No. 2013-1021, -1022

IN THE

United States Court of Appeals
for the Federal Circuit

ORACLE AMERICA, INC.,

Plaintiff-Appellant,

v.

GOOGLE INC.,

Defendant-Cross Appellant.

ON APPEAL FROM THE UNITED STATES DISTRICT COURT FOR THE NORTHERN
DISTRICT OF CALIFORNIA, CASE NO. 10-CV-3561, HON. WILLIAM H. ALSUP

RESPONSE AND REPLY BRIEF OF PLAINTIFF-APPELLANT

Dorian Daley
Deborah K. Miller
Matthew Sarboraria
Andrew C. Temkin
ORACLE AMERICA, INC.
500 Oracle Parkway
Redwood Shores, CA 94065

E. Joshua Rosenkranz
Mark S. Davies
Andrew D. Silverman
Kelly M. Daley
ORRICK, HERRINGTON & SUTCLIFFE LLP
51 West 52nd Street
New York, NY 10019
(212) 506-5000

Annette L. Hurst
Gabriel M. Ramsey
Elizabeth C. McBride
ORRICK, HERRINGTON & SUTCLIFFE LLP
405 Howard Street
San Francisco, CA 94105-2669

Dale M. Cendali
Diana M. Torres
Sean B. Fernandes
Joshua L. Simmons
KIRKLAND & ELLIS LLP
601 Lexington Avenue
New York, NY 10022

Attorneys for Plaintiff-Appellant

CERTIFICATE OF INTEREST

Counsel for appellant certifies the following:

1.   We represent Oracle America, Inc.

2.   That is the real name of the real party in interest.

3.   Oracle America, Inc., is a wholly owned subsidiary of Oracle Corporation.

4.   The following law firms and partners or associates appeared for Oracle America, Inc., in the Northern District of California or are expected to appear in this court:

ORRICK, HERRINGTON & SUTCLIFFE LLP:

E. Joshua Rosenkranz
Mark S. Davies
Annette L. Hurst
Gabriel M. Ramsey
Andrew D. Silverman
Kelly M. Daley
Elizabeth C. McBride

KIRKLAND & ELLIS LLP

Dale M. Cendali
Susan Davies
Diana M. Torres
Sean Fernandes
Joshua L. Simmons

MORRISON & FOERSTER LLP

Michael A. Jacobs
Kenneth A. Kuwayti
Marc David Peters
Daniel P. Muino
Mark E. Ungerman
(Richard S. Ballinger)*
Roman A. Swoopes
(Ruchika Agrawal)
Rudolph Kim
Yuka Teraguchi

BOIES, SCHILLER & FLEXNER LLP

David Boies
Steven C. Holtzman
Alanna Rutherford
Beko Osiris Ra Reblitz-Richardson
Meredith R. Dearborn
(William F. Norton, Jr.)

Date: July 3, 2013   By: /s/ E. Joshua Rosenkranz
                                Attorney for Plaintiff-Appellant

---

* Parentheses indicate that the attorney is no longer with the firm.

TABLE OF CONTENTS

TABLE OF AUTHORITIES

Page(s)

FEDERAL CASES

vii

FEDERAL CONSTITUTION, STATUTES AND REGULATIONS

LEGISLATIVE HISTORY

MISCELLANEOUS

REPLY ARGUMENT[1]

Google does not dispute that it copied Oracle's work as wantonly as our hypothetical Ann Droid copied *Harry Potter*. It does not dispute that the work it copied was, in its realm, as creative as a novel: Without a hint of dissent, Google parrots Oracle's demonstration that the copied work was "'original,' … 'intuitive,' 'attractive,' 'appealing,' 'intricate,' 'efficient,' and 'user-friendly.'" GB 5 (OB citations omitted). Nor does it dispute that its defenses are equivalent to Ann's defenses, "But I wrote most of the words from scratch," and "I copied only the portions necessary to tap into the Harry Potter fan base." And it agrees that Ann could never get away with those defenses. But Google should get away with those same defenses, it insists, for one simple reason: Oracle's work—"[h]owever creative and useful"—is not a "work of imaginative fiction like Harry Potter," but a computer program that "is fundamentally a functional, utilitarian work." GB 1.

Google concedes that § 102 of the Copyright Act, which defines copyrightability, codifies the traditional dichotomy between ideas and

---

[1] For ease of reference: "OB" is Oracle's Opening Brief, "GB" is Google's Brief, and amicus briefs will be cited as "___ Br.," according to name or abbreviation of the lead amicus.

expression. But it avoids the consequence of that concession: an original work—even one that has a function—is entitled to copyright protection so long as the author had multiple possible ways to express an idea. Instead, Google asserts that if a computer program—or any other literary work—is functional, it loses protection no matter how creative it is. Google never explains how any software can be protected under its view, since all computer programs are "fundamentally a functional, utilitarian work." GB 1.

Google only emphasizes its lack of case support by filling its brief with quotations from the writings of an academic who has, for decades, argued for *changes* in the law on the ground that Congress never should have granted copyright protection to software. Beyond that, Google's effort to portray its position as Ninth Circuit law revolves around a single sentence plucked out of context from *Sega Enterprises Ltd. v. Accolade, Inc.*, 977 F.2d 1510 (1993). Google ignores the section of *Sega* that held that creative software *is* protectable, relying instead on the fair-use analysis which has nothing to do with copyrightability.

As eager as Google is to import fair-use principles into copyrightability, it offers only the most cursory response to Oracle's

argument that the copying was unfair as a matter of law. Google copied the most useful portions of Oracle's work for commercial purposes and derailed Oracle's already-thriving business of licensing Java for mobile devices. Google tries to portray this as fair by cobbling together snippets of the trial record into a narrative that is itself a work of "imaginative fiction," a story the district court already rejected. A24,654-56.

For example, Google repeatedly asserts that Oracle's software packages were "open and free for anyone to use." GB 9-10, 16, 34. That is nothing but a play on words. As this Court recognizes, that is not what "open source" means. Open-source licenses "are used by … software developers … who wish to create collaborative projects and to dedicate certain works to the public," while, at the same time, "provid[ing] creators of copyrighted materials a means to *protect and control their copyrights*." *Jacobson v. Katzer*, 535 F.3d 1373, 1378 (2008) (emphasis added). Google does not dispute that every other business that wanted to put Oracle's Java packages to commercial use—including businesses that wanted to use only the declaring code— took licenses. OB 2, 15-16, 76-77.

Equally fictitious is Google's protestation that it was shocked to learn of the "Java copyrights." GB 17 & n.79. The feigned surprise comes at the end of a paragraph acknowledging that Google was in intense negotiations about "taking a license" to the very packages it copied. GB 17. And nowhere does Google dispute that its head of Android reported up the chain of command that the Java packages "are copyrighted[,] [a]nd Sun gets to say who they license … to," A1200; *see* A2689; OB 18-20, or that Google considered "buy[ing] the rights to Java from Sun (patents, *copyrights*, etc[.])," A2191 (emphasis added). Google has more copyright lawyers than an AIPLA conference. It did not need a tutorial from Oracle to know that it was unlawful to plagiarize.

To take one final example, Google asserts that Sun welcomed "Android with open arms." GB 20. In support, Google cites a blog post. Google does not suggest that its platoons of copyright lawyers mistook the post for the elusive license it had tried for years to secure. In any event, the post was written *before* Google first released Android's code to outside programmers. A2198, 5828-31, 6541. Until that release, Oracle "presumed" that Google either (1) used "GPL [General Public License] code" and would donate the code back to Oracle under the

open-source terms of the GPL license, A22,198; OB 14 (describing GPL), or (2) would take a commercial license, A2685, 2688-90, 5828-30.

Fairly portrayed, the undisputed record confirms that there was nothing fair about Google's copying.

I.   COPYRIGHT LAW PROTECTS ORACLE'S SOFTWARE PACKAGES.

This case is about copyright protection for computer software—here, the particular declaring code and organization Oracle used in Java SE. We warned of the mischief from using the verbal chameleon "API" to describe these packages of code. OB 9. Google and its amici exploit the ambiguity for maximum advantage, accusing Oracle of imperiling anything that might be called an API, including "interfaces," GB 52, "access protocols," "interface specifications," CCIA Br. 3, 12-13, "network sockets," "rules for how programs communicate within a single computer," "firmware," and "hardware," Computer Scientists Br. 6-7, 12-15, 23, 31; *see* Start-ups Br. 3, 10. This case is not about those abstractions or devices. It is only about protection for intricate and highly creative packages of software that Oracle authored.

As to the packages actually before this Court, Google concedes away all that matters—at least under established copyright principles.

§ I.A. Unable to prevail under settled principles, Google concocts a

veritable mash of copyright law, consisting of large helpings of fair-use

doctrine and infringement principles but only a trace of copyrightability

law. The concoction bears no relation to the law Congress adopted, as

interpreted by the Supreme Court, and just about every circuit,

including the Ninth Circuit and this Court.

Google's position rests on several flawed premises. First, it rests

on a reading of § 102 that the Supreme Court rejects. § I.B. Second,

Google incorrectly asserts that the fair-use analyses of *Sega* and *Sony*

Computer Entertainment, Inc. v. Connectix Corp., 203 F.3d 596 (9th Cir.

2000), dispose of the case. § I.C. Then Google makes merger and

interoperability arguments that have no bearing on the copyrightability

of the original work in question (Java packages). § I.D. Finally, Google

resorts to a meritless waiver argument, § I.E, and to policy arguments

that are best addressed to Congress, § I.F.

A.   **Google Concedes Away The Entire Case, At Least
Under Established Copyright Principles, And Declines
To Rebut Half The Copyrightability Arguments.**

Google concedes away—either explicitly or tacitly—everything

necessary to resolve this case under established copyright principles.

To start, Google concedes that it literally copied "7,000 lines of declaring code," GB 63; *accord* GB 18, and does not dispute that "the structure, sequence and organization of the 37 accused API packages in Android is substantially the same as the structure, sequence and organization of the corresponding 37 API packages in Java," A985, 22,771-72 (cited by OB 43).

