

Is Your Software Code Unique?

Litigating Copyright Infringement Claims Related to Competing Software Applications

Introduction

Software copyright infringement litigation, frequently called software anti-piracy claims, involve disputes between software publishers and their end-user customers¹. Suits by one software publisher against another over trade secrets and copyright ownership generally involve competing works. However, “competing works litigation” typically requires substantially more effort from the parties and the tribunal than either of the above types of disputes. In these cases, the developer or owner of one program complains that a different product created or distributed by the defendant consists, in whole or in part, of the work in which the plaintiff holds the copyright. These cases on average involve significantly higher stakes than other software copyright disputes. The legal analyses and factual development in such matters can approach the level of complexity usually associated with patent disputes, and, indeed, many of the judicial frameworks regarding infringement analysis are similar to patent jurisprudence.

Is the Software Copyrightable

An alleged infringer may choose to argue initially that the program at issue was not copyrightable. According to the Fifth Circuit, “for a work to qualify for copyright protection, it must be original...and originality, as the term is used in copyright, requires both ‘independent creation’ and ‘a modicum of creativity.’”² It is settled that “the literal elements of computer programs, i.e., their source and object codes, are the subject of copyright protection.”³ If the work has been copied from the public domain, it cannot be copyrighted.⁴ A defendant may be able to contend that the software at issue is merely a compilation of pre-existing protocols. Such a compilation is copyrightable under 17 U.S.C. § 101. *In Feist*

Publications, Inc. v. Rural Tel. Serv. Co.,⁵ the Supreme Court indicated that “notwithstanding a valid copyright, a subsequent compiler remains free to use the facts contained in an author’s publication to aid in preparing a competing work, so long as the competing work does not feature the same selection and arrangement.”⁶ Subsequent cases have interpreted *Feist* to mean that compilation copyright protection is very limited and usually requires substantial verbatim copying.⁷ In addition, “some programming choices are either too trivial to support a finding of originality or are so constrained by practical reality as to lack originality.”⁸ “Selection from among competing ideas or methods of operation generally does not result in copyright-protectable expression.”⁹ To the extent the defendant is able to demonstrate that the software program was a compilation of non-original elements, the court may conclude that the copyright was invalid, particularly if the court concludes that the amount of original contribution to the program was *de minimis*.¹⁰

Claim Establishment

If a dispute is able to move past the question of whether the software at issue is eligible for copyright protection, the claimant will need to be prepared to present a *prima facie* claim of infringement. The Copyright Act and relevant case law do not distinguish between software and other copyrightable works for the purposes of setting forth the requirements of a claim. To establish a claim for software copyright infringement, a plaintiff must show (1) ownership of a valid copyright; (2) factual copying of the code or documentation; and (3) substantial similarity between the software applications.¹¹

Ownership in a Software Copyright

The developer of a software application acquires a common-law copyright interest at the time the software is published, unless the developer has entered into an agreement assigning the interest to another party. The filing of a certificate of registration creates a rebuttable presumption of the validity of a copyright and of the facts contained in the certificate regarding ownership.¹² However, the fact that software is often developed through subcontractors or using a collaborative approach may complicate the analysis. In some cases, for example, the “work made for hire” doctrine may control the question of who legally owns the software. Under this doctrine, “the employer or other person for whom the work was prepared is considered the author” of a “work made for hire” and owns the copyright, absent an agreement between the parties to the contrary.¹³ The Copyright Act defines a “work made for hire,” *inter alia*, as “a work prepared by an employee within the scope of his or her employment...”¹⁴ The Supreme Court has held that general principles of agency law apply when deciding if a work was prepared by an employee rather than an independent contractor.¹⁵ If an alleged infringer was an employee of a copyright owner when he or she developed software and/or documentation, it will be presumed that the works belonged to the employer for purposes of the Copyright Act.¹⁶ The alleged infringer then would have to prove the existence of a written agreement signed by both parties under which he or she retained ownership rights.¹⁷

Factual Copying

Once the question of who owns the software is established, the copyright owner must be able to prove that the alleged infringer copied the works at issue. Factual copying may be proved by direct or circumstantial evidence, but in most cases a copyright owner must rely on circumstantial evidence. To make out a circumstantial claim, a plaintiff must prove (1) that the defendant had access to the copyrighted work before creation of the infringing work and (2) that the works contain similarities that are probative of copying.¹⁸ If a plaintiff is able to combine the existence of access to the copyrighted work and similarities between the two works, this establishes the presumption as a matter of law that copying in fact occurred.¹⁹ However, the presumption is rebuttable, and once a plaintiff circumstantially establishes factual copying, the defendant may respond with evidence that he independently created the work at issue.²⁰

Substantial Similarity: Abstraction-Filtration Method

Substantial similarity between competing software works is the third element of the copyright infringement claim. In assessing whether a computer program has been infringed, the Fifth Circuit has adopted the “abstraction-filtration” method proposed by the Tenth Circuit in *Gates Rubber Company v. Bando Chemical Industries*.²¹

