

1 KEKER & VAN NEST LLP  
ROBERT A. VAN NEST - # 84065  
2 rvannest@kvn.com  
CHRISTA M. ANDERSON - # 184325  
3 canderson@kvn.com  
MICHAEL S. KWUN - # 198945  
4 mkwun@kvn.com  
633 Battery Street  
5 San Francisco, CA 94111-1809  
Tel: 415.391.5400  
6 Fax: 415.397.7188

KING & SPALDING LLP  
DONALD F. ZIMMER, JR. - #112279  
fzimmer@kslaw.com  
CHERYL A. SABNIS - #224323  
csabnis@kslaw.com  
101 Second Street, Suite 2300  
San Francisco, CA 94105  
Tel: 415.318.1200  
Fax: 415.318.1300

7 KING & SPALDING LLP  
SCOTT T. WEINGAERTNER  
8 (*Pro Hac Vice*)  
sweingaertner@kslaw.com  
9 ROBERT F. PERRY  
rperry@kslaw.com  
10 BRUCE W. BABER (*Pro Hac Vice*)  
1185 Avenue of the Americas  
11 New York, NY 10036  
Tel: 212.556.2100  
12 Fax: 212.556.2222

IAN C. BALLON - #141819  
ballon@gtlaw.com  
HEATHER MEEKER - #172148  
meeckerh@gtlaw.com  
GREENBERG TRAURIG, LLP  
1900 University Avenue  
East Palo Alto, CA 94303  
Tel: 650.328.8500  
Fax: 650.328.8508

13 Attorneys for Defendant  
14 GOOGLE INC.

15 UNITED STATES DISTRICT COURT  
16 NORTHERN DISTRICT OF CALIFORNIA  
17 SAN FRANCISCO DIVISION

18 ORACLE AMERICA, INC.,  
19 Plaintiff,  
20 v.  
21 GOOGLE INC.,  
22 Defendant.

Case No. 3:10-cv-03651 WHA

**GOOGLE'S APRIL 22, 2012 COPYRIGHT LIABILITY TRIAL BRIEF**

Dept.: Courtroom 8, 19<sup>th</sup> Floor  
Judge: Hon. William Alsup

1 **I. INTRODUCTION**

2 Google responds to the questions the Court has asked regarding copyright issues and the  
3 37 API packages.

4 **II. ARGUMENT**

5 **A. Structure, sequence and organization can be patented.**

6 Structure, sequence and organization can be patented, assuming it is part of a “process,  
7 machine, manufacture, or composition of matter . . .” 35 U.S.C. § 101. As with any patent  
8 claim, the scope of any such claim would be limited to embodiments that fall within the claim  
9 language. The patent claims would also need to satisfy the other requirements of the Patent Act.

10 Patentable subject matter is defined by Section 101 of the Patent Act:

11 Whoever invents or discovers any new and useful process, machine, manufacture,  
12 or composition of matter, or any new and useful improvement thereof, may obtain  
a patent therefor, subject to the conditions and requirements of this title.

13 35 U.S.C. § 101; *see also Bilski v. Kappos*, 130 S. Ct. 3218, 3225 (2010) (discussing 35 U.S.C.  
14 § 101). The structure, sequence and organization of a set of APIs could be part of a patentable  
15 machine. *See In re Alappat*, 33 F.3d 1526, 1545 (Fed. Cir. 1994) (en banc) (“such programming  
16 creates a new machine, because a general purpose computer in effect becomes a special purpose  
17 computer once it is programmed to perform particular functions pursuant to instructions from  
18 program software”).