These concessions collapse the challenge to the district court's ruling into a single question: Is there *any protected expression at all* in either the copied code or the copied structure? *See Feist Publ'ns, Inc. v. Rural Tel. Serv. Co.*, 499 U.S. 340, 361 (1991). Google does not dispute that copyright protection extends to computer programs—both to the literal elements and to structure and organization. *See* GB 2, 37, 42-44; OB 4, 31-32, 41, 44-45. Google then makes a key—indeed, a dispositive—concession: that the declaring code and the structure and organization of the packages are original under § 102. GB 5, 29-30. Google also does not challenge—and even concedes—the district court's finding that the declaring code could have been written in any number of ways "and still have worked," A132; *accord* A133, 140-41, and that the packages could have been organized in "many ways … [and] still

7

duplicate[d] the same range of functionality," A133. *See* A24,670

(Google's concession); *see also* OB 32-33, 49-52, 60-61. Indeed, Google

goes one step further: It acknowledges Oracle's extensive showing that

the packages are "'original,' 'creative,' 'intuitive,' 'attractive,' 'appealing,'

'intricate,' 'efficient,' and 'user-friendly.'" GB 5 (OB cites omitted); *see*

OB 12-13, 30 (district court's finding), 38-40, 40-43 (declaring code), 43-

48 (structure and organization); McNealy Br. 10-20. But Google does

not dispute a word of that showing.

As our opening brief demonstrated (at 40-66), these concessions

are dispositive under established copyright principles. Copyright

protects original expression. *Feist*, 499 U.S. at 345. The protection

persists even if the original expression *also* performs a function or is

contained within a method of operation. Under the classic

idea/expression dichotomy that the Supreme Court famously articulated

in *Baker v. Selden*, 101 U.S. 99 (1880), the only time expression is not

protectable is if the author's expression is the only possible way to

achieve the result, in which case the idea and the expression merge and

protecting the expression is tantamount to protecting the idea. OB 32-

33, 48-49. But so long as there are multiple ways to express that

function or perform that process or method, the work is protectable.
OB 49-53.

Our opening brief documented all these points at length and with numerous citations to caselaw. Google does not suggest that Oracle misquoted these authorities or otherwise misstated established copyright law. Yet, beyond paying occasional lip service to some of these principles, *see, e.g.,* GB 47-53, Google ignores them along with the cases—from this Court, the Ninth Circuit, and numerous others—that articulate and apply them to computer programs.

Google's silent treatment extends to more than just established copyright principles. Google also makes the startling strategic decision to ignore half of Oracle's copyrightability argument and to abandon any effort to defend the keystone of the district court's copyrightability decision. As explained, Oracle presses "two distinct bases" for finding copyrightability: "(1) the expressive declaring code; and (2) the creative arrangement of each package." OB 40. Google largely ignores the first. As if to lampoon its own stance, Google asserts that it "did not use" any of Oracle's "underlying computer code … —unless one counts the 7,000 lines of declaring code" that Google copied. GB 63 (internal quotation

marks omitted). They count. Front and center in the opening brief was *Harper & Row, Publishers, Inc. v. Nation Enterprises*, 471 U.S. 539 (1985), the dominant Supreme Court case on "admitted" plagiarism. OB 40-41. Google does not even cite, much less discuss, this case.

Similarly, a centerpiece of the district court's copyrightability holding was that declaring code is unprotected as short phrases. A143-44, 162, 164-65. Google leaves a lengthy attack on that holding unrebutted. OB 53-57; *see* GB 68. Oracle demonstrated that the regulation covering short phrases prohibits protection for a "work" that consists of nothing but a naked phrase but does not remove protection for an assemblage of 7000 lines of code. OB 53-55 (collecting cases). Google does not disagree. Oracle demonstrated that much of the declaring code reads more like run-on sentences than short phrases. OB 56. No response. Quoting Google's Java guru, Oracle demonstrated that there can be—and is here—"creativity and artistry in a single method declaration." OB 56-57. Again, nothing from Google.

Google buries its excuse in the recesses of its brief (at 67): It asserts that the district court wasted its time in analyzing these issues after the verdict, because they were waived at trial. That is as absurd

as it sounds and, for reasons explained below (at 39-42), demonstrably false. For present purposes, suffice it to say that if Google is wrong about waiver, this Court can reverse without reading another page of this copyrightability argument (and jump straight to fair use).

B.   The Traditional Understanding Of § 102(b) Does Not Strip Creative Computer Programs Of Copyright Protection.

Google urges a copyright construct that flouts every established copyright principle described above. The crux of Google's argument is that the protection § 102(a) grants with one hand, § 102(b) withdraws with the other, because the copied expression, though original and creative, is a "functional element." GB 2.

Section 102 provides in relevant part:

> (a) Copyright protection subsists … in original works of authorship …. Works of authorship include … literary works ….
>
> (b) In no case does copyright protection for an original work of authorship extend to any idea, procedure, process, system, method of operation, concept, principle, or discovery ….

17 U.S.C. § 102. The parties present two starkly different readings of § 102. Oracle's reading is supported by the overwhelming weight of

11

authority. In support of its alternative reading, Google cites mainly law review articles.

Oracle's reading of § 102. Section 102 codifies the traditional idea/expression dichotomy described above and the merger doctrine that flows from it. OB 59-63. At one point, Google agrees, conceding that § "102(b) codifies the Supreme Court's landmark holding in *Baker v. Selden.*" GB 31; *accord* GB 33. This is undoubtedly correct. Explains the Supreme Court: "'Section 102(b) in no way enlarges or contracts the scope of copyright protection under the present law. Its purpose is to restate[, in the context of the new single Federal system of copyright,] that the basic dichotomy between expression and idea remains unchanged.'" *Feist*, 499 U.S. at 356 (bracketed text omitted in *Feist*) (quoting H.R. Rep. No. 94-1476, at 57 (1976) *reprinted in* 1976 U.S.C.C.A.N. at 5670, and S. Rep. No. 94-473, at 54 (1975) *available at* 1975 WL 370212). In other words, § "102(b) is intended … to make clear that the *expression* adopted by the programmer is the copyrightable element in a computer program" while "the actual processes or methods embodied in the program are not." H.R. Rep. No. 94-1476, at 57.

Again, Google seems to agree: Quoting the same congressional
reports relied on by *Feist* above, Google argues that § "102(b) …
'make[s] clear' that 'the *writing expressing a programmer's ideas*'—that
is, *the code*—is 'the copyrightable element in a computer program,'
while 'the actual processes or methods embodied in the program are not
within the scope of the copyright law.'" GB 30-31 (alterations omitted;
emphasis altered); *accord* OB 59-60. Exactly so. "[T]he writing
expressing a programmer's ideas" is "the code"—here, the declaring
code—which is protectable.

Google never acknowledges the consequence of these concessions.
This Court's decision in *Atari Games Corp. v. Nintendo of America, Inc.*,
975 F.2d 832 (1992) (discussed at length in our opening brief (at 41-42,
44-45, 49, 60)) is instructive. *Atari* involved the copyrightability of
Nintendo's program to unlock its gaming console by transmitting a
unique message from the game to the console. Under *Atari*, the "idea"
of a particular method or class is whatever function it is designed to
perform (like sending a data signal, 975 F.2d at 840, or opening an
internet connection, OB 9-10). Oracle cannot copyright the idea of
programs that open an internet connection. But it can copyright the

13

precise strings of code used to do so, at least so long as "other language *is* available" to achieve the same function. National Commission on New Technological Uses of Copyrighted Works, Final Report ("CONTU Report") at 20 (1979); *accord Atari*, 975 F.2d at 840 (no merger when "alternative expressions are available"); *see* OB 32-34, 48-52, 59-62.

Critically, Google does not suggest that Oracle is trying to protect the *idea* of any of its packages. Nor could it, in light of the district court's unchallenged finding that the declaring code could have been written and organized in any number of ways and still achieved all the same functions. *Supra* at 7-8 (citing A132-33, 140-41; OB 32-33, 49-52, 60-61). Accordingly, under the principles Google *claims* to embrace, § 102(b) does not strip the highly expressive and creative packages and their declaring code of protection just because they *also* perform functions. *See also* Oman (former Copyright Register) Br. 7.

Google's reading of § 102. Despite its concessions, Google recasts § 102 in a way that bears no resemblance to *Baker*'s idea/expression distinction, traditional merger analysis, or any other longstanding copyright norm. Google maintains that § 102 requires a "two-stage copyrightability analysis." GB 29. In step (a), you assess

14

"[o]riginality and creativity." GB 45. But no matter how original and creative a work, Google maintains, step (b) lists a series of "exclusions" that withdraw protection to "functional and factual aspects of an otherwise copyrightable work." GB 30. Thus, according to Google, the packages would receive copyright protection under subsection (a) because they are "original," but subsection (b) would then extinguish every trace of copyright protection because the code is *also* functional. Under Google's reading, unlike in established merger analysis, it matters not that there were infinite ways in which Oracle could have written and organized the declaring code to express the thousands of functions performed by the packages. GB 45. All that matters to Google is that the code performs a function.

If Google's reading is right, then Congress played a trick on the software industry. Congress defined "literary work" to encompass computer programs, thereby granting protection to "original" software at step (a). But it then withdrew protection from all software at step (b), because computer code always performs a function. OB 61-62. And to guarantee that no software would ever survive step (b), Congress defined "computer program" in functional terms: as a "set of statements

or instructions [that are] used directly or indirectly in a computer *in order to bring about a certain result*." 17 U.S.C. § 101 (emphasis added). Google does not try to suggest a limiting principle that would strip only Oracle's declaring code of protection but somehow leave other software intact. The most Google (and its amici) can muster is the tepid assurance that Oracle's implementing code "may be copyrightable." GB 65 n.165; *see* Rackspace Br. 4. But since implementing code is also "functional[]," GB 13 n.49, there is no distinction. Thus, in one fell swoop, Google's statutory construct wipes out the unanimous view of the courts and Congress that software has copyright protection. OB 31-32, 37, 41, 44-45 (collecting cases); *see* Oman Br. 7-8, 14-17.

Actually, if Google is right, the trick is on just about every author of any literary work. Google insists that "the same rules of copyrightability" apply to "all works." GB 42. So, any original work that is also "functional" or "utilitarian" loses copyright protection: encyclopedias, textbooks, *The Bluebook*, instruction manuals, maps, car value guides, dental and medical taxonomies, and any other work ("[h]owever creative") that is also "functional" or "utilitarian." GB 1; *see Am. Dental Ass'n v. Delta Dental Plans Ass'n*, 126 F.3d 977, 978 (7th

Cir. 1997). Moreover, the way § 102 is structured, what is true of a "method of operation" must be true of "any idea, … concept, [or] principle." So by Google's reading, every "original" work would get protection at step (a), but would lose it in step (b) if it *also* expresses "any idea."

