4] Christina M. Reger, *Let’s Swap Copyright for Code: The Computer Software Disclosure Dichotomy*, 24 *Loy. L.A. Ent. L. Rev.* 215, 219 (2004).

[5] *Id.*

[6] For example, in the United States, we may call someone a “windbag” if they are overly boastful, but in China, they have a phrase “chūī niú,” which literally means to “blow on cows.” Vivian Wang, *10 Unique Chinese Expressions You Need to Know*, *Culture Trip* (Oct. 12, 2016), <https://theculturetrip.com/asia/china/articles/10-unique-chinese-expressions-you-need-to-know/>.

[7] Simply look up “list of programming languages” and a link to Wikipedia should show up with the same name. *E.g.*, *List of Programming Languages*, Wikipedia, https://en.wikipedia.org/wiki/List_of_programming_languages (Feb. 21, 2023, 07:38 UTC).

[8] “Seems” may be a bit of an understatement. Depending on the source, several coding languages were developed over the past year in 2022. Go to Google and search “coding languages created in 2022.” See, e.g., Sayantani Sanyal, *Ditch the Old: These are the new Programming Languages of 2022*, *Analytics Insight* (Jan. 25, 2022), <https://www.analyticsinsight.net/ditch-the-old-these-are-the-new-programming-languages-of-2022/>. An additional implication of this forward progress is, as new programming languages are developed, older programming languages may be left behind. Then we may ask, what are the odds someone may take a code written in an older, relatively “abandoned” programming language and translate it into a new programming language?

[9] 17 U.S.C. § 107 (2022).

[10] 17 U.S.C. § 103 (2022).

[11] 17 U.S.C. §§ 103, 106 (2022).

[12] *Id.*


[13] 17 U.S.C. § 107 (2022); see generally Clark D. Asay, Arielle Sloan, & Dean Sobczak, *Is Transformative Use Eating the World?*, 61 *B.C. L. Rev.* 905 (2020) (discussing the prevalence of the transformative use doctrine, a major subcategory of the fair use doctrine, and how it has become the focal point of fair use since the mid-2000s).

[14] *Computer Associates International, Inc. v. Altai, Inc.*, 982 F.2d 693, 712 (2d Cir. 1992) (holding the AFC test first determines which non-literal elements of the codes may receive copyright protection and which elements may not receive copyright protection through the abstraction step. Second, the test filters out the elements that may not receive copyright protection. Finally, the test compares the remaining non-literal elements of the copyrighted computer code with the defendant's computer code to determine if the elements are substantially similar.).

[15] *Whelan Associates, Inc. v. Jaslow Dental Laboratory, Inc.*, 797 F.2d 1222, 1225–27 (3d Cir. 1986) (detailing how the facts in *Whelan* involved comparing two codes written in different coding languages while the test essentially determined the overall purpose of the computer code, which was considered an unprotectable idea, and deemed everything else within the code a copyrightable expression.); *Altai*, 982 F.2d at 706–12.

[16] *Altai*, 982 F.2d at 712.

 (<https://www.facebook.com/sharer/sharer.php?u=https%3A%2F%2Fwww.kentuckylawjournal.org%2Fblog%2Fshould-we-take-a-closer-look-at-the-copyright-implications-of-tr>)

 (<https://twitter.com/intent/tweet?url=https%3A%2F%2Fwww.kentuckylawjournal.org%2Fblog%2Fshould-we-take-a-closer-look-at-the-copyright-implications-of-translating-c>)



PREVIOUS

The Dilemma of Judicial Remedies to Non-Compete Agreements

(/blog/the-dilemma-of-judicial-remedies-to-non-compete-agreements)

NEXT

The Pending Tragedy in Kentucky's Family Court System

(/blog/the-pending-tragedy-in-kentuckys-family-court-system)



(859) 257-1678 | kentuckylawjournal112@gmail.com | 620 Limestone Lexington, KY 40508

(<https://kentuckylawjournal.org>)