

Google v. Oracle: Should SCOTUS Declare Code is an Expression or an Idea?

March 17, 2020 – *The TMCA*

by Connor J. Hansen and Stefan Szpajda



Related People

- Connor J. Hansen

Related Capabilities

- Trademark, Copyright + Advertising
- Intellectual Property Litigation

In our [prior post](#), we introduced the controversy at the center of “the copyright lawsuit of the decade” between Google and Oracle. Since then, both parties and **61** amici have submitted their briefs to the Supreme Court. This post focuses on the first question before the Court: whether copyright protection extends to the software interfaces underlying the Java coding language. New to this dispute? Don’t worry, we’ll explain what this means!

Terminology and Overview

Let’s begin with the terminology used by the parties, as the parties’ disagreement over what this dispute is about can help frame the issue. [Oracle characterizes](#) the dispute as one re-hashing the well-settled law that *software* is protected by copyright. [Google characterizes](#) the dispute as one asking a novel question: whether discrete portions of software called “declaring code” are protected by copyright. Recall from our last post that the declaring code is the portion of the Java API (application programming interface) that

calls upon pre-programmed implementing code to perform certain functions, which allows programmers to bypass the need to program each function from scratch. When Google made the Android platform, it programmed its own implementing code, but used the same declaring code that Oracle uses in the Java platform. The copied declaring code in Google's Android platform ultimately performs the same function as in other Java applications, albeit through different implementing code. Google refers to the declaring code as a “software interface” to distinguish it from software generally; Oracle responds that the term software interface is something Google made up for this lawsuit.

Google relies on two limitations on the scope of copyright law to argue that the declaring code is not protected by copyright: (1) the declaring code is a method of operation, and (2) under the merger doctrine, the expression of the declaring code merges with the function of the declaring code. Oracle responds by relying on Congressional intent and judicial precedent that have reaffirmed that software can have copyright protection even though all computer code inherently serves some purpose. Each party also invokes several policy arguments, which are highlighted below.

Method of Operation

Under 17 U.S.C. § 102(a), copyright protection extends to original works of authorship, including software generally. But, per § 102(b), copyright does not extend to “any idea, procedure, process, system, method of operation, concept, principle or discovery” embodied in the work. This is commonly referred to as the idea/expression dichotomy. In short, a copyright only protects the particular expression of an idea and not the underlying idea itself.

Google argues that the declaring code serves the purpose of calling upon the pre-written implementing code. In other words, declaring code is not an artistic expression under § 102(a), but is a functional tool for causing a program to perform a certain, known function under § 102(b). As Google puts it, the declaring code and implementing code “create an interface as an exclusively matched set: a key and an unpickable lock.” If Oracle has exclusive rights to the key, then Android developers cannot open the lock to Google’s implementing code.

Oracle stresses that although the declaring code and implementing code work together to perform a function, “Google, or anyone else could write different code in Java—including different declaring code—to perform that exact same function.” Thus, Oracle insists that it is not claiming copyright in the function (the idea), just the exact code (the expression) that Oracle used for that function when designing the Java platform.

Even if the Court were to find the declaring code expressive, however, Google argues that it is not protectable under the merger doctrine.

The Merger Doctrine

The merger doctrine is an application of the idea/expression dichotomy: if an idea can only be expressed in one (or very few) ways, then the expression of the idea “merges”

with the idea and cannot be the subject of copyright protection. The merger doctrine is intended to prevent one entity from claiming ownership over an idea merely because it was the first to discover the only way to express it.

Google has three primary arguments for why Oracle's declaring code merges with the function performed by the declaring code, thus making it ineligible for copyright protection.

First, Oracle has not identified any Java instructions other than the declaring code that can call upon corresponding implementing code; therefore, per Google, the declaring code must be the only way to achieve the function of implementing a pre-programmed function. Oracle is quick to point out, however, that there are no other ways to call upon implementing code because Oracle chose to use the specific declaring code sequences that Google ultimately copied. According to Oracle, a proper inquiry must consider whether the senior user (i.e., Oracle) had more than very few ways to express the work at the time of creation, not whether a junior user (e.g., Google) has but one way to use the senior user's work. When initially designing Java, Oracle argues that its predecessor Sun Microsystems had "unlimited options in writing the declaring code" and the final declaring codes that Google reused are the expression chosen by Sun's programmers.