Google's proposed hermetic division between subsections (a) and (b) is wrong. Over 50 years ago, the Supreme Court observed: "We find nothing in the copyright statute to support the argument that the intended use or use in industry of an article eligible for copyright bars or invalidates its registration. We do not read such a limitation into the copyright law." *Mazer v. Stein*, 347 U.S. 201, 218 (1954). The same is evident from the CONTU Report, whose recommendations Congress adopted "almost verbatim" and which is considered "the authoritative guide to congressional intent." *Sega*, 977 F.2d at 1519 n.5. Invoking *Baker*, CONTU rejected Google's position, specifically as to software: "Nor has copyright been denied to works simply because of their utilitarian aspects…. *That the words of a program are used ultimately in the implementation of a process should in no way affect their*

17

copyrightability."   CONTU Report at 21 (emphasis added); *see* BSA Br. 8-11 (discussing CONTU Report).

Google does not even try to reconcile its two-step analysis with the authorities cited above and in the opening brief granting full copyright protection to computer code even though it performs functions. OB 41-45, 49, 60. If Google's two-step approach were the law, this Court in *Atari* would not have held the 10NES protectable because it served the *function* of locking and "unlock[ing] the NES console." 975 F.2d at 840. And *Apple Computer, Inc. v. Franklin Computer Corp.* would not have granted copyright protection to "operating system program[s]" that performed functions, including "allow[ing] data to be passed between different parts of a program," "turn[ing] on the [computer's] circuits," "translat[ing] instructions," and "control[ing] the reading and writing functions of the disks." 714 F.2d 1240, 1244 n.4 (3d Cir. 1983); *see Johnson Controls, Inc. v. Phoenix Control Sys., Inc.*, 886 F.2d 1173, 1174 (9th Cir. 1989).

Again and again, courts find computer programs copyrightable— despite their "utilitarian" or "functional" purpose—because the developers had "some discretion and opportunity for creativity … in the

18

structure" of the program. *Johnson Controls*, 886 F.2d at 1176. Thus, courts routinely read *Baker*'s idea/expression principles to find programs protectable simply because their authors seek protection of their own expression—their own way of implementing the function, their own "expressi[ve]" "structure, sequence, and organization of the [program]," *id.* at 1175-76—and not the function itself. *Atari*, 975 F.2d at 839; *accord Atari Games Corp. v. Oman*, 888 F.2d 878, 885 (D.C. Cir. 1989) (Ginsburg, J.) ("variety of ways to perform the same function sustains the classification of such works as 'expression'"); *Mitel, Inc. v. Iqtel, Inc.*, 124 F.3d 1366, 1372 (10th Cir. 1997) ("Section 102(b) does not extinguish the protection accorded a particular expression of an idea merely because that expression is embodied in a method of operation at a higher level of abstraction"; thus, "an element … may be characterized as a method of operation, … [and] nevertheless contain expression that is eligible for copyright protection"); *Toro Co. v. R & R Prods. Co.*, 787 F.2d 1208, 1212 (8th Cir. 1986) (§ "102(b) does not answer the question of whether [plaintiff's] particular expression of that idea is copyrightable"); *see also* Oman Br. 3, 14-15, 18.

19

Only one case has ever so much as suggested "that expression that is part of a 'method of operation' cannot be copyrighted." *Lotus Dev. Corp. v. Borland Int'l Inc.*, 49 F.3d 807, 818 (1st Cir. 1995), *aff'd by an equally divided Court*, 516 U.S. 233 (1996). It is hard to believe that *Lotus* actually meant to be quite that expansive, particularly in light of the Supreme Court's opinion in *Feist*. But if it did, the case cannot overcome this Court's opposite conclusion, interpreting Ninth Circuit law, that "expression of [a] process or method" *is* protectable. *Atari*, 975 F.2d at 839.

The statement in *Lotus* quoted above elicited howls of protest among commentators, the most vocal of whom is now Google's Senior Copyright Counsel, William Patry,[2] who lambasted the case for the shallowness of its logic (e.g., "not even bother[ing] to cite the definition [of computer program]"); "[i]gnoring Congress's judgment and the objective standard of originality"; and making "all computer programs … unprotectable methods of operation." William F. Patry, *Copyright and Computer Programs: It's All in the Definition*, 14 Cardozo Arts &

---

[2] Http://en.wikipedia.org/wiki/William_F._Patry (last accessed, July 2, 2013).

Ent. L. J. 1, 5-8, 13, 59-63 (1996) (footnote omitted)); *see* Paul I.

Kravetz, *"Idea/Expression Dichotomy" and "Method of Operation":*

Determining Copyright Protection for Computer Programs, 8 DePaul

Bus. L. J. 75, 77, 97-101 (1995) (predicting that *Lotus* "will seriously

impact the software industry").

While Google contends (at 3, 61) that the Ninth Circuit has

embraced *Lotus*, the truth is that, in the 18 years since *Lotus*, the Ninth

Circuit has cited *Lotus* only once—as subsequent history on a

procedural issue (delay in filing). In fact, as demonstrated below (at

24-25), the two Ninth Circuit cases that Google relies upon most—*Sony*

and *Sega*—tacitly reject *Lotus* by finding the software copyrighted. *See*

Lotus, 49 F.3d at 821 (Boudin, J., concurring) (discussing dueling

options to address copyright for software). When presented with the

opportunity, the Ninth Circuit will undoubtedly embrace this Court's

reading of Ninth Circuit law and explicitly reject *Lotus*, like the Tenth

Circuit. *Mitel*, 124 F.3d at 1372; *see* OB 62-63.

But this Court need not join the chorus of *Lotus* detractors here.

It can simply distinguish *Lotus*. First, Google copied Oracle's code

verbatim, but, as Google concedes (at 58, 63), the *Lotus* defendant did

21

not, 49 F.3d at 810. Second, *Lotus* found that the rudimentary

commands that were copied ("copy," "print," etc.) were not creative. *Id.*

at 809, 817; *accord id.* at 821 (Boudin, J., concurring). By contrast, the

declaring code and the structure and organization here are

undisputedly creative and original. GB 5, 29-30. Third, while the

commands in *Lotus* were "essential to operating Lotus," *id.* at 817, the

same is not true of the vast majority of what Google copied, *infra* at 32.

Perhaps the best evidence of how far Google's reading of § 102

strays from the caselaw lies in how heavily Google relies on articles—

specifically, the writings of one author, Professor Pamela Samuelson (on

this topic, GB 30-33, and others, GB 40, 51). *See also* Law Professor Br.

7-8, 10. Google cribs so much from this one academic that she should

demand royalties. The professor "firmly believes" that Congress was

wrong to grant copyright protection to computer programs and should

have created "a new form of intellectual property law." Pamela

Samuelson, *CONTU Revisited: The Case Against Copyright Protection*

for Computer Programs in Machine-Readable Form, 1984 Duke L.J.

663, 764 (1984). That is what Google is pressing as well: a new form of

intellectual property law that, in the unanimous view of the Supreme
Court, the Ninth Circuit, and this Court, Congress has rejected.

> **C.   *Sega*'s and *Sony*'s Fair-Use Analysis And Other Cases Considering Infringement Have No Bearing On Whether Oracle's Software Packages Are Copyrightable.**

Google's § 102 argument, however flawed, at least addresses the
right issue: copyrightability. The same cannot be said of the rest of
Google's argument, particularly its focus on fair-use and infringement
principles in *Sega* and *Sony*.

Sega *and* Sony *on fair use*. A keystone of Google's argument is
that "*Sega* and *Sony* dispose of Oracle's key [copyrightability]
arguments." GB 40. Google cites those two cases—*Sega* in particular—
on virtually every page of its 27-page copyrightability discussion,
GB 28, 31, 33-34, 36-44, 46, 48-49, 52, and quotes one isolated sentence
in *Sega* six times, GB 3, 36, 38, 42-43, 50, 52. But those portions of the
opinions discuss fair use, not copyrightability, and Google ignores what
those cases teach about copyrightability.

Sega sold video games to run on its own consoles. A competitor
(Accolade) wanted to distribute its own games to run on Sega's console.
So it made copies of the software in Sega's games to learn how they

worked and to figure out how to make the console play Accolade's games. 977 F.2d at 1514-15, 1523-27. In *Sony*, a competitor (Connectix) wanted to create a program that could play PlayStation games on computers (not just on PlayStation consoles). So Connectix copied chunks of Sony's code to learn how the console functioned. 203 F.3d at 593. In each case, the infringement allegation centered around the copies these competitors made of the games' software for research and development purposes. And in each, the defendant asserted that making those copies was fair use.

Before ever getting to fair use, *Sega* discussed at length the threshold question whether software is copyrightable. Google omits that analysis. *Sega* considered—and *rejected*—the "argument that object code is not eligible for the full range of copyright protection." 977 F.2d at 1519. Discussing § 102(a), *Sega* recognized that Congress "unambiguously extended copyright protection to computer programs." *Id.* "The statutory language, read together with the CONTU report, leads inexorably to the conclusion that the copyright in a computer program extends to the object code version of the program." *Id.* at 1520. Accordingly, *Sega* held that the code Accolade "copied [was] protected

24

expression." *Id.* at 1524-25. In so concluding, *Sega* rejected "scholarly authority" "that object code is not [protectable]," *id.* at 1519, including some of the same commentators who now appear before this Court, *see* Law Professors Amicus Br. (including Samuelson) in *Sega v. Accolade*, *republished at* 33 Jurimetrics J. 147, 153-54 n.17 (1992-1993) (citing Prof. Samuelson in advocating against protection for object code).

In *Sony*, the copied code was so clearly copyrightable that Connectix did not contest the point. Still, the court never would have addressed fair use if it believed the code was unprotectable, since fair use applies only to "*copyrighted* work." 17 U.S.C. § 107 (emphasis added).

Ignoring all this, Google fixates on the opinions' fair-use analyses, characterizing *them* as "controlling" on "copyrightability." GB 29. Google justifies the ploy on the ground that the "fair-use rulings were predicated and dependent upon fully considered holdings that the compatibility elements are not copyrightable under section 102(b)." GB 39 (emphasis omitted). That is where Google's favorite sentence comes in: "[C]omputer programs ... contain many logical, structural, and visual display *elements* that are dictated by the function to be

25

performed, by considerations of efficiency, or by external factors such as compatibility requirements and industry demands." *Sega*, 977 F.2d at 1524 (emphasis added).

Nothing in this sentence in any way contradicts the point, explained above (at 12-20), that functional works are protectable so long as there are alternative ways to achieve that same function. Nor does it change the law (explained at OB 52-53, 63-65, and addressed below, at 34-37) that, for the copyrightability analysis, external factors such as compatibility requirements and industry standards are measured at the time of creation of the original work (here, Java), and no such considerations constrained the original Java developers.