19 In explaining Section 102(b) of the Copyright Act, the Federal Circuit has noted that  
20 “patent and copyright laws protect distinct aspects of a computer program.” *Atari Games Corp.*  
21 *v. Nintendo of Am. Inc.*, 975 F.2d 832, 839 (Fed. Cir. 1992) (citing *Baker v. Selden*, 101 U.S. 99,  
22 103 (1879)). “Title 35 [i.e., the Patent Act] protects the process or method performed by a  
23 computer program; title 17 [i.e., the Copyright Act] protects the expression of that process or  
24 method.” *Id.* As the Supreme Court has explained:

25 There is no doubt that a work on the subject of book-keeping, though only  
26 explanatory of well-known systems, may be the subject of a copyright; but, then, it  
27 is claimed only as a book. Such a book may be explanatory either of old systems,  
28 or of an entirely new system; and, considered as a book, as the work of an author,  
conveying information on the subject of book-keeping, and containing detailed  
explanations of the art, it may be a very valuable acquisition to the practical  
knowledge of the community. ***But there is a clear distinction between the book,***

1 *as such, and the art which it is intended to illustrate.* The mere statement of the  
 2 proposition is so evident, that it requires hardly any argument to support it. *The*  
 3 *same distinction may be predicated of every other art* as well as that of book-  
 4 keeping. A treatise on the composition and use of medicines, be they old or new;  
 5 on the construction and use of ploughs, or watches, or churns; or on the mixture  
 6 and application of colors for painting or dyeing; or on the mode of drawing lines to  
 7 produce the effect of perspective, — would be the subject of copyright; but *no one*  
 8 *would contend that the copyright of the treatise would give the exclusive right to*  
 9 *the art or manufacture described therein.* The copyright of the book, if not  
 10 pirated from other works, would be valid without regard to the novelty, or want of  
 11 novelty, of its subject-matter. The novelty of the art or thing described or  
 12 explained has nothing to do with the validity of the copyright. *To give to the*  
 13 *author of the book an exclusive property in the art described therein, when no*  
 14 *examination of its novelty has ever been officially made, would be a surprise and*  
 15 *a fraud upon the public. That is the province of letters-patent, not of copyright.*  
 16 The claim to an invention or discovery of an art or manufacture must be subjected  
 17 to the examination of the Patent Office before an exclusive right therein can be  
 18 obtained; and it can only be secured by a patent from the government.

19 *Baker*, 101 U.S. at 101-02 (emphases added).<sup>1</sup>

20 The core of Oracle’s copyright claims—everything except its documentation claims and  
 21 its claims about portions of 12 files—is barred by *Atari* and *Baker*. Oracle claims that “the  
 22 selection and arrangement” of the elements of the 37 APIs is copyrightable. *See* Oracle 4/5/12  
 23 Br. [Dkt. 859] at 10. But the selection and arrangement of APIs elements cannot, standing alone,  
 24 support a copyright infringement verdict. If, for the 37 APIs at issue, Google had substituted  
 25 APIs that had *exactly* the same structure, selection and organization as the Oracle APIs, but that  
 26 *did* different things, the resulting work as a whole would not be substantially similar. For  
 27 example, if every method always returned the same result, regardless of what inputs were  
 28 provided (e.g., a zero for methods with numerical results, an “a” for those that return strings,  
 “true” for those that return true or false, and so on), the resulting APIs would not be substantially  
 similar, notwithstanding having *precisely* the same structure, selection and organization as  
 Oracle’s APIs. The true premise of Oracle’s claim is that the Google APIs *do* the same thing that

<sup>1</sup> In *Mazer v. Stein*, 347 U.S. 201 (1954), the Supreme Court stated that “[n]either the Copyright Statute nor any other says that because a thing is patentable it may not be copyrighted. We should not so hold.” *Id.* at 217. In the footnote immediately following that statement, however, the Court cited an article about the overlap between copyrights and *design* patents. *Id.* at 217 n.38 (citing Richard W. Pogue, *Borderland—Where Copyright and Design Patent Meet*, 6 Copyright L. Symp. 3 (1955)). The Court held that a statuette can qualify both as “art for the copyright” and an “invention of original and ornamental design for design patents.” *Mazer*, 347 U.S. at 218. Section 102(b) of the Copyright Act does not preclude “designs” from copyright protection, but *does* exclude procedures, processes, systems and methods of operation. *See* 17 U.S.C. § 102(b).

1 the Oracle APIs *do*—that the Google APIs perform the same *functions* as the Oracle APIs. In  
2 short, in contradiction to Supreme Court precedent, Oracle is claiming that it owns the “the  
3 exclusive right to the art or manufacture described” in its specifications. *See Baker*, 101 U.S. at  
4 102.