Under the abstraction-filtration test, the court first dissects

the program according to its varying levels of generality as provided in the abstractions test.²² Learned Hand presented a summary of the analysis involved in the abstractions test when he wrote, in the context of alleged infringement of a theatrical script:

Upon any work, and especially upon a play, a great number of patterns of increasing generality will fit equally well, as more and more of the incident is left out. The last may be no more than the most general statement of what the play is about, and at times might only consist of its title; but there is a point in this series of abstractions where they are no longer protected, since otherwise the playwright could prevent the use of his “ideas,” to which, apart from their expression, his property is never extended.²³

According to the *Gates* court, “a computer program can often be parsed into at least six levels of generally declining abstraction: (i) the main purpose [‘a description of the program’s function or what it is intended to do’], (ii) the program structure or architecture [‘a description of how the program operates in terms of its various functions, which are performed by discrete modules, and how each of these modules interact with each other’], (iii) modules [consisting of ‘operations,’ which identify a particular result or set of actions that may be performed, and ‘data types,’ which define the type of item that an operator acts upon], (iv) algorithms [‘more specific manifestations of operations... a specific series of steps that accomplish a particular operation’] and data structures [‘precise representation[s] or specification[s] of... data type[s] that [consist] of (i) basic data type groupings such as integers or characters, (ii) values, (iii) variables, (iv) arrays or groupings of the same data type, (v) records or groupings of different data types, and (vi) pointers or connections between records that set aside space to hold the record’s values’], (v) source code [‘the literal text of a program’s instructions written in a particular programming language’], and (vi) object code [‘the literal text of a computer program written in a binary language through which the computer directly receives its instructions’].”²⁴ Expert testimony often is necessary in order to educate the court and the jury regarding the organization of a program into the appropriate levels of abstraction.²⁵

Second, poised with this framework, the court should examine each level of abstraction in order to filter out those elements of the program that are not protectable.²⁶ Filtration should eliminate from comparison those aspects of the software that are not eligible for copyright protection, including ideas, processes, facts, and public domain information.²⁷ Filtration also should include application of the doctrines of merger²⁸ and *scenes a faire*²⁹ to remove any additional elements that are not eligible for protection.

Finally, the court compares the remaining protectable elements of the original program with the allegedly infringing program to determine whether the defendants have misappropriated substantial elements of the plaintiff’s



program.³⁰ The goal of the analysis should be to determine whether any copied elements constitute “matter that is significant in the plaintiff’s program.”³¹ This is a qualitative, rather than quantitative analysis, the outcome of which will depend heavily on the unique facts of each case.³²

If an alleged infringer is able to establish a *prima facie* case of circumstantial infringement based on the apparent similarities between the defendant’s software and documentation and those used by defendants, the burden of proof then shifts to the defendant to demonstrate that no copying occurred. An expert with experience in software coding likely will be needed to rebut any expert testimony presented by the plaintiff during the abstracted analysis of the two programs. While a line-by-line comparison of the documentation may reveal many similarities between the works, experts may testify that they are merely surface similarities between the plaintiff’s and defendant’s software and documentation, many of which may be attributable to the similar functions and purposes of the systems.

Damages

Software copyright plaintiffs typically seek both permanent injunctive relief as well as damages. Recovery of statutory damages under 17 U.S.C. § 504 often hinges on whether the copyrights claimed to have been infringed before or after discovery of the alleged infringement.³³ However, plaintiffs in competing works litigation typically seek an actual damages award, because a potential actual damages recovery often is greater. In addition, the marginal costs of developing the necessary factual record to support an actual damages award are not significant, because the underlying elements of the claim already require the devotion of significant time and effort to evidence collection and presentation. Under 17 U.S.C. § 504, a plaintiff may recover the actual damages it suffered as a result of the infringement or any profits of the infringer attributable to the infringement. Under 17 U.S.C. § 504(b), the plaintiff could recover any profits of the infringer that are attributable to the infringement. Under the statute, “in establishing the infringer’s profits, the copyright owner is required to present proof only of the infringer’s gross revenue, and the infringer is required to prove his or her deductible expenses and the elements of profit attributable to factors other than the copyrighted work.”³⁴ Those damages could be substantial, depending on the amount of business and profit the plaintiff is able to demonstrate is attributable to use of its works. Claims for attorneys’ fees also usually are the norm, though, again, recovery may depend on whether the copyrights at issue were registered before or after discovery of the alleged infringement. Costs also may be recoverable.³⁵

Joint and Several Liability for Software Copyright Infringement

Competing works cases often involve one or more primary, individual alleged infringers as well as the corporate entities with which they are associated. If the plaintiff is able to

establish any actual damages as a result of infringement, all defendants could be held jointly and severally liable for those damages.³⁶ In addition, the plaintiff in the action may seek to hold the individual defendants liable for the “profits” they made independently as a result of the alleged infringement. Specifically, the plaintiff could attempt to recover a portion of the individuals’ income earned while developing and/or selling the competing work at issue.