In any event, nothing in that sentence, or the rest of *Sega*, can be read as suggesting that an intricate array of 7000 lines of concededly original and creative computer code, GB 5, 29-30, is completely devoid of copyright protection—especially when *Sega* already found that the software in question was entitled to copyright protection. 977 F.2d at 1519-20, 1525. Saying that certain "*elements* of computer programs are not copyrightable," GB 39 (emphasis added), is not the same as saying that the *entire work* loses all copyright protection. As our opening brief

26

explains (at 46), but Google never addresses, the Supreme Court has

held that even if a work "contains absolutely no protectable written

expression," the original "selection or arrangement" of the unprotectable

elements is protected so long as it "entail[s] a minimal degree of

creativity," *Feist*, 499 U.S. at 348, which is already conceded here, GB 5,

29-30.

The rest of *Sega*'s fair-use discussion confirms the point. Google's

favorite sentence was the culmination of the court's analysis of the

"second [fair use] factor, the nature of the copyrighted work." 977 F.2d

at 1524. The discussion concerned "the *extent* of copyright protection,"

id. (emphasis added)—not whether the entire program was unworthy of

copyright protection. And *Sega* acknowledged that "the programmer's

choice of program structure and design may be highly creative and

idiosyncratic," *id.*, as Oracle's structures are, even though "computer

programs are, in essence, utilitarian articles—articles that accomplish

tasks." *Id.*[3]

---

[3] *Sony* adds nothing more. It reaches the same holding: "We
conclude that, under the facts of this case and our precedent,
[defendant's] intermediate copying and use of [plaintiff's] copyrighted
[work] was a fair use for the purpose of gaining access to the
unprotected elements of [plaintiff's work]." 203 F.3d at 602.

In the end, to read *Sega*'s and *Sony*'s fair-use analyses as Google does—removing protection for a work just because it has functional elements—would require this Court to conclude that those fair-use cases silently rejected the Supreme Court's teachings in *Feist* and *Mazer*; the CONTU Report's analysis of the interaction between § 102(b) and computer programs; the Ninth Circuit's teaching that "the structure, sequence and organization" "of computer software may be protected by copyright where they constitute expression," *Johnson Controls*, 886 F.2d at 1175, 1177; and this Court's interpretation of Ninth Circuit law in *Atari*. Worse, it would require this Court to accept that those two cases somehow preempted more recent Ninth Circuit opinions recognizing that "source and object codes[] can be subject to copyright protection," *JustMed, Inc. v. Byce*, 600 F.3d 1118, 1125 n.3 (9th Cir. 2010), as can the "original selection and arrangement" of computer programs without any reference to their functionality, *Apple Computer, Inc. v. Microsoft Corp.*, 35 F.3d 1435, 1445 (9th Cir. 1994) (quotation marks omitted). These fair-use analyses cannot be stretched that far.

Infringement principles. Google also littered its brief with references to the notion that protection of a particular work can be "thin" or "weak." *E.g.,* GB 2, 5, 42, 70. This concept, too, is not about the threshold question of copyrightability. Whether copyright protection is "thin" or "broad" can be relevant to fair use (which is how *Sega* used it). But it typically arises in the first instance in the infringement analysis: i.e., whether the accused work is sufficiently similar to the original to constitute infringement. Where protection is broad, plaintiffs need to prove only that the accused work is "substantially similar," *Mattel, Inc. v. MGA Entm't, Inc.*, 616 F.3d 904, 913-14 (9th Cir. 2010), whereas if thin, plaintiffs must prove that the accused work is "virtually identical," *Satava v. Lowry*, 323 F.3d 805, 813 (9th Cir. 2003).

Infringement concepts, like fair use, are irrelevant to the threshold question whether Oracle's declaring code is copyrightable. First, such cases do not hold that computer programs are devoid of all copyright protection. Citing *Sega*, Google repeatedly asserts that generally software "receives only *weak* protection," GB 2, 42, 70 (emphasis added; quotation marks omitted); *see* GB 2, 5, 42 ("thin"). As

29

discussed in fair use (at 53-55), that is wrong. But even if that repeated assertion were right, it amounts to a concession that the work here is not entirely *devoid* of copyright protection, as the district court erroneously concluded.

Second, cases considering the robustness of copyright protection are irrelevant where, as here, the defendant admits that it copied. *Feist* teaches that "[t]o establish infringement, two elements must be proven: (1) ownership of a valid copyright, and (2) copying of constituent elements of the work that are original." 499 U.S. at 361. Focusing on the second prong, where the "copying of constituent elements" is admitted, the *only* question is whether the elements "are original"— that is, contain protected expression. *Norse v. Henry Holt & Co.*, 991 F.2d 563, 565-66 (9th Cir. 1993) (quotation marks omitted). "[A] substantial similarity analysis may be useful in a copyright case when the alleged infringer denies that he in fact copied the plaintiff's work…. But … *the substantial similarity analysis is inapposite* to the copying issue" where defendants "admit that they in fact copied phrases from" the original work. *Id.* at 566.

Because Google admits it copied, the only question is therefore whether what Google copied was sufficiently original to be protected.

D. Google's Merger And Interoperability Arguments Have No Bearing On The Copyrightability Of Oracle's Work.

Google's merger and interoperability arguments are irrelevant to the question whether Oracle's packages are copyrightable. Both arguments rely on the same premise: even though *Oracle* had limitless choices when it *authored* the declaring code and structured the packages—and the code and structure and organization of the packages were, therefore, creative and protectable when written—the copyright protection evaporated because of something that happened later. What happened later was that the packages became so popular that Google wanted to copy them to tap into the Java fan base. GB 48-51.

If this formulation of Google's argument was not immediately evident, it is because Google twists the meaning of ordinary words to obfuscate their true import. Here are some translations:

"Industry standard": Google calls the declaring code "a *de facto* industry standard," GB 10—and then subtly drops the hedge, calling it "an industry standard," GB 5, 55. But Google does not mean that there is some standard-setting organization that sets out voluntary disclosure standards (there isn't) or that Oracle promised to let Google or any

31

other commercial enterprise use its work without a license (it didn't). Google just means that the software packages became wildly popular.

"Necessary": When Google says that everything it copied was "necessary for compatibility," GB 33, 46-48, it does not actually mean "necessary." The most Google can assert is that "[t]hree of the [37] accused packages"—more precisely, 750 methods in 61 classes in those three packages—"were 'fundamental to … the Java language.'" GB 19 (quoting A140-41); *see* A20,946-49, 22,385-86, 20,847-49 (discussing A5859-60). But embedded in that assertion is the concession that it was not necessary to copy a thing from the other 34 packages—specifically, 6088 methods in 616 classes and interfaces across all 37 packages. A1065; *see* A22,464.

"Interoperable" and "Compatible": When Google says that it copied the Java packages to ensure "interoperability with existing programs written in" Java, GB 49, and that any use of the Java packages "was dictated by … external factors such as compatibility requirements," it does not mean that it developed a product that was in fact "interoperable" or "compatible." Google admits that Android is *not* interoperable with Java, OB 65-66, in that "you won't be able to write an Android app while also using [Java]-specific classes," A2102. Rather, Google means that it wanted to harness the popularity of the declaring code and organization of certain Java packages so that Android would have a pre-existing community of followers who were accustomed to Java. OB 17-18.

"Imperfect interoperability": When Google invokes the district court's assertion that Android achieved "a degree of interoperability" or "imperfect interoperability," A167, GB 26, it does not mean that there is any significant degree to which Android apps run on Java and vice versa. A21,503-04, 22,386-87, 22,463. There isn't. Instead, Google means that it took the packages of code that were most valuable to it—because programmers would expect to find them in a

smartphone development platform, GB 18-19; McNealy Br. 25—and left behind the packages that were less useful.

With this Google-English Dictionary, this Court can begin to understand Google's interoperability and merger arguments. Both arguments build upon the district court's observation that *once Google decided to tap into the Java community by using commands that were familiar*, "there is only one way to declare a given method functionality" and "everyone using that function must write that specific line of code in the same way." GB 15 (quoting A136). Both arguments rely on Google's idiosyncratic definition of what is "necessary": "Google was just trying to make Android more attractive to programmers who know the Java API conventions," GB 55, and *once Google decided to do that*, "[t]here is no choice in how to express" a particular command, GB 15 (quoting A139). But there is no dispute that Google always had the option of writing its own platform in the Java language without using the bulk of the declaring code that Oracle made so popular. Which is why Google does not deny that both arguments are no different from Ann Droid saying, "My copying was 'necessary' to tap into the *Harry Potter* fan base." OB 1, 63.

As our opening brief explained, Google's notion that copyright
protection evaporates with popularity has never been the law.
OB 52-53, 63-65. The Copyright Act declares that a work acquires
copyright protection at the moment it is "fixed in any tangible medium
of expression," 17 U.S.C. § 102(a), and that "[c]opyright in a work …
subsists from its creation and[] … endures for [the copyright] term," *id.*
§ 302(a); *see* Oman Br. 24. As the Ninth Circuit explains, "the design of
the [Copyright] Act … was clearly to protect all works of authorship
from the moment of their fixation in any tangible medium of
expression." *Apple Computer, Inc. v. Formula Int'l, Inc.*, 725 F.2d 521,
524 (1984) (emphasis added) (quoting CONTU Report at 21).[4]

This rule yields two conclusions. First, once the author creates
the work and fixes it in a tangible medium, a defendant's subsequent

---

[4] It is telling that one of Google's main arguments for why this
should be the law for copyright is that a *trademark* may become a
generic term that others may use if consumer perception changes over
time. GB 50. But as the statutory quotes in the text demonstrate,
unlike trademarks, copyright protection does not turn on consumer
perception. "[I]t is to be expected that … expression in a highly
successful copyrighted work will become part of the language. That
does not mean they lose all protection in the manner of a trade name
that has become generic." *Warner Bros. Inc. v. Am. Broad. Cos.*, 720
F.2d 231, 242 (2d Cir. 1983).

desire "to achieve total compatibility … is a commercial and competitive objective which does not enter into the … question of whether particular ideas and expressions have merged." *Apple Computer*, 714 F.2d at 1253 (quoted by OB 64-65 but never addressed by Google). The cases supporting this conclusion are legion. *See, e.g., Atari*, 975 F.2d at 840 ("External factors did not dictate the design of [plaintiff's] 10NES program"); *Dun & Bradstreet Software Servs., Inc. v. Grace Consulting, Inc.*, 307 F.3d 197, 215 (3d Cir. 2002) (expert's "testimony" about "interoperability … wholly misplaced" because expert "focused on externality from the viewpoint of [defendant's] program, not [plaintiff's]"); *Mitel*, 124 F.3d at 1375 (chastising district court for considering "whether external factors … justified [*defendant's*] *copying*" because inquiry "should have remained upon the external factors that dictated [*plaintiff's*] *selection*" of programming); *Eng'g Dynamics, Inc. v. Structural Software, Inc.*, 26 F.3d 1335, 1347 & n.12 (1994) (rejecting that defendant's desire to achieve "compatibility" is relevant to copyright), *amended on other grounds*, 46 F.3d 408 (5th Cir. 1995); *see also* Oman Br. 18-23 (collecting cases).