5 **B. The Copyright Office did not consider whether the structure, sequence and**  
6 **organization of the 37 APIs is copyrightable subject matter.**

7 Counsel for Oracle has admitted, “I don’t think there’s any—I don’t think there’s any  
8 imprimatur on the theory of copyrightability. I think there is an imprimatur on the registerability  
9 of the underlying material.” RT 884:10-13. Here, the “underlying material” was the J2SE 5.0  
10 platform *as a whole*. Nothing in the registration application offered any hint to the Copyright  
11 Office that copyright protection was sought specifically for the 166 API packages that were a part  
12 of that platform, let alone for the 37 API packages specifically at issue in this case—and certainly  
13 no notice was given that Sun sought protection for the structure, sequence and organization of  
14 those 37 API packages. *See TX 475, 476; see also RT 884:3-7* (registration application did not  
15 give the Copyright Office a “heads up” that Sun was “seeking copyright protection on the  
16 structure, sequence, and organization” of the 37 API packages).

17 Any contrary assertion would border on the absurd. If, for example, an author seeks to  
18 register copyright in a biography of John F. Kennedy, the Copyright Office does not examine the  
19 book to determine whether the book includes quotations of President Kennedy’s speeches over  
20 which the biographer cannot claim copyright. The Copyright Office does not note that the  
21 copyright does not extend to the individual words in the book. The Copyright Office does not  
22 add a disclaimer that the registration does not preclude others from “copying” the idea of writing  
23 a biography of our thirty-fifth president. The Copyright Office does not consider whether the  
24 structure, sequence and organization of the biography is copyrightable. Instead, the registration  
25 for that biography means only that the Copyright Office has concluded that the biography *as a*  
26 *whole* contains copyrightable subject matter. *See 3-12 Nimmer § 12.11[B][3]* (other than  
27 examining the work *as a whole* to determine copyrightability, “unlike a patent claim, a claim to  
28 copyright is not examined for basic validity before a certificate is issued”).

1           **C.     The plaintiff must prove that an alleged derivative work infringes the**  
2           **plaintiff’s work.**

3           Oracle concedes that for its derivative work claim, it must prove the same elements as it  
4 must for any infringement claim. *See* RT 885:17-21, 886:8-9 (counsel for Oracle); *id.* at 886:13-  
5 15 (counsel for Google: “the Ninth Circuit requires that a derivative work, to be an infringement,  
6 has to include protectable elements, copyrightable elements of the first work”); *id.* 886:20-21  
7 (counsel for Oracle: “I don’t think we disagree with that, Your Honor.”); *see also Mirage*  
8 *Editions v. Albuquerque A.R.T. Co.*, 856 F.2d 1341, 1343 (9th Cir. 1988) (“a work will be  
9 considered a derivative work only if it would be considered an infringing work if the material  
10 which it has derived from a preexisting work had been taken without the consent of a copyright  
11 proprietor of such preexisting work”) (quotation marks and citation omitted). Applying these  
12 concepts in the context of computer video games, the Ninth Circuit has held that “[a] derivative  
13 work must *incorporate a protected work* in some concrete or permanent form.” *Lewis Galoob*  
14 *Toys, Inc. v. Nintendo of America, Inc.*, 964 F.2d 965, 969 (9th Cir. 1992) (emphasis added).

15           **D.     There are no subsidiary fact questions for the jury.**

16           The Court must resolve any factual issues relevant to copyrightability; there are no  
17 subsidiary fact issues for the jury. *Lotus Dev. Corp. v. Borland Int’l, Inc.*, 788 F. Supp. 78, 96 (D.  
18 Mass. 1992) (“issues of copyrightability, *including any fact questions bearing upon them*, must be  
19 determined by the court, not the jury.”) (emphasis added). As Judge Easterbrook has explained,  
20 “[a] jury has nothing to do with” the copyrightability determination. *Pivot Point, Int’l, Inc. v.*  
21 *Charlene Prods., Inc.*, 932 F. Supp. 220, 225 (N.D. Ill. 1996).

