

```

208 limit_val = a;
209 $("#limit_val").a(a);
210 update_slider();
211 function(limit_val);
212 $("#word-list-out").a(" ");
213 var b = k();
214 h();
215 var c = 1(), a = "", d = parseInt(
  parseInt($("#slider_shuffle_num
216 function("LIMIT total:" + d);
217 function("rand:" + f);
  function("c = d, function("check re

```

Google v. Oracle – Supreme Court Issues Landmark Opinion on Fair Use; Sidesteps Interface Copyrightability Questions

Apr 09, 2021

Categories:

[Blogs](#)

[Technology Blog](#)

Authors:

[William S. Morriss](#)

On April 5, the U.S. Supreme Court found that Google's copying of Oracle code was fair use, putting an end to a \$9 billion infringement suit, clarifying or revising several important points of copyright law, and significantly altering the legal rules for software interoperability.

The dispute between Oracle and Google stemmed from Google's desire to (1) make its Android smartphone platform similar enough to Oracle's Java programming language that the significant existing population of Java programmers could write Android applications without learning a new language, but (2) not agree to the terms Oracle insisted on for a license. Google ultimately resolved these conflicting objectives by rewriting everything that actually performed a function that a Java programmer would want performed, but keeping the names and organization of those functions (collectively referred to as an Application Programming Interface or API) so that a Java programmer could use them without having to learn an entirely new system. Oracle was not amused, and sued, alleging that by copying its API Google had infringed its copyrights. Google responded that Oracle's API was not copyrightable and, even if it was, Google's copying was fair use and so could not be a basis for liability. The Supreme Court was presented with an opportunity to speak on both copyrightability and fair use. While it did not explicitly address copyrightability, its discussion of fair use may have made up for that omission.

The Supreme Court began its fair use analysis by focusing on the nature of the work Google had copied and concluded that this favored Google. It explained that, both because an API was "inextricably bound" with uncopyrightable concepts, and because it was a type of work that gained its value from the value of time invested by non-copyright holders (i.e., computer programmers) in learning the API's system, it was (if copyrightable at all, which was

assumed for the purposes of the fair use analysis) “further than are most computer programs ... from the core of copyright.” *Google LLC v. Oracle Am., Inc.*, No. 18–956, 2021 WL 1240906, at *13–14 (U.S. Apr. 5, 2021).

These considerations were then repeatedly invoked as the Supreme Court found that the fair use factors actually favored Google. It explained that Google’s use was transformative because it sought to create new products by providing a new platform that programmers could readily use. *Id.* at *15. It explained that the amount and substantiality of the portion used favored Google because its copying was needed to permit programmers use their knowledge and experience when they wrote new programs for Android smartphones. *Id.* at *17. It explained that the market effects also favored Google, at least in part because it was necessary to weigh the benefits of new expression Google’s copying was likely to produce and allowing Oracle to prevent others from using its API in their platforms could effectively lock up the creativity of the programmers who had invested their time and energy in learning Java. *Id.* at *17, 19. Given that the fair use factors uniformly favored Google, the Supreme Court reversed the circuit court’s determination in favor of Oracle and remanded for further proceedings in conformity with its opinion.

While the Supreme Court did not address the question of copyrightability of APIs, its treatment of fair use may ultimately have rendered that question irrelevant. This case was not the first time that courts have faced the question of copyrightability of an interface. A quarter-century ago, the First Circuit grappled with this same issue in *Lotus Development Corp. v. Borland Int’l, Inc.* In that case, Lotus had created a set of commands and organized them into a menu hierarchy that users could take advantage of to write small computer programs called macros. *Lotus Development Corp. v. Borland Int’l, Inc.*, 49 F.3d 807, 809 (1st Cir. 1995).

Just as there was no question that Google had copied the names and organization of Oracle’s functions, in *Lotus* “Borland d[id] not dispute that it factually copied the words and arrangement of the Lotus menu command hierarchy.” *Id.* at 812. Instead, the only question was whether that copying could give rise to liability and,

while the First Circuit relied on copyrightability rather than fair use, its underlying concern was the same – i.e., that allowing the copyright holder to monopolize the commands used to interface with its programs would effectively lock up users who had learned and written macros for the Lotus 1-2-3 software. *Id.* at 818. Given that this issue is essentially inherent for any kind of software interface, it is, as Justice Clarence Thomas noted in dissent, “difficult to imagine any circumstance in which [the type of code copied by Google] will remain protected by copyright.” *Google LLC v. Oracle Am., Inc.*, No. 18-956, 2021 WL 1240906 at *23.

While the Supreme Court still has not spoken on the copyrightability of APIs, its opinion in *Google v. Oracle* may nonetheless send a message that creators of APIs would be wise not to rely on the copyright system for protection. For further information, please contact [William Morriss](#) or any other member of Frost Brown Todd’s [Intellectual Property Practice Group](#).

© 2026 FBT Gibbons LLP. May be considered attorney advertising in some jurisdictions.