Second, Google's "industry standard" argument fails, too. GB 50.

It also ignores a massive body of caselaw—including from the Ninth

Circuit—rejecting the argument that a work's overwhelming success is

a basis for denying copyrightability. *See Practice Mgmt. Info. Corp. v.*

Am. Med. Ass'n, 121 F.3d 516, 517, 520 n.8 (9th Cir. 1997) ("*PMI*") (as

amended) (physician coding system copyrightable even when

government mandated its use, making it the "industry standard"); *CCC*

Info. Servs., Inc. v. Maclean Hunter Market Reports, Inc., 44 F.3d 61, 73

(2d Cir. 1994) (rejecting that car values in the Red Book "[fell] into the

public domain" and lost copyright protection); *Kepner-Trego, Inc. v.*

Leadership Software, Inc., 12 F.3d 527, 537 (5th Cir. 1994) (rejecting

that "wildly successful" work was no longer copyrightable); *Educ.*

Testing Servs. v. Katzman, 793 F.2d 533, 540 (3d Cir. 1986) ("possible

domination in [a particular] field … cannot excuse copying … and

patently does not affect the validity of [plaintiff's] copyright").

Google proposes an alternative rule: In determining

copyrightability, "it only makes sense to look at the options available to

the alleged infringer (*ex post*)." GB 48. But Google does not even try to

reconcile its proposed rule with the above-quoted language from the

36

Copyright Act or the overwhelming caselaw cited above and in the opening brief (at 52-53, 63-64).[5]

Instead, Google, once again, cites only the fair-use discussions in *Sega* and *Sony*, GB 48-50, and commentators (who, themselves, rely on *Sega*), GB 50-51. But the answer here is the same as above: Those passages were about fair use, not copyrightability—about whether it was fair to use a work despite its copyright protection. So naturally they focused on the *infringer's* use rather than the original author's clean slate. These fair-use cases do nothing to overcome the statute and overwhelming body of law holding that copyrightability is determined at inception.

To the same effect is Google's one squib from CONTU: "[W]hen specific instructions even though previously copyrighted, are the only and essential means of accomplishing a given task, their later use by another will not amount to infringement." GB 48 (alterations omitted) (quoting *Sega*, 977 F.2d at 1524 (quoting CONTU Report at 20)). That

---

[5] Google addresses only *PMI*, which it tries to minimize as applying only to cases of "wholesale copying." GB 49-50 n.145. But Google's copying was wholesale. *Supra* at 7, 10. Plus, *PMI*'s rejection of the very same "industry standard" argument had nothing to do with the amount of copying.

passage does not say, as Google contends, that work that was "copyrighted" loses copyright protection. Rather, it just restates the idea/expression distinction. As the immediately preceding CONTU text confirms: "The 'idea-expression identity' provides that copyrighted language may be copied without infringing when there is but a limited number of ways to express a given idea [i.e., merger]." CONTU Report at 20 (emphasis added). Significantly, CONTU repeatedly emphasizes (including in the passage quoted by *Sega* and Google), *only* the merger doctrine bars copyright protection, and merger rarely applies because "[t]he availability of alternative noninfringing language is the rule rather than the exception" for computer programs. *Id.* at 20 n.106.

This case is the rule, not the exception. At the risk of repetition: Google never disputes (or even addresses) that there were countless ways for Oracle to write the declaring code and to design, structure, and organize the packages to achieve particular functions. *Supra* at 7-8, 14. Thus, merger does not apply. And years later, when Google wrote Android, Google, too, had any number of ways to write declaring code for the same functions, so long as it was prepared to build its own fan base rather than hitching its wagon to Oracle's success. OB 52;

38

Spafford Br. 7-8, 12-19 (ellipse-drawing example); McNealy Br. 18-20 (time-zone example). So Google's interoperability argument fails as well.

E.   Oracle Preserved All Facets Of The District Court's Copyrightability Ruling.

As noted above (at 10-11), Google scarcely addresses the legal conclusions that flow from its concession that it literally copied 7000 lines of original code. Google's excuse is that these arguments are "waived due to Oracle's failure to object to instructions and a verdict form that effectively eliminated that theory from the case." GB 67. Oracle did not waive that theory, and neither Google nor the district court thought it did.

When Google asserts that the instruction and verdict form "eliminated that theory," it tacitly concedes, as it must, that the literal copying arguments were in the case *going into* trial. Before trial, both Google and Oracle identified the "declarations of the API elements in the Android class library source code" as the code "accuse[d] of copyright infringement." A24,632, 24,636. But there was no need to give the issue to the jury, because Google conceded it. While Google quotes one part of the instruction, it leaves out the critical sentence

39

where the court instructed the jury: "[Android] uses the same names

and declarations," i.e., declaring code. A984-85, 22,771-73; *accord*

GB 18; A134. On infringement, the court wanted the jury to decide

whether Android's structure and organization were substantially

similar to Java's, but there was nothing for the jury to decide regarding

verbatim copying of declaring code itself.

Google understood that the literal copying remained very much in

the case when it argued, both at the close of evidence and after the

verdict, that it "[wa]s entitled to judgment as a matter of law that the

names *and declarations* from the 37 API packages that appear in the

Android source code are not copyrightable." A24,644 (emphasis added);

see A974, A86-88. Google has not explained why it needed that ruling if

the theory was already waived.

The district court, too, confirmed that verbatim copying of

declaring code remained in the case. It issued its copyright order *after*

the verdict. The order stated: "This order addresses and resolves …

whether the elements replicated by Google from the Java system were

protectable by copyright in the first place." A132. The "elements

replicated by Google" included "*identical* lines [of code] that specify the

40

names, parameters and functionality of the methods and classes, lines

called 'declarations' or 'headers'"— i.e., declaring code. A136 (emphasis

added). The copyright order then addressed copyrightability of the

declaring code, including the short-phrases analysis. A162-65; *see*

SanDisk Corp. v. Kingston Tech. Co., 695 F.3d 1348, 1355 (Fed. Cir.

2012) (no waiver where parties "understood" specific argument had

broader applicability). Google never explains why the district court

would have gone to the trouble of deciding that issue if it had been

waived.

Accordingly, Google is wrong in arguing that "[a] reversal based

on copyrightability of the 7,000 lines viewed apart from SSO could not

alter the judgment." GB 67. The court's judgment of noninfringement

addressed both theories. And reversing that judgment on either theory

necessarily revives a judgment of infringement whether by jury verdict

(for structure and organization) or by judgment as a matter of law (for

verbatim copying).

F.   Google's Policy Arguments Are Unavailing.

Google and its amici argue that copyright protection for the

declaring code is "anti-competitive" and "over-broad," Rackspace Br. 22,

and will be used "as a weapon to block innovation," Start-ups Br. 12; *see*
GB 45; Computer Scientists Br. 2-4, 21; CCIA Br. 5. But history shows
the opposite. Since the Copyright Act expanded to protect computer
programs, software companies—including Facebook,[6] Instagram,[7] and
even Google[8]—have used copyright to protect their work. Yet the
software industry continues to thrive. BSA Br. 3-4. Still, as Google's
amicus contends, programs that are open and available are more likely
to succeed, Rackspace Br. 8, 17-18, which provides copyright owners a
strong incentive to license reasonably—as Oracle has. OB 13-15;
McNealy Br. 21-23; Microsoft Br. 15-16.

No doubt it is faster, cheaper, easier, and more profitable for later
companies to steal an innovator's work rather than license or create it
themselves. *See* Start-ups Br. 8-10, 13. But our copyright laws have
always resisted such expediency on the philosophy that "the best way to

---

[6] *Compare* Rackspace Br. 8 *with* https://www.facebook.com/
legal/terms ¶ 9.9 ("You will not sell, transfer, or sublicense our code,
APIs, or tools.") (last accessed, June 30, 2013).

[7] *Compare* Computer Scientists Br. 25-27 *with* http://instagram.com/
about/legal/terms/api/ ¶ 3 ("The Instagram APIs may be protected by
copyrights.") (last accessed, June 30, 2013).

[8] *Compare* Rackspace Br. 8 *with* http://code.google.com/apis/youtube/
terms.html (last accessed, June 30, 2013).

advance public welfare" is to "encourage[]" authors to engage in "individual effort by" offering them "personal gain." *Mazer*, 347 U.S. at 219; *accord* U.S. Const. art. I, § 8; 17 U.S.C. § 106. For all their policy expositions, Google and its amici never address, much less refute, that Oracle would never have "invested as heavily in Java" if it knew its "investment of millions of dollars and years of development time would not receive copyright protection." McNealy Br. 1; *see id.* at 26 (no protection "would have deterred Sun from maintaining its decades-long mission to revolutionize computer software development"); *cf.* Microsoft Br. 3-4, 8 (district court's opinion undermines the balance copyright law achieves in incentivizing development and promoting innovation); Oman Br. 2-3, 27.

Of course, Google is free to advocate for changes in copyright law, and is doing so vigorously. *See, e.g.*, Picture Archive Council of America ("PACA") Br. 12-13 (discussing Google's anti-copyright history). But "[n]o matter how persuasive the policy arguments … , this [C]ourt is not the proper forum in which to debate them. Where Congress has the clear power to enact legislation, [this Court's] role is only to interpret and apply that legislation … not rewrite the … laws …." *Roche Prods.*

v. Bolar Pharm. Co., 733 F.2d 858, 865 (Fed. Cir. 1984), *superseded on*

other grounds by 35 U.S.C. § 271(e)(1).

II.   GOOGLE'S COMMERCIALLY MOTIVATED AND ILLICIT VERBATIM COPYING IS NOT FAIR USE

Every day Google uses pilfered Oracle work to achieve ever

greater popularity and marginalize Oracle in a marketplace where

Oracle had positioned itself to be a major participant. Google suggests

(at 68-74) that another trial is necessary to determine whether it can

demonstrate fair use. No such retrial is necessary. Google's defense

fails as a matter of law. *See* OB 68-77. Eager to bring its product to

market quickly, Google copied the declaring code and structure and

organization of 37 of the 166 packages in Oracle's Java SE and then

added its own packages. In so doing, Google not only advanced its own

commercial interests but produced a platform incompatible with Java

and stole the Oracle business customers licensing Java SE in the

mobile-phone market. Fair use does not authorize copying critical

portions of a copyrighted software platform and then incorporating the

copied work into a competing platform.[9]

---

[9] Google leads its fair-use argument with the assertion that "nine presumably reasonable jurors reportedly found that Google had proved

44

Despite its heavy reliance on the fair-use portions of *Sega* and *Sony,* Google has little to say in response to Oracle's fair-use points. *Compare* OB 68-77 *with* GB 68-74. Below, we address what little Google does say.

