Congressional Research Service
Informing the legislative debate since 1914

Legal Sidebar

Google v. Oracle: Supreme Court Rules for Google in Landmark Software Copyright Case

May 10, 2021

On April 5, 2021, the Supreme Court issued its highly anticipated decision in *Google LLC v. Oracle America Inc.*, the culmination of a decade-long software copyright dispute between the two tech giants. Resolving what observers have hailed as the "copyright case of the century," the Court held in Google's favor, finding that Google's copying of the "declaring code" of the Java SE application programming interface (API) was a fair use and thus did not infringe Oracle's copyright in Java.

As a formal matter, the Court's holding was relatively narrow, concluding that Google's copying of certain code from the Java API—what the Court characterized as "reimplementation of a user interface"—was a fair use under the "case-by-case" balancing of the statutory fair use factors. As a practical matter, however, the Court's decision is likely to have major significance for the software industry, and may also potentially affect fair use for other copyrightable subject matter—such as art, music, and television. This Sidebar reviews the basics of copyright in software; the dispute in *Google v. Oracle*; the Court's decision; and potential effects for computer technology and other copyright-intensive industries. It then briefly highlights some considerations for Congress.

Software Copyright Basics

Copyright law grants certain exclusive legal rights to authors of original creative works, such as books, music, visual art, and architecture. Since 1980, the Copyright Act has explicitly protected computer programs as a type of literary work. Applying legal principles originally crafted for books to computer code has not always been a straightforward task, in part because computer programs are more functional than most other copyrightable subject matter. Courts have thus long wrestled with the appropriate scope of copyright protection in computer code. When the Supreme Court last heard a case involving software copyright in the 1990s, it divided 4-4 and therefore issued no precedential decision.

Three key copyright doctrines affect the scope of copyright protection for computer code. The first is the idea/expression dichotomy, codified in Section 102(b) of the Copyright Act, which states that copyright protection does not "extend to any idea, procedure, process, system, method of operation, concept, principle, or discovery." This doctrine derives from the Supreme Court's 1880 decision in *Baker v. Selden*, which held that the copyright in a book describing a system of accounting extended only to the

Congressional Research Service
https://crsreports.congress.gov
LSB10597

author's particular description of that system (the book's "expression") and not to the accounting system itself (the book's "idea").

The second doctrine, known as merger, is a corollary to the idea/expression distinction. When there are only a few ways to express an idea, the expression is said to "merge" with the idea, and neither is copyrightable. Otherwise, copyright could protect *all* of the few ways to express the idea, which would effectively monopolize that idea while the copyrights last. One central purpose of both the idea/expression distinction and merger is to prevent the use of copyright to monopolize general ideas or functional systems. Unlike books or television, however, computer programs primarily serve functional purposes, which complicates applying the idea/expression distinction.

The third doctrine is fair use, which permits limited uses of copyrighted works that might otherwise be infringements, such as using portions of a copyrighted work in a parody or book review. To determine whether a use is fair, courts consider several factors, including (1) the purpose and character of the use, (2) the nature of the original work, (3) the amount and substantiality of what was copied, and (4) the commercial effect from the use on the potential market for the original work. As part of the first factor, courts also consider whether the alleged fair use is "transformative"; that is, whether it adds new expression, serves a different purpose, or alters the original work with new expression or meaning. Applications of fair use are wide-ranging; under "the common law tradition of fair use adjudication," courts rely on fair use to avoid "rigid application" of copyright liability when it would "stifle the very creativity which [copyright] is designed to foster."

The Dispute in *Google v. Oracle*

The dispute in *Google v. Oracle* concerns the Android operating system for smartphones. In developing Android, Google copied certain elements of Oracle's Java programming language and platform. In particular, Java contains thousands of *methods*, sometimes referred to collectively as the API. Methods are modules that application developers can invoke (or "call") to perform certain functions, rather than writing basic code from scratch. For example, Java's "*Math*" class includes, among other related methods, the "*max*" method, a pre-built function that Java programmers can use to output the greater of two input values. Thus, a programmer can call *java.lang.Math.max(x, y)* to determine whether x or y is a larger number (and output that number), rather than independently writing code to perform this function. Java groups related methods into classes, and related classes into packages.

In building Android, Google copied the "declaring code" of 37 of the Java API's 166 packages. The declaring code includes the name for the function (for example, "max") and its syntax, as well as its place within Java's taxonomy of methods (for example, within the "math" class). Google independently wrote Android's "implementing code," the operative code that performs the method. In all, Google copied more than 11,000 lines of code (of about 15 million in Android) so that developers writing applications for Android could rely on the Java calls with which they were already familiar.