22           As the Supreme Court has noted in the patent context, “functional considerations” play a  
23 role in deciding whether a judge or jury should make various determinations. *See Markman v.*  
24 *Westview Instrs.*, 517 U.S. 370, 388 (1996). Thus, in the patent context, the Supreme Court held  
25 that “there is sufficient reason to treat construction of terms of art like many other responsibilities  
26 that we cede to a judge in the normal course of trial, notwithstanding its evidentiary  
27 underpinnings.” *Id.* at 390. In the context of a copyrightability determination, the *Lotus* court  
28 considered these issues at length, and correctly concluded any subsidiary fact questions should be

1 decided by the court, not the jury. *See Lotus*, 788 F. Supp. at 95-96.<sup>2</sup>

2 **E. What are the relevant “works” for copyright purposes?**

3 **1. Oracle’s “work” (i.e., the work allegedly infringed) is the work it**  
 4 **registered—the Java platform as a whole.**

5 In any copyright case, the work of the plaintiff that is at issue is – and can only be – the  
 6 specific “work” that is the subject of any copyright registration pleaded and relied on by the  
 7 plaintiff. That work, as a whole, is the work to which the defendant’s accused work, as a whole,  
 8 must be compared for purposes of determining: (1) substantial similarity (or virtual identity) and  
 9 therefore infringement; (2) whether the *de minimis* doctrine applies; and (3) defendant’s fair use  
 10 defense under 17 U.S.C. § 107. This conclusion is the only one consistent with the overall  
 11 Copyright Act read in its entirety and is compelled by the unambiguous statutory language of the  
 12 fair use section, 17 U.S.C. § 107. Any other result would defeat the purpose of the statutory  
 13 requirement under 17 U.S.C. § 411 that a plaintiff’s claim of copyright in the asserted works be  
 14 registered as a prerequisite to filing suit,<sup>3</sup> would improperly allow a plaintiff in a copyright case  
 15 to assert claims of infringement for “works” in which no claim of copyright has been registered,  
 16 and would turn a carefully-constructed statutory framework into a guessing game, with the  
 17 plaintiff free to define and re-define its “work” at its whim. *See* RT 922:2-5 (“It seems strange  
 18 that I would have to go through an entire trial and only then would it—would the scales fall from  
 19 my eyes and I would see clearly what the work as a whole is.”).

20 The Copyright Act pervasively refers to the concept of a “work” as the “thing” that is  
 21 subject to copyright protection. Section 408 of the Act provides for the registration, by the owner  
 22 of copyright in a work, of the owner’s “copyright claim” in the work. *See* 17 U.S.C. § 408.

23 While such registration is not “a condition for copyright protection,” *see id.*, no infringement

24 <sup>2</sup> Although subsidiary fact questions about *what* is copyrightable are for the Court, the Ninth  
 25 Circuit has suggested that the Court may define what is copyrightable by category rather than by  
 26 specific item. *See infra* Part II.E.3.a.

27 <sup>3</sup> Section 411’s requirement that the work be registered prior to suit applies only to “United  
 28 States” works. *See* 17 U.S.C. § 411. This exception reflects the fact that the Berne Convention,  
 to which the United States is a signatory, prohibits the United States from imposing formalities  
 such as registration as a pre-condition to suit on works originating in non-United States Berne  
 Convention countries. There is no dispute that Oracle’s work is a United States work, to which  
 section 411 applies.



1 action may be instituted until registration of the owner’s copyright claim has been made in  
2 accordance with the Act. *Id.* § 411(a). Registration, in turn, requires compliance with the Act’s  
3 deposit requirement, set forth in section 411(b). Section 411(b) requires that two complete copies  
4 of the best edition of a published work (or other identifying or other material acceptable to the  
5 Copyright Office) be submitted to the Copyright Office, and section 411(a) requires that those  
6 copies be submitted together with the application for registration.

7 Because of the registration requirement, only “copyrighted works”—i.e., works in which  
8 the owner’s copyright claim has been registered—can be the subject of a claim of infringement.  
9 The Act uses the phrase “copyrighted works” throughout section 106—which defines the  
10 exclusive rights of the owner of a “copyrighted work”—and in other sections, including section  
11 107, which provides that the fair use of a copyrighted work is not an infringement. 17 U.S.C.  
12 §§ 106, 107.