A. Factor 1: Google's Commercially Motivated Copying Is Neither Transformative Nor Intermediate.

Google does not dispute that its use is commercial—to the tune of "billions of dollars," OB 70 (quoting A21,594)—which "tends to weigh against a finding of fair use," *Campbell v. Acuff-Rose Music, Inc.*, 510 U.S. 569, 585 (1994); *Sony*, 203 F.3d at 606. But Google asserts that it satisfies the first factor ("the purpose and character of the use") because its use of Oracle's work is "transformative" and merely aimed at achieving "interoperability." GB 68-70. Both arguments are incorrect.

Android is not transformative. A work is "transformative" when it "adds something new, with a *further purpose or different character*," thereby "*altering the first* [work] with new expression, meaning, or message." *Campbell*, 510 U.S. at 579 (emphases added).

---

that defense." GB 68. There is nothing in the record about the jury vote. Google tried to cite a blog posting . The blog is not in the record and therefore not in the appendix.

45

To be transformative, the new use must change the very nature of the original, giving it "an *entirely different function.*" *Kelly v. Arriba Soft Corp.*, 336 F.3d 811, 818 (9th Cir. 2003) (emphasis added). As abstract as that sounds, concrete examples demonstrate just how different the new work must be in order to qualify as transformative.

Turning a pop song into a rap song is not transformative, inasmuch as both have the purpose of entertaining a listening audience. *See Campbell*, 510 U.S. at 580-82. Even turning it into a rap *parody*, which obviously has at least some additional purpose of "commenting" on the original, *id.* at 579, is not necessarily transformative, *id.* at 578-82. If an "infringer merely uses [the work] to get attention or *to avoid the drudgery in working up something fresh*, the claim to fairness in borrowing from another's work diminishes accordingly (if it does not vanish)." *Id.* at 580 (emphasis added); *see also Dr. Seuss Enters., L.P. v. Penguin Books USA, Inc.*, 109 F.3d 1394 (9th Cir. 1997) (O.J. Simpson satire utilizing Dr. Seuss's *A Cat in the Hat* is not a fair-use parody).

Similarly, in *Harper & Row* (*see* OB 71, 74), *The Nation* failed the first factor when it verbatim reproduced the juiciest tidbits of President Ford's memoir even though the new work was couched as news, and

46

readers typically read newspapers and memoirs for different purposes.
471 U.S. at 565. Likewise, it is not transformative to turn an image in
a photograph into a sculpture, even though the two media are very
different, *see Rogers v. Koons*, 960 F.2d 301, 310 (2d Cir. 1992), or to
display copyrighted paintings on a television show when used for "the
same decorative purpose" as the originals, *Ringgold v. Black Entm't
Television, Inc.*, 126 F.3d 70, 79 (2d Cir. 1997).

Android is far less transformative than any of the foregoing
illustrations. Oracle's packages allow programmers to use prewritten
functions rather than write them from scratch. The same code in
Android (e.g., new URL().openConnection()) enables programmers to
invoke the same pre-programmed functions in exactly the same way.
Use of declaring code and packages in Android does not serve "an
entirely different function" from Java. *Arriba Soft*, 336 F.3d at 818. All
Google did was use the declaring code "to get attention" from
programmers and "avoid the drudgery in working up something fresh."
Campbell, 510 U.S. at 580.

Google argues that "Android is transformative" because "Google's
implementation … 'accounts for 97 percent of the lines of code in [those]

47

packages.'" GB 69 (quoting A178). But Google does not dispute that a

plagiarist cannot excuse copying by showing how much of his work was

not copied. OB 42 (citing authorities). Moreover, though the

percentage of copying might be relevant to the third factor, it is

irrelevant to the first factor's determination of whether the new work is

transformative. Thus, as discussed (OB 73)—but not addressed by

Google—*The Nation* failed the first factor even though the copying was

less than 1% of the original. *Accord* PACA Br. 17-18.

Also meritless is Google's argument that it incorporated the

packages into "an entirely new smartphone platform." GB 69. Oracle

already had a platform for sophisticated mobile devices. *Infra* at 56-58.

Google did not transform anything by updating the platform for still

more sophisticated mobile devices. Plus, a use is not transformative

where the copyist "has done no more than find a new way to exploit the

creative virtues of the original work." *Blanch v. Koons*, 467 F.3d 244,

252 (2d Cir. 2006); *see Worldwide Church of God v. Phila. Church of*

God, Inc., 227 F.3d 1110, 1118 (9th Cir. 2000).

***Not intermediate copying under* Sega *and* Sony.** Google

argues that *Sega* and *Sony* authorize "[a] defendant's use of copyrighted

48

material to create a new platform that is compatible with existing programs." GB 69. That is just another formulation of the argument, discussed above (at 31-39), that Google was justified in copying because it wanted to (but did not have to) tap into the popularity of the software packages.

Sega and *Sony* do not help Google in fair use any more than they did for copyrightability. Those cases authorized a use that was much more modest than Google's. As Google acknowledges, those cases addressed only "whether defendants could engage in 'intermediate copying' of copyrighted computer programs" to "analyze" how to "achieve compatibility with these programs." GB 33. The copies at issue in *Sega* and *Sony* were not final, commercial products, but intermediate copies used to figure out how the software works. In contrast, Google's copying here was for a final, commercial product: Android.

This Court's opinion in *Atari*, which set the stage for those Ninth Circuit decisions, draws this distinction explicitly. *See Sega*, 977 F.2d at 1514 n.1 (*Sega*'s "analysis and … result" is "consistent" with *Atari*); *accord id.* at 1524 n.7. In *Atari*, this Court applied fair-use principles to

49

"new technological innovations," and distilled the four factors to a simple rule: "fair use reproductions of a computer program must not exceed what is necessary *to understand* the unprotected elements of the work." 975 F.2d at 843 (emphasis added).

Sega drew the same distinction. There, Accolade's "use of the copyrighted material[] was simply to study the functional requirements" needed for compatibility—i.e., "reverse engineer[ing]" to learn the "twenty to twenty-five bytes of data" needed to make any game work on a Sega console. 977 F.2d at 1516, 1522. Accolade "did not include any of Sega's code" to generate the needed data string, and wrote its own games that included "a total of 500,000 to 1,500,000 bytes." *Id.* at 1515-16. Contrary to Google's repeated insinuations that *Sony* and *Sega* condone "use" of copied code (GB 33, 35, 39), the Ninth Circuit emphasized that the result would have been different if Accolade had done what Google (or Atari) did: "Our conclusion does not, *of course*, insulate Accolade from a claim of copyright infringement with respect to its *finished products*." *Id.* at 1528 (emphases added); *accord id.* at 1515. The same was true in *Sony*. *Sony* emphasizes repeatedly that "none of the Sony copyrighted material was copied into, or appeared in,

50

[defendant's] final product," 203 F.3d at 600; *accord id.* at 598-99, 602, 606-07, 608 n.11.

This Court has confirmed that "*Sega* … does not stand for the proposition that any form of copyright infringement is privileged as long as it is done as part of an effort to explore the operation of a product that uses the copyrighted software." *DSC Commc'ns Corp. v. Pulse Commc'ns, Inc.*, 170 F.3d 1354, 1363 (1999). The *DSC* defendant created a telephone-exchange card that had to download and copy the software of the plaintiff's exchange system to function. *Id.* at 1358. In words equally apt here, this Court held that use of the plaintiff's software was not fair because it was not "part of an attempt at reverse engineering" but rather "part of the ordinary operation of those cards" for business. *Id.* at 1363.

Even if *Sega* and *Sony* could be stretched to encompass copying that is neither intermediate nor restrained, Google's entire focus on "interoperability" is misplaced. Here again, Google cannot invoke interoperability as an excuse for copying when it intentionally produced a platform that was *not interoperable* in the relevant sense of the word. *Supra* at 32-33. Google's expert and the district court agreed that

51

interoperability means a program that "runs on both the Android platform and the Java platform." A22,348; *see* A167 (programs that "run" on each platform); Spafford Br. 20. Nevertheless, Google contends Android achieved "a degree of interoperability with existing Java programs" that use only the 37 copied packages. GB 26, 55, 72 (quoting A167). But that is just wrong. Google and its Technical Program Manager for Android Compatibility concede that Android is not "Java compatible" and does not support "existing Java apps." OB 66. Google's expert admitted that, because "[t]he entry points [in the code] are different on the platforms," any Android or Java "program itself would need to be modified … so that it would run on the [the other] platform." A22,386-87. "[Y]ou don't really have compatibility," Oracle's expert explained, A21,503-04, since the Android and Java codes have different bytecode architecture, A22,463, and different file formats, *compare* Trial Exhibit ("TX") 610.2[10] (Java ".jar" files) *with* A24,657-59 (Android ".apk" files).[11]

---

[10] Submitted in native-file form in Supplemental Joint Appendix.

[11] Google's observation (at 52-53) that the *Sony* defendant implemented some, but not all, of Sony's functions is misplaced. The

B.   Factor 2: Oracle's Packages Are Highly Creative.

As to the "nature of the copyrighted work," 17 U.S.C. § 107(2), the

conversation begins and ends with Google's decision not to contest

Oracle's extensive presentation that the work is "original, creative,

intuitive, attractive," and so forth. GB 5 (quoting OB; internal

quotation marks omitted); *see* OB 72; McNealy Br. 10-21. Instead of

challenging that undeniable reality, Google's analysis revolves around

this: "As *Sega* recognized, computer programs are 'essentially

utilitarian' in nature, and, under the Copyright Act, 'if a work is largely

functional, it receives only weak protection.'" GB 70 (quoting *Sega*, 977

F.2d at 1527). *Sega* never said that all computer programs receive

weak protection. It said: "To the extent that there are many possible

ways of accomplishing a given task or fulfilling a particular market

demand, the [developer's] choice of program structure and design may

be *highly creative and idiosyncratic*." 977 F.2d at 1524 (emphasis

added). The passage Google selectively quotes says merely that *certain*

elements of *some* computer programs can be dictated by "logical,

_____

Ninth Circuit did not even bother mentioning the fact, because the
games in *Sony* "run" on each system and were thus interoperable.

structural, and visual display elements," and thus receive only weak

protection. *Id.* at 1524. Google fails to demonstrate any such restraints

on the original Java developers. *See* OB 12-13, 32-33, 39-40, 51-52.