Oracle sued Google in 2010, asserting copyright infringement and other claims, and seeking billions of dollars in damages. A jury heard the copyright claims and found that Google infringed, but deadlocked on Google's fair use defense. The district court judge, however, set aside the infringement verdict, holding that the declaring code at issue—including the Java API's structure, sequence, and organization (SSO)—was not copyrightable pursuant to the idea/expression distinction and the merger doctrine. The U.S. Court of Appeals for the Federal Circuit affirmed in part, reversed in part, and remanded, holding that the declaring code and the API's SSO were copyrightable because neither the idea/expression distinction (Section 102(b)) nor merger applied. On remand, a second jury found that Google's use of the declaring code was fair. Oracle again appealed, and the Federal Circuit again reversed, holding that Google's use of Java's declaring code and the API SSO was not fair as a matter of law.

The Supreme Court granted certiorari to address (1) "[w]hether copyright protection extends to a software interface" and (2) "[w]hether, as the jury found, [Google's] use of a software interface in the context of creating a new computer program constitutes fair use." On its own accord, the Supreme Court ordered supplemental briefing to address a third issue, the appropriate standard of review for a jury verdict on fair use, "including but not limited to the implications of the Seventh Amendment, if any, on that standard." (The Seventh Amendment guarantees the right to a trial by jury in certain civil cases, including copyright cases seeking monetary damages.)

The Court's Decision

By a vote of 6-2, the Supreme Court ruled for Google, relying on the fair use doctrine to conclude that Google's copying did not infringe Oracle's copyright in Java. (Justice Amy Coney Barrett took no part in the decision, presumably because she had not yet joined the Court when the case was argued.) One notable aspect of the Court's opinion is that the majority declined to answer the first question it agreed to hear: whether the Java API's declaring code was copyrightable at all. Citing the "rapidly changing technological, economic, and business-related circumstances" of the software industry, the Court chose to rest its holding solely on fair use without deciding the copyrightability question. Some commentators have speculated that the eight-member Court chose to avoid addressing copyrightability because it could not reach a consensus on the issue. The Court itself states only that "we believe we should not answer more than is necessary to resolve the parties' dispute."

Justice Stephen Breyer's opinion for the majority therefore focused on fair use. First, the Court clarified the standard of review for fair use jury verdicts, explaining that fair use is a mixed question of fact and law, in accordance with the Court's past statements to that effect. The Court acknowledged that analysis of the fair use factors may involve "subsidiary factual questions" (such as market harm) that a jury may resolve, but held that the ultimate conclusion as to whether a use is fair is a question of law that courts may review de novo. The Court thus denied Google's claims that reviewing courts must defer to jury verdicts on fair use and that the Seventh Amendment requires a jury trial on fair use.

The Court then turned to the core of its opinion—an analysis of the fair use factors. As it has done in its prior fair use decisions, the Court emphasized that fair use is a "flexible" doctrine, which courts should approach in a contextual, "case-by-case" manner. In *Google*, the Court explained that fair use's flexibility may "distinguish among technologies" and between the "expressive and functional features of computer code."

Consistent with that approach—but uncommonly for judicial fair use opinions—the Court analyzed the fair use factors out of order, beginning with the second factor (the nature of the copyrighted work). The Court's emphasis on the second factor was also not common, as this factor has sometimes been neglected by courts in the past. Examining the nature of Java's API, Justice Breyer's opinion relied on the highly functional nature of declaring code as weighing in favor of fair use. The opinion found that declaring code, as part of a user interface, is "inherently bound together with uncopyrightable ideas" such as "general task division and organization." As a result, it is "further than are most computer programs (such as the implementing code) from the core of copyright." In other words, assuming that declaring code is copyrightable at all, the Court found that the scope of its copyright protection is narrower (and the scope of fair use accordingly broader).

Turning next to the first fair use factor—the "purpose and character of the use"—the majority found that Google's use was "transformative." Although Google copied the relevant lines of declaring code verbatim, the Court observed that it did so in creating a "new platform"—the Android operating system—and for a "different computing environment" than Java (smartphones, as opposed to desktop computers). Relying on amicus briefs from industry, the Court characterized Google's use as a form of interface "reimplementation," in which another person repurposes the words and syntax of an existing system so that programmers can use their existing skills in a new system. Finding that Google's reimplementation of the Java API was transformative for these reasons, the majority gave less weight to Google's profit-seeking motive, noting that "many common fair uses are indisputably commercial."

As for the third factor, the Court found that the "amount and substantiality" of what Google copied weighed in favor of fair use when viewed in context. Although Google copied 11,500 lines of code in all, the Court found, in light of Google's transformative purpose, that this number should be viewed as a small part (0.4%) of the larger Java API (including the implementing code), which amounted to 2.86 million lines.