Conclusion

Competing works software copyright litigation presents practitioners and parties with both a unique set of legal challenges and factual burdens as well as the potential of high-dollar reward or exposure. For that reason, a software developer considering or involved in litigation should involve knowledgeable IP counsel from the very earliest stages of claim assessment. Consideration should be given to mediation and arbitration as potential cost-saving measures, but a thorough and early evaluation of all of the evidence that can be presented in support of a claim of infringement is the single most important step toward reaching a result that achieves the party’s principal goals while keeping the expense of litigation under reasonable control.

¹These causes of action stem from the remedies allowed for copyright infringement under the U.S. Copyright Act at 17 U.S.C. § 501, *et seq.*

²Alcatel USA, Inc. v. DGI Technologies, Inc., 166 F.3d 772, 787 (5th Cir. 1999).

³Computer Assocs. Int’l, Inc. v. Altai, Inc., 982 F.2d 693, 702 (2d Cir. 1992).

⁴Folio Impressions, Inc. v. Byer Cal., 937 F.2d 759, 763-64 (2d Cir. 1991).

⁵499 U.S. 340 (1991).

⁶*Id.* at 349.

⁷Key Publications v. Chinatown Today, 945 F.2d 509, 514 (2d Cir. 1991).

⁸Ross, Brovins & Oehmke, P.C. v. Lexis Nexis Group., 463 F.3d 479, 485 (6th Cir. 2006).

⁹Lexmark Int'l Inc. v. Static Control Components, Inc., 387 F.3d 522, 540-41 (6th Cir.2004); see also Bateman v. Mnemonics, Inc., 79 F.3d 1532, 1546, n. 29 (11th Cir.1996) (“methods of operation and processes are not copyrightable.”).

¹⁰Lexmark, 387 F.3d at 540. See *Sega Enterprises, Ltd. v. Accolade, Inc.*, 977 F.2d 1510, 1524 n.7 (8th Cir. 1992) (noting that 20-byte code is of *de minimis* length and therefore likely a “word” or “short phrase” that is not protected by copyright law).

¹¹Armour v. Knowles, 512 F.3d 147, 152 (5th Cir. 2007).

¹²17 U.S.C. § 410(c).

¹³Quintanilla v. Texas Television, Inc., 139 F.3d 494, 496 (5th Cir. 1998), citing 17 U.S.C. § 201(b).

¹⁴17 U.S.C. § 101.

¹⁵Community for Creative Non-Violence v. Reid, 490 U.S. 730, 737 (1989).

¹⁶Quintanilla, 139 F.3d at 496.

¹⁷17 U.S.C. § 201(b); see *Saenger Organization, Inc. v. Nationwide Ins. Licensing Assoc., Inc.*, 199 F.3d 55 (1st Cir. 1997).

¹⁸Armour, 512 F.3d at 152.

¹⁹Id.

²⁰Id.

²¹9 F.3d 823 (1993).

²²Computer Management Assistance Co. v. Robert F. DeCastro, Inc., 220 F.3d 396, 400 (5th Cir. 2000).

²³Nichols v. Universal Pictures Corp., 45 F.2d 119 (1930), cert. denied, 282 U.S. 902, 51 S.Ct. 216, 75 L.Ed. 795 (1931).

²⁴Gates, 9 F.3d at 835 (quoting John W.L. Ogilvie, *Defining Computer Program Parts Under Learned Hand's Abstractions Test in Software Copyright Infringement Cases*, 91 Mich. L. Rev. 526 (1992)).

²⁵Gates, 9 F.3d at 835.

²⁶DeCastro, 220 F.3d at 400.

²⁷Id.

²⁸“Under the merger doctrine, copyright protection is denied to expression that is inseparable from or merged with the ideas, processes, or discoveries underlying the expression.” *Gates*, 9 F.3d at 838.

²⁹According to the Fifth Circuit, “the scenes a faire doctrine excludes from copyright protection work serving functional purposes or work that is dictated by external factors such as particular business practices... In the area of computer programs these external factors may include: hardware standards and mechanical specifications, software standards and compatibility requirements, computer manufacturer design standards, target industry practices and demands, and computer industry programming practices.” *Computer Management*, 220 F.3d at 401.

³⁰DeCastro, 220 F.3d at 400.

³¹Gates, 9 F.3d at 839.

³²Id.

³³See 17 U.S.C. § 412; *Mason v. Montgomery Data, Inc.*, 967 F.2d 135, 143-44 (5th Cir. 1992).

³⁴17 U.S.C. §504(b).

³⁵17 U.S.C. § 412.

³⁶Id.

Robert J. Scott
Managing Partner
Scott & Scott, LLP

Christopher C. Barnett
Associate Attorney
Scott & Scott, LLP