Google does not respond to other Ninth Circuit authority

discussed in our opening brief (at 73) finding software to be sufficiently

creative to warrant broad copyright protection, where, as here, it is

creative, intuitive, and attractive. Nor does Google grapple with *Atari*,

where this Court reasoned that the software was necessarily "creative"

and "original," thus "protectable," because there were "a multitude of

expressions" available. 975 F.2d at 840.

C.   Factor 3: Google Copied The Most Important Parts Of The Packages.

On this factor, Google emphasizes, again, that it copied "a small

fraction of the overall code." GB 71. But the third factor is not just

about the "amount" but also about the "substantiality of the portion

used." *See* PACA Br. 21-23. Google does not respond to *Harper & Row*,

which is dispositive. The copying there, though only 1% of the total,

was unfair because the "quoted excerpts" were the work's "dramatic

focal points" and played a "key role in the infringing work." 471 U.S. at

566; *see* OB 74. Applying this principle, Google does not dispute that it

copied the declaring code of the packages it thought programmers would

want in a smartphone platform. GB 18-19; A21,956-58, 21,152-53,

21,503; McNealy Br. 25 (Google "copied Java's packages that were most

useful for mobile platforms"). Nor does Google dispute that declaring

code is all-important, because that is the only code programmers ever

see or use. OB 20-21, 43-44, 74.

Here, again, Google latches onto the district court's reasoning that

"Google replicated what was necessary to achieve a degree of

interoperability—but no more." GB 71. But this argument depends on

the same flawed logic as to what was "necessary." *Supra* at 32. Since

the undisputed fact is that Google did not need to copy the vast majority

of the declaring code to write programs in the Java language, *id.*, the

copying was not necessary in any relevant sense of the word. It was

merely desirable to achieve a commercial objective. That makes it

unfair.

D.   Factor 4: Android Hurt The Market And Potential Market For Derivative Works Of Java SE.

Android caused significant "harm to the market" and "*potential*

market" "for derivative works" of Java SE. *See Harper & Row*, 471 U.S.

at 568; 17 U.S.C. § 107(4). Google completely ignores the "harm to the

potential market" that would ensue from infringement such as

Android's "becom[ing] widespread," *Harper & Row*, 471 U.S. at 568

(citation omitted), and barely addresses the actual market harm

Android caused.

Perhaps the starkest illustration of harm was Oracle's market for

licensing of Java ME for mobile devices and of its dedicated smartphone

platform. Market harm does not get any more concrete than the

illustration in our opening brief, to which Google never responds:

Amazon licensed Java ME and its platform for its electronic reader,

Kindle. OB 28-29, 77. But, when Amazon selected the platform for its

new e-reader, the Kindle Fire, it abandoned Java in favor of the (free)

Android platform.

Google concedes that, before Android, Oracle "dominated" the

mobile platform market, GB 17, through licensing of a Java SE

derivative, Java ME, *see* OB 15; A20,708. But Google insists that does

not matter because Java ME was for "feature phones" while Android is

for "smartphones." GB 72 (citation omitted). Google does not dispute,

however, that Oracle licensed Java ME to "just about every smart

phone carrier … around the world," A22,237, including smartphones

such as RIM's BlackBerrys, Danger's Sidekicks, and Nokia's Series 60s,

OB 71. To differentiate some smartphones from others (as Google does)

slices the "market" too thin. The "market for derivative works" here is

the market for mobile device platforms, not the market for platforms

that support a specific generation of phones. *See Twin Peaks Prods.,*

Inc. v. Publ'ns Int'l, Ltd., 996 F.2d 1366, 1377 (2d Cir. 1993)

("extraordinar[ily] detail[ed]" reports of TV episodes were "adequate

substitute[s]" and undermined market for the shows).

Google only underscores the profound impact it had on Oracle's

market by quoting the district court's observation (albeit not from any

fair-use analysis) that "Oracle never successfully developed its own

smartphone platform." GB 72 (quoting A135). "When Google made

Android available for free, Oracle was effectively competing with a free

version of its own program." McNealy Br. 7. The business case for

Oracle selling a dedicated smartphone platform (Java FX) evaporated,

A20,490—another classic example of a superseding use. OB 76-77;

A6142 ("Google is both a potentially key adopter of Java FX due to their

strong support of Java on non-Android devices … but they are also a

potentially dangerous competitor.").

Android also harmed the potential market for a Java smartphone device. Sun/Oracle contemplated creating a Java smartphone—separate from the Java FX platform. A20,490-95; *see* GB 21 (citing A20,487). Building a Java smartphone was one reason Oracle purchased Sun. A8248, 20,487-90. Oracle even built a prototype and devised a strategy to charge carriers for the phone rather than advertising (like Google does). A20,490, 20,495-96. Thus, there can be no doubt that smartphones represented a "potential market" for the Java platform. *See* 17 U.S.C. § 107(4); *A&M Records, Inc. v. Napster, Inc.*, 239 F.3d 1004, 1017 (9th Cir. 2001).

Against all this, Google argues that Android is not a derivative of Java. GB 72-73. That is beside the point. Factor four considers harm *to* derivatives (i.e., harm to Oracle's Java SE derivatives), which has nothing to do with whether the infringing work (Android) is itself a derivative. *Harper & Row*, 471 U.S. at 568.

Google also responds with *Sega* (at 73), but there the copying likely helped the market prospects of the original author: Accolade's copying "led to an increase in the number of independently designed video game programs offered *for use with the [Sega] console*." 977 F.2d

58

at 1523 (emphasis added). Accordingly, Sega suffered only "minor economic loss," *id.* at 1524, and the court found "no basis" for concluding that Accolade's copying "significantly affected the market" for Sega's work, *id.* at 1523. Here, by contrast, no Android program can run on any Java platform, and Oracle has no possible upside in the unlicensed copying of its work.

E.   Any Remand Must Be Limited To Fair Use Only.

Google argues in one sentence that if there is a remand on fair use, this Court must also order a new trial on infringement. GB 73-74. Google's argument improperly relies on incorporation-by-reference of briefs it filed below. *Monsanto Co. v. Scruggs*, 459 F.3d 1328, 1335 (Fed. Cir. 2006). Moreover, Google admits that it copied the declaring code and the structure and organization of the packages. It is not entitled to a trial on an issue that it conceded and on which it offers no legitimate defense.

Even if Google had a triable issue, it is not entitled to a retrial of an issue it already tried and lost. The question whether Google's copying of Oracle's code constitutes infringement is "distinct and separable" from whether Google can establish fair use. *Gasoline Prods.*

Co. v. Champlin Refining Co., 283 U.S. 494, 500 (1931). The two issues

are not "interwoven," and it would not create any "confusion and

uncertainty" to submit fair use to the jury without allowing that jury to

decide infringement as well. *Id.*

CROSS APPEAL

STATEMENT OF THE ISSUE

Google admits it copied portions of Oracle's code verbatim into the

Android platform and that the portions of code are copyrightable. Is

Google's copying copyright infringement?

STATEMENT OF FACTS

The Java platform includes eight files that contain security

functions governing access to network files. A21,501-02. Google

decompiled those files and then copied them identically, "in their

entirety," into Android. A989, 1058A-B; *accord* A21,431-32, 21,919.

Google "agrees that the accused lines of code" (i.e., the eight files)—

containing hundreds of lines of code—"came from [Oracle's] copyrighted

material." A989; *accord* A1058A-B, 3552-69, 21,431-32, 21,875-76.

The Java platform also includes nine lines of original source code

from "java.util.arrays.java," called "rangeCheck." A21,426-27.

"rangeCheck" facilitates an important sorting function, frequently called upon during the operation of Java and Android. A2877-937, 20,923, 21,501, 21,427, 21,436.

As the parties agreed and the jury was instructed, A989, Android includes the identical nine lines of code in the files "Timsort.java" and "ComparableTimSort.java." A21,426-27, 5980; *compare* A2877-937 *with* A1819-37. Android mobile devices call the rangeCheck code 2,600 times just while powering on. A21,501. After start-up, many other files continue to call upon rangeCheck. A21,488.

At trial, Google conceded direct copying of both rangeCheck and the security files. Google did not claim "fair use," A42, instead raising as a defense "only" that its copying "[wa]s de minimis." A989.

The jury was instructed: "With respect to … rangeCheck and other similar files [i.e., security files]*, Google agrees that the accused lines of code and comments came from the copyrighted material* but contends that the amounts involved were so negligible as to be *de minimis* and thus should be excused." A989 (emphasis added).

61

As to rangeCheck, the jury rejected Google's de minimis defense and found infringement. A42. The court denied Google's JMOL. A124, 129.

As to the security files, the jury returned a noninfringement verdict. A42. But, because the testimony showed that Google's "us[e of] the copied files … would have been significant," and "[t]here was no testimony to the contrary," the court concluded that "[n]o reasonable jury could find that this copying was *de minimis*" and granted Oracle's JMOL. A1058A-B.

The district court entered final judgment for Oracle as to the security files and the rangeCheck code. Google appeals.

SUMMARY OF ARGUMENT

Google's efforts to hide behind a so-called "de minimis" exception to copyright protection fails. Where defendants concede verbatim copying, the Ninth Circuit does not recognize any de minimis defense. Instead, the amount of the copying is evaluated under fair use. Even assuming a de minimis defense, Oracle's rangeCheck and security files are significant. The judgments of infringement should be affirmed.

ARGUMENT

I.   THERE IS NO "DE MINIMIS" DEFENSE TO COPYRIGHT INFRINGEMENT.

As a threshold matter, the district court's judgment can be affirmed on the ground that the Copyright Act prohibits copying subject only to the limitations of §§ 107-22. Accordingly, "even a small taking may sometimes be actionable." *Norse v. Henry Holt & Co.,* 991 F.2d 563, 566 (9th Cir. 1993).

Google argues that its copying of rangeCheck and the security files is not infringement because the copying was "*de minimis.*" GB 76-79. The district court properly concluded that the copying here was not de minimis. § II. Also, as Oracle urged below, § 102 does not include any de minimis exception. *See* A24,626 (contending "[n]o *de minimis* exception can, *as a matter of law,* defeat Oracle's claim based on this copying"); *accord* A24,602A.[12]

Though Google contends the amount of verbatim copying must be "significant enough to constitute infringement," GB 76 (quotation marks

_____

[12] While Oracle did not argue this throughout the proceedings below, this Court can affirm "on any ground that finds support in the record." *Jaffke v. Dunham*, 352 U.S. 280, 281 (1957); *Rexnord Indus., LLC v. Kappos*, 705 F.3d 1347, 1356 (Fed. Cir. 2013); *Livid Holdings Ltd. v. Salomon Smith Barney, Inc.*, 416 F.3d 940, 950 (9th Cir. 2005).

omitted), the *only* provision contemplating consideration of "the amount" of copying is the third fair-use factor. 17 U.S.C. § 107(3) ("amount and substantiality of the portion" copied).

