Intellectual Property Today™
www.iptoday.com
381 W. Northwest Hwy., Palatine, IL 60067
A Publication of Omega Communications

Challenging the Validity of Software Copyright Registrations: The Deposit Copy Requirement

BY JEFF PARMET, BRENDAN McPARLAND, AND BRYCE HELFER

Challenging the validity of a software copyright registration is a fundamental and effective strategy for a party accused of copyright infringement. If a federal court finds that the registration is invalid, the court will dismiss the infringement suit.[1] A registration is invalid if it contains *material* errors.[2] While *immaterial* errors, such as inaccurately stating the date of the work's creation or failing to list pre-existing works or coauthors, do not jeopardize the validity of a registration, the failure to submit a "proper deposit copy" is deemed a material error, and therefore can invalidate the entire copyright registration.[3] In this article we explore methods for assessing whether a deposit copy is proper in the context of software copyright registrations.

THE DEPOSIT COPY REQUIREMENT: EXAMINING SECTION 408 OF THE COPYRIGHT ACT

Section 408 of the Copyright Act requires applicants for copyright registration to include "one complete copy" of their work as part of the application.[4] However, applicants seeking to register computer programs for copyright are permitted to instead deposit "identifying material," consisting of the first and last 25 pages of source code.[5]

An Original or Bona Fide Copy… And Nothing Less

A copyright registration of a computer program ideally includes either an original copy or a *bona fide* copy of the program.[6] An original copy consists of a clean, non-updated, unaltered copy of the software source code as it was originally published. The safest methods of source code storage to preserve an original copy are: (1) a repository within a source code control system[7] maintained by the developer of the registered software; or (2) an escrow account established with a third-party escrow agent. The preserved original copy may be utilized to produce a *bona fide* copy which is "virtually identical to the original" and was "produced by directly referring to the original."[8] Unless an applicant utilizes these preservation and authentication methods, the certificate of application may be vulnerable to an invalidity challenge because the deposit copy will likely not represent a *bona fide* copy of the original work at the time of its creation. As a result of some applicants' attempts to re-create original works, courts have held that an "original work" means exactly that: the deposit copy must be virtually identical to the original work and produced directly from reference to the original rather than from memory.[9]

PROPOSED METHODS FOR EXAMINING THE VALIDITY OF SOFTWARE DEPOSIT COPIES

Before examining whether the deposit copy is valid, the defendant's software expert should determine whether the complaining party is alleging infringement of the deposit copy. In *Airframe Systems, Inc. v. L-3 Communications*, the First Circuit Court of Appeals found that the plaintiff had failed to establish a *prima facie* case of copyright infringement because the plaintiff compared the defendant's allegedly infringing software with an unregistered updated version of the plaintiff's source code rather than the code filed with the Copyright Office.[10] Consequently, the plaintiff had not demonstrated that there was "substantial similarity" between the registered code and the defendant's accused product.[11] Before examining the validity of the deposit copy, the software expert should ensure that the plaintiff's infringement claim relates to a work that has actually been registered for copyright and not a subsequent, unregistered version.

First Step: Examination of Copyright Notice

The software expert should first ensure that the copyright notice included in the deposited code corresponds to the information contained in the registration certificate. All applicants for software copyright registration must include with their deposit material the page (or portion of source code) containing the software's notice of copyright.[12] If the year indicated in the copyright notice falls after the date of publication, this may be the first indication that the applicant has filed an inaccurate deposit copy.

Examination of Source Code File Dates

One technique commonly used by software experts is analyzing the creation and last-modified dates embedded in the metadata of the deposited source code. If the expert finds that the deposit copy contains file creation and last-modified dates that fall after the date of publication indicated on the registration certification, then he will likely conclude that the deposit copy could not possibly represent a *bona fide* copy of the source code as it existed at the time of its publication.

It should be noted that, without access to the version control system in which the software source code was stored, the expert will likely be unable to determine much more about the source code files beyond the times of their creation and last modification. The metadata alone provides no information regarding which programmer changed a source code file, how many times a file was changed since its creation, or what was the exact nature and purpose of the change. However, even without the benefit of studying the data about source code files stored in a version control system, the expert may discover additional clues regarding the origin and development of source code files by examining code comments.[13]

Examination of Code Comments

Programmer comments in source code can reveal useful information, such as the purpose of a particular modification to a source code file, the individuals responsible for the modification, and even the time of the modification. If, upon examining code comments, the expert finds multiple references to code alterations that were made after the publication date stated on the registration certificate, the expert may conclude that the deposited source code does not represent a *bona fide* copy of the original work.

In order to determine whether modifications were made to the original source code, the software expert should first examine the "identifying material" deposited as part of the copyright registration application, which is comprised of the first and last 25 pages of source code. Even this relatively small portion of the program may contain references to file creation or modification dates that are later than the date of publication indicated on the certificate of registration. For example, file change history logs embedded in code comments might contain entries indicating post-publication modifications to pre-existing files or the addition of newly created files. Furthermore, an analysis of the comments embedded throughout the entire set of deposited source code files may reveal additional references to post-publication dates of file creation and modification.

Similarly, if the code comments contain the names of the programmers who modified the code, the expert can then refer to deposition testimony to determine the period during which individual programmers were involved in the development of the program. If the testimony indicates that particular individuals did not join the software development team until after the date of publication, this may also undercut any argument that the submitted source code is a copy of the original work.

Keyword Searches for Post-Publication Subject Matter

If the deposit copy is in fact a *bona fide* copy as it existed on the date of its publication, the expert would not expect the deposited code to reference events that occurred *after* that date. For example, if a plaintiff produces business records that refer to contracts that were not even entered into until after the publication date, any reference to such contracts in the deposited code would raise questions about the authenticity of the deposit copy. If the expert searches the deposited source code for keywords relating to such post-publication contracts, and that search results in multiple relevant hits, the expert may conclude that the deposit copy cannot represent a *bona fide* copy of the original work.

Filtration of Open Source and Previously Copyrighted Code

Often, the deposit copy contains portions of source code written by parties other than the applicant. Because copyright protection extends only to works of original authorship, it is necessary to identify those portions of the deposit copy that were authored by third parties and are therefore not protected by the applicant's copyright registration. A filtration analysis, such as one similar to the "abstraction-filtration-comparison" test established by the Second Circuit in Computer Associates International, Inc. v. Altai, Inc., may be helpful in identifying certain portions of the deposited code to which the applicant may not claim copyright.15 An expert performing filtration for open-source code typically runs a script that performs an Internet search on each line of code. If identical copies of the code lines appear on multiple websites, it is much more likely to be open source than proprietary code, which one would not expect to be in open circulation on the Internet. Obviously, depositing open-source or other third-party code with the Copyright Office as part of a registration application places the originality of that work in doubt.

CONCLUSION

The courts have yet to endorse one preferred method for experts conducting examinations of deposited source code in cases of software copyright infringement. However, each of the techniques described above yields findings regarding the origins of the deposited source code that a court would likely find persuasive in judging whether a deposit copy is indeed a *bona fide* copy of the original work.

ENDNOTES

[1] *Torres-Negrón v. J&N Records LLC*, 504 F.3d 151, 160 (1st Cir. 2007).

[2] *Id.* The party may also show fraud or that the application is incomplete.

[3] *Id.* at 158.

[4] 17 U.S.C. § 408(c)(1) (2005).

[5] *Id.* The applicant must also submit the page containing the copyright notice. *See* U.S. Copyright Office Circular 61, "Copyright Registration for Computer Programs," Reviewed: 03/2012 (http://www.copyright.gov/circs/circ61.pdf).

[6] In order to preserve the option of winning statutory damages and attorneys' fees in an infringement suit, the registration should be made within three months of publication. *See* 17 U.S.C. § 412 (2008).

[7] A "source code control system" (also known as a "version control system" or "revision control system") is a software tool designed to help manage changes to source code. A repository for source code refers to a place where large amounts of source code are kept, either publicly or privately. *See* Wikipedia, http://en.wikipedia.org/wiki/Codebase.

[8] *Kodadek v. MTV Networks, Inc.*, 152 F.3d 1209, 1211 (9th Cir. 1998).

[9] *Coles v. Wonder*, 283 F.3d. 798, 802 (6th Cir. 2002).

[10] *Airframe Systems, Inc. v. L-3 Communications Corp.*, 658 F.3d 100, 104 (1st Cir. 2011).

[11] *Id.*

[12] *See* U.S. Copyright Office Circular 61, p. 2.

[13] In computer programming, a "comment" is a programmer's annotation in the source code added for the purpose of making the source code easier to understand. Comments are ignored by compilers and interpreters.

[14] Open source software generally contains a free license for public use. *See* http://opensource.org. Open source software is gathered and maintained by Open Source Initiative which is the community recognized body for reviewing and certifying source code to become part of the Open Source Initiative database. *See* http://opensource.org/docs/osd. For example, an expert may refer to the list of "Approved Licenses" published by the Open Source Initiative website when examining the deposited code to determine whether it contains any open-source licenses. *See* http://www.opensource.org/licenses/alphabetical.

[15] The "abstraction-filtration comparison test" is a three-step procedure that determines whether any non-literal elements of two or more computer programs are substantially similar. *Computer Associates Int'l, Inc. v. Altai, Inc.*, 982 F.2d 693, 706 (2nd Cir. 1992).

About the Authors

Jeff Parmet is a widely respected consulting and testifying expert and the founder and managing partner of DisputeSoft (as well as a non-practicing attorney and member of the State Bar of California). He has been involved in more than 125 software disputes, and has served as a computer software expert witness in high-profile software product development and licensing. During the past 15 years as an expert witness, Mr. Parmet has consulted and testified in software patent infringement, copyright infringement and trade secret misappropriation cases, as well as in software failure disputes. Mr. Parmet has a sub-specialty in computer forensics. He can be reached at jparmet@disputesoft.com.

Brendan McParland serves as General Counsel at DisputeSoft. He works closely with clients and assists on cases related to software failure and patent infringement. Mr. McParland has helped prepare affidavits and expert reports for cases before federal district courts, the United States Court of Federal Claims, and international tribunals such as the International Trade Commission. Before joining DisputeSoft, Mr. McParland was an associate with Am Law 100 law firms handling matters involving corporate mergers and acquisitions, investment management, and complex commercial litigation. Mr. McParland can be reached at bmcparland@disputesoft.com.

Bryce Helfer is a student in the Graduate Tax Program at Georgetown University Law Center, with an anticipated graduation date of May 2013. Prior to enrolling at Georgetown, Mr. Helfer graduated from the University of Maryland Francis King Carey School of Law. At DisputeSoft, Mr. Helfer researched legal topics related to copyright and patent infringement, and assisted in preparing expert reports on behalf of clients. Before joining DisputeSoft, Mr. Helfer served as a summer associate at a mid-sized law firm handling issues relating to tax controversies, estate and trust litigation, and complex commercial lease arrangements. Mr. Helfer can be reached at bhelfer@umaryland.edu.