Second, Google argues that it reused the declaring code because it was the only way developers would know how to call upon Google's pre-programmed functions. In essence, Google argues that it had no choice but to use the declaring code if it wanted downstream developers to know how to write programs for Android. Usually, the merger doctrine would allow a junior user (Google) to use an element of the senior user's (Oracle) work if there were no other way for the junior user to achieve the same goal or purpose as the senior user. Google's argument, however, is that Android programmers (i.e., not Google itself, but Google's customers) would not know how to use Android if Google were not able to reuse Oracle's declaring code. Oracle responds to this by noting that nothing prevented Google from drafting its own declaring code in the same way that Google drafted its own implementing code—in fact, Oracle stresses that both Microsoft and Apple developed smartphone platforms without copying Oracle's declaring code. According to Oracle, the only reason Google needed to copy Oracle's declaring code was to capitalize on the Java platform's popularity: "Once Google decided to copy parts of Java SE that developers already knew, Google 'had no other choice' but to copy Oracle's declaring code."

Third, Google argues that if Oracle were to write the instructions contained in the declaring code in sentence form, there is no doubt that the function described by the instructions would not be protected by copyright. In other words, the declaring code is not creative or expressive, but is instead a rote set of instructions needed to implement a function. Oracle responds by reiterating that it is not attempting to claim copyright in the function at all. Rather, Oracle is claiming copyright in the declaring code's "fully realized expression."

Policy Considerations

Apart from the legal arguments raised above, the copyrightability issue implicates policy questions that could drastically impact the software industry. Appropriately, given the parties' looming presence in Silicon Valley, each side accuses the other of the region's gravest possible transgression: stifling innovation.

Google argues that if copyright protection extends to declaring code, software developers will be dissuaded from releasing code as open source or otherwise allowing others to build upon their developments. Freely building upon existing software is common practice in the software industry and it will be more difficult for innovators to build upon the state of the art if copyright protection extends to discrete, functional portions of software programs. According to Google, this will stifle innovation.

Oracle responds that if copyright protection does not extend to declaring code, software developers will be dissuaded from innovating their own operating systems and coding languages. Rather, they will copy and paste from existing, popular languages. This result will be particularly damaging to startups and small companies because, if anyone is free to copy software, "monopolists and corporate giants" will be able to flood the market with superseding versions of any software that achieves some success. According to Oracle, this will—you guessed it—stifle innovation.

Could it really be that no matter how the Court rules it will stifle innovation? Likely not. Nonetheless, the sheer number and quality of amici this dispute has drawn out suggests that even if the parties are engaged in some measure of hyperbole, the stakes are indeed very high.

What the Amici Are Saying

61 amici have submitted briefs in connection with this case. Of those 61, 44 go to whether copyright protection extends to the software interfaces underlying the Java coding language. Of those 44, 21 support Google and 23 support Oracle. Most of the major companies weighing in on the copyrightability issue—such as IBM and Red Hat Inc.—support Google. Also supporting Google are several legal scholars and organizations, such as a group of 72 Intellectual Property Scholars and the American Antitrust Institute.

Oracle's supporters mostly comprise policy makers, legal scholars, and industry groups. For example, Oracle is supported by a group of Former Congressmen and Ten Creators' Rights Organizations. The United States also supports Oracle's position that software interfaces, including declaring code, are protected by copyright.

It is difficult to draw conclusions as to whether Google or Oracle has the better position based on the support of their respective amici. But the sheer number of amici briefs filed shows that the decision in this case is likely to have significant impact on the software industry and copyright law.

* * *

Oral arguments were scheduled to occur on March 24, 2020, however, the Supreme Court announced on March 16, 2020, that all oral arguments in the March term have been postponed “in keeping with public health precautions recommended in response to COVID-19.” Oral arguments will be rescheduled in due course. In the meantime, be on the lookout for our next update, which will focus on the second issue presented: whether Google’s use of Oracle’s declaring code when creating the Android platform constitutes fair use.