Accordingly, the Ninth Circuit does not recognize a freestanding de minimis defense. *Norse*, 991 F.2d 566. In *Norse,* defendants published a poet's biography containing verbatim copies of phrases from plaintiff's unpublished letters to the poet. The copied phrases appeared in a single paragraph of the 768-page biography. The Ninth Circuit held that the case could not be dismissed on de minimis grounds: "The question of whether a copying is substantial enough to be actionable may be best resolved through the fair use doctrine, which permits courts to avoid rigid application of the copyright statute when, on occasion, it would stifle the very creativity which the law is designed to foster." *Id.* (quotation marks omitted). *See* 2 Nimmer & Nimmer, *Nimmer on Copyright*, § 8.01[G], at 8-26 (the "*de minimis* ... defense should be limited largely to its role in determining either substantial similarity or fair use").

Google invokes *Newton v. Diamond*, 388 F.3d 1189, 1193 (9th Cir. 2004), involving music sampling. GB 76-77. But, as an earlier decided

panel decision whose holding is directly on point, *Norse* is binding here.

Koerner v. Grigas, 328 F.3d 1039, 1050 (9th Cir. 2003) ("one three-judge

panel of this court [generally] cannot reconsider or overrule the decision

of a prior panel" (citation omitted)). Accordingly, subsequent sampling

cases in the Ninth Circuit have not followed *Newton, see, e.g., Swirsky v.*

Carey, 376 F.3d 841 (9th Cir. 2004) (analyzing sampling claim under

rubric of sufficient original expression), and *Newton* has been rejected

elsewhere, *see, e.g.*, *Bridgeport Music, Inc. v. Dimension Films*, 410 F.3d

792, 802 (6th Cir. 2005) (expressly rejecting "mental gymnastics" of *de*

minimis defense).

As the Ninth Circuit's disregard of *Newton* confirms, *Newton* was

wrongly decided. In *Newton*, the defendant copied a three-note

sequence from a recording without a license to the composition. The

issue was whether the average audience could discern the plaintiff's

"hand as a composer … from [the defendants'] use of the sample." 388

F.3d at 1196. Failing to cite *Norse* and relying instead on out-of-circuit

authority interpreting the predecessor 1909 Copyright Act (not the

operative 1976 Copyright Act), *Newton* held that defendants' use of the

three-note composition was de minimis—seemingly carving out a de

minimis exception from "the general test for substantial similarity." *Id.*

at 1193-95. This was error. "'Substantial similarity' is not an element

of a claim of copyright-infringement. Rather, it is a doctrine that helps

courts adjudicate whether copying … actually occurred …." *Range*

Road Music, Inc. v. East Coast Foods, Inc., 668 F.3d 1148, 1154 (9th

Cir. 2012). It has no relevance to cases where, as here, verbatim

copying is conceded. *Id.*; *Norse*, 991 F.2d at 566; *see Nimmer, supra,*

§ 8.01[G], at 8-26.

In light of the Ninth Circuit's binding authority in *Norse*, Google's

reliance on Second Circuit precedent is misguided. GB 77 (citing

Ringgold, 126 F.3d at 74).

Because Google concedes it copied the rangeCheck code and the

security files, A989, and because Google's "only" defense to

infringement—de minimis—is not a cognizable defense, A989, this

Court should affirm the judgments of infringement.

II.   GOOGLE'S COPYING WAS SIGNIFICANT.

Even if there is a de minimis defense, the jury reasonably found

Google's copying more than de minimis as to rangeCheck and the

district court correctly held that the defense would fail as a matter of

66

law as to the security files. Google concedes that copying is not de minimis if the copied work is "*qualitatively* or quantitatively significant." GB 77 (emphasis added). The unrefuted record establishes that the security files and rangeCheck had substantial significance—both to Oracle and to Google. Thus, the jury was correct to find infringement of rangeCheck, A42, and the district court was correct to hold that no reasonable juror could find anything but infringement of the security files, A1058A-B.

As to rangeCheck, Dr. Mitchell presented unrebutted testimony that it is significant and called upon often by other files. In one experiment testing the importance of rangeCheck, Dr. Mitchell observed that it is called upon at least 2600 times just when an Android device is powering on. A21,501. This is not disputed.

As to the security files, Google contends that the "files [a]re qualitatively [in]significant to the [Java] platform." GB 77. But that is not what the undisputed testimony was: Dr. Mitchell testified that these files are significant because they are "the default implementation for the security functions" that "govern access to a network or other

67

resource" in the Java platform. A21,501-02. Google offered no evidence in response.

Google also argues the security files are not "qualitatively significant to … Android" because "they were used to test the [Android] platform" but not shipped on any Android devices. GB 77. But companies invest significant time and money developing code to test products to ensure that they reach the market without quality issues. A21,502-03. Such testing "is a very important part of the software development." A21,502. Indeed, "by many measures the testing and quality assurance process can be twice as time consuming or twice as expensive as coding originally." *Id.* In short, code used for testing can be as valuable as that used in the end product.

Google suggests that a minimum number of lines of code must be copied before finding infringement. GB 78-79. "[N]o bright line rule exists as to what quantum of similarity is permitted." *Baxter v. MCA, Inc.*, 812 F.2d 421, 425 (9th Cir. 1987); *Brocade Commc'ns Sys., Inc. v. A10 Networks, Inc.,* C 10-3428-PSG, 2013 WL 831528, *8 (N.D. Cal. Jan. 10, 2013) (copying 145 lines of code out of 10 million not de minimis); *CyberMedia, Inc. v. Symantec Corp.,* 19 F. Supp. 2d 1070,

1077 (N.D. Cal. 1998) (copying not de minimis even where the code copied comprised a relatively small percentage of the program).

Because no particular quantum of lines of code needs to be copied for infringement, and because the qualitative value of the copied code is so substantial, Google cannot show that the district court erred.

III.   THE JURY INSTRUCTION ON THE "WORK AS WHOLE" WAS NOT ERRONEOUS OR REVERSIBLE.

Google argues that the jury instruction incorrectly defined "the work as a whole" with regard to rangeCheck and the security files. GB 75-76. As an initial matter, this Court need not even reach this issue if it agrees with either of the arguments presented above. An instruction regarding *quantitative* analysis for proving de minimis copying is inconsequential if there is no such defense. The argument also does not matter if Oracle proved as a matter of law that the copied code was *qualititatively* significant.

In any event, the district court correctly held, on the facts of this case, that the individual code files copied were the work for the purpose of evaluating Google's culpability for direct copying of these files. Google's two-paragraph argument does not show otherwise. The Ninth Circuit holds that "an entire work" can be a subset of a larger work if

that work "can stand totally alone." *Hustler Magazine, Inc. v. Moral*

Majority, Inc., 796 F.2d 1148, 1154-55 (1986) (cited by GB 75 n.193).

"[W]hen a single published unit contains multiple elements 'that are

otherwise recognizable as self-contained works,' the unit is considered a

single work for the limited purpose of registration, while its elements

may be recognized as separate works for other purposes." A33 (quoting

37 C.F.R. § 202.3(b)(4)(i)(A)); *accord Super Future Equities v. Wells*

Fargo Bank, 553 F. Supp. 2d 680, 699-700 (N.D. Texas 2008) (finding

whole work was the website page, not entire website).

Google does not show the trial court erred in determining that the

copied code files were stand-alone works. These files, which are not

within the 166 packages, can and do stand alone as individual units in

source- and object-code form. *See* A2877-964. Indeed, the files are the

tangible medium in which the copied expression is fixed and from which

Google actually copied, and accordingly were admitted into evidence as

individual files, each with its own copyright notice and author

information. *Id.* Under those circumstances, the copying was

quantitatively significant since, as the district court found, Google

copied the eight security files "in their entirety." A1058A-B. And

Google failed to ever rebut any of the evidence of the files' function and

significance. *Supra* at 67-69.

CONCLUSION

On Oracle's appeal, this Court should reverse the district court's

judgment and enter judgment on liability for Oracle. On Google's cross-

appeal, this Court should affirm the district court's judgment.

Dated: July 3, 2013   Respectfully submitted,

                                   By: /s/ E. Joshua Rosenkranz
                                       E. Joshua Rosenkranz
                                       Orrick, Herrington & Sutcliffe LLP
                                       51 West 52nd Street
                                       New York, NY 10019
                                       (212) 506-5000

                                       Attorney for Plaintiff-Appellant

71

CERTIFICATE OF SERVICE

I hereby certify that on July 3, 2013, I caused the foregoing

Response And Reply Brief Of Plaintiff-Appellant to be electronically

filed with the Clerk of the Court using CM/ECF, which will

automatically send email notification of such filing to the following

counsel of record:

Christa M. Anderson   Renny Hwang
Robert A. Van Nest   Google Inc.
Steven Hirsch   1600 Amphitheatre Parkway
Michael Kwun   Mountain View, CA 94053
Dan Jackson
Keker & Van Nest, LLP   Ian C. Ballon
633 Battery Street   Heather Meeker
San Francisco, CA 94111   Greenberg Traurig, LLP
                                1900 University Avenue
Daryl L. Joseffer   5th Floor
Bruce W. Baber   East Palo Alto, CA 94303
King & Spalding LLP
1700 Pennsylvania Ave., N.W.
Washington, D.C. 2006

Counsel for Defendant-Cross Appellant

By: /s/ E. Joshua Rosenkranz
Attorney for Plaintiff-Appellant

CERTIFICATE OF COMPLIANCE
UNDER FEDERAL RULES OF APPELLATE PROCEDURE
32(a)(7) AND FEDERAL CIRCUIT RULE 32

Counsel for Plaintiff-Appellant certifies that the brief contained

herein has a proportionally spaced 14-point typeface, and contains

13,979 words, based on the "Word Count" feature of Word 2007,

including footnotes and endnotes. Pursuant to Federal Rule of

Appellate Procedure 32(a)(7)(B)(iii) and Federal Circuit Rule 32(b), this

word count does not include the words contained in the Certificate of

Interest, Table of Contents, Table of Authorities, Abbreviations, and

Statement of Related Cases.

Dated: July 3, 2013   Respectfully submitted,

                                 By: /s/ E. Joshua Rosenkranz
                                 Attorney for Plaintiff-Appellant