13 In the three copyright registrations identified in Oracle’s Amended Complaint, Sun  
14 Microsystems registered with the Copyright Office two separate “works”:

- 15 1. a work entitled “Java Standard Edition 1.4,” *see* Am. Complaint, Ex. H [Dkt. 36-  
16 1];<sup>4</sup> and
- 17 2. a work entitled “Java 2 Standard Edition, Version 5.0,” *see id.*<sup>5</sup>

18 In accordance with normal Copyright Office practice, the registration forms do not provide a  
19 description or more specific identification of the works. As part of indicating the “nature of  
20 authorship” for Sun and the other authors whose materials were included in the works and the  
21 material that was added to prior versions of the works, Sun stated that the works included both  
22 “computer code” and “accompanying documentation and manuals.” *See* TX 464, 475, 476. The  
23 registrations do not suggest that the “work” being registered was just the APIs (however defined),  
24

25 <sup>4</sup> The registration for this work, No. TX 6-196-514, states that the work is also known as “J2SE  
26 1.4,” “Java 2 Platform, Standard Edition, v 1.4,” “Java 2 Standard Edition Software Development  
27 Kit 1.4” and “SDK 1.4.”

28 <sup>5</sup> The original and supplementary registrations for this work, Nos. TX 6-066-538 and TX 6-143-  
306, state that the work is also known as “J2SE 5.0,” “Java 2 Platform, Standard Edition, Version  
5.0,” “Java 2 Standard Edition 5.0 Development Kit,” “Java 2 Platform Standard Edition 5.0  
Development Kit,” “J2SE Development Kit” and “JDK 5.0.”

1 just the Java class libraries (the source code implementing those APIs, separate and apart from the  
2 source code implementing the rest of the relevant Java platform), just the documentation, or just  
3 the “selection, arrangement and structure” or “selection, structure and organization” of any parts  
4 of the entire work or any portion thereof.

5 It is part of the plaintiff’s burden in a copyright t case to both plead and prove that it owns  
6 the copyright rights in a work that is the subject of a copyright registration. *See Miller v.*  
7 *Facebook, Inc.*, No. C 10-00264 WHA, 2010 U.S. Dist. LEXIS 31534 at \*6-\*7 (N.D. Cal. 2010);  
8 *see also Gee v. CBS, Inc.*, 471 F. Supp. 600, 634 (E.D. Pa. 1979) (“To be sufficient under Rule 8  
9 a claim of infringement must state, inter alia, which specific original work is the subject of the  
10 copyright claim, that plaintiff owns the copyright, that the work in question has been registered in  
11 compliance with the statute and by what acts and during what time defendant has infringed the  
12 copyright.”). In this case, Oracle has argued that it met that pleading burden by pleading “the  
13 pertinent copyright registrations”; in Oracle’s words, “[i]dentification of the copyright  
14 registrations issued is sufficient to establish ownership of the protected materials.” Oracle Opp.  
15 To Google Mot. To Dismiss [Dkt. 40] at 3. In the parties’ Joint Case Management Statement  
16 filed on November 11, 2010 [Dkt. 53], Oracle confirmed that its complaint alleged that  
17 components of Android infringe seven Oracle patents and “*two* of Oracle America’s Java-related  
18 copyrights.” Dkt. 53 at 2 (emphasis added).

19 Once a copyright plaintiff has identified the work and registration on which it bases its  
20 complaint, it is that work, as submitted with the registration, that defines the scope of the  
21 plaintiff’s claim. Courts have, for example, “assumed” that a plaintiff’s claim of infringement  
22 “may be maintained only to vindicate infringement of its work deposited with the registration.”  
23 *E. Mishan & Sons, Inc. v. Marycana, Inc.*, 662 F.Supp. 1399, 1346 (S.D.N.Y. 1987); *see also*  
24 *Tradescape.com v. Shivaram*, 77 F. Supp.2d 408, 414 (S.D.N.Y. 1999) (“The burden of course is  
25 on the plaintiff to show