On the fourth factor—market effects—the majority emphasized that courts should consider not only the commercial harm to copyright holders, but also the "public benefits" that the copying may produce. In *Google*, the Court deferred to the jury's finding that Android did not harm the market for Java, reasoning that Android was part of a market (smartphones) distinct from Java, as opposed to a direct competitor. The Court maintained that allowing Oracle to enforce its copyright in Java's API in this context would risk "harm to the public." In the Court's view, by the time of Google's copying, the Java API had become valuable less because of its expressive qualities, and more because it was a standard interface that programmers had "already learned to work with." Allowing copyright enforcement in those circumstances would create "a lock limiting the future creativity of new programs" that undermined interoperability and was at odds with copyright's core purpose of facilitating creativity.

In dissent, Justice Clarence Thomas (joined by Justice Samuel Alito) argued that the majority "wrongly sidesteps the principal question" in the case: whether declaring code is copyrightable. The dissent would have squarely held that such code is copyrightable, maintaining that the Copyright Act protects all computer programs without distinguishing between various types of code. As Google conceded, the Java API is original because its authors could have created it in any number of ways (e.g., different organizations, terms, or syntax). Although Google argued that declaring code is an uncopyrightable "method of operation" pursuant to the idea/expression doctrine (Section 102(b)), the dissent asserted that this statutory language cannot be read so broadly. Because the Copyright Act's definition of "computer program" encompasses both declaring and implementing code, the dissent reasoned this specific language must govern over the general language of Section 102(b).

As to fair use, the dissent agreed with the majority that Congress did not intend to categorically shield computer code from the ordinary operation of the fair use doctrine. That said, the dissent argued that the majority's fair use analysis effectively enacts a copyrightability distinction between declaring and implementing code that Congress rejected. Proceeding through the fair use factors, the dissent found that Google's use was "overwhelmingly commercial" and more derivative of Java than transformative; that Google's copying of the API was substantial because it took the "heart" of Java's API; and that Google's actions had "a disastrous effect on Oracle's potential market" for Java. Given that at least three factors weighed against Google, the dissent concluded that the second factor, "the sole factor possibly favoring Google," could not alone support a conclusion of fair use.

Impact for Tech and Other Copyright-Intensive Industries

Reactions to the decision from tech-industry stakeholders were generally positive, if mixed. Many stakeholders expressed relief at the decision, viewing it as vindicating the industry's "longstanding" practice of software reimplementation, at least for reuse of APIs and other interfaces. Although the Court avoided ruling on all potential reuses of declaring code or other interfaces, the dissent observes that the majority's fair use analysis makes it "difficult to imagine any circumstance in which declaring code will remain protected by copyright." Some observers viewed this as a positive development for innovation, facilitating greater interoperability and reuse of interfaces, and encouraging new entrants into software markets. Other commentators viewed the decision as undermining the value of copyright in software and likely to discourage investment by reducing incentives to create interfaces like Java's API.

The decision's effect on other copyright-intensive industries is perhaps less clear. Some stakeholders in the entertainment industry, for example, raised concerns that the *Google* decision will expand the scope of fair use in other contexts, relying on broader notions of what is "transformative" or which uses have "public benefits" to erode protections against unauthorized derivative works. Litigants have already relied on *Google*, for example, in a high-profile copyright dispute concerning certain works by Andy Warhol.

In some ways, the Court sought to limit the scope of its decision, explicitly stating that "we do not overturn or modify our earlier cases involving fair use." If history is a guide, however, the Court's past pronouncements have influenced how lower courts approach fair use. It remains to be seen how aspects of the *Google* decision—such as its emphasis on the second fair use factor, its broad conception of "transformativeness," and its explicit weighing of larger public harms and benefits under the fourth factor—will affect lower court copyright decisions across all types of creative works.

Considerations for Congress

U.S. copyright law is a statutory creation. Congress made the decision in 1980 to recognize copyright in computer programs, and it has the power to change the scope of copyright protection for software, at least prospectively. *Google*'s basic holding assumes that API declaring code is copyrightable, but affords a broad scope for fair uses of declaring code and other user interfaces. If Congress is content with that result, it need not respond legislatively. If Congress disagrees with the Court's holding, it could consider amending the Copyright Act accordingly. (Although Congress has broad authority, such legislation must remain consistent with constitutional limitations—such as the Takings Clause of the Fifth Amendment—and the United States' international-treaty commitments, such as those contained in the Agreement on Trade-Related Aspects of Intellectual Property Rights [TRIPS].)

To the extent Congress believes that the Court did not go far enough in permitting uses of declaring code or other software reimplementations, it could consider amending Section 102(b) (or the Copyright Act's definition of "computer program") to make explicit that declaring code is not copyrightable at all—a step the Court did not take. If, on the contrary, Congress thinks the Court was wrong to conclude that Google did not infringe Oracle's copyright in Java, it could clarify legislatively that declaring code is worthy of greater copyright protection. That end could be accomplished by amending the fair use factors (as Congress did in 1992 regarding unpublished works) or through other statutory amendments.

Author Information

Kevin J. Hickey
Legislative Attorney

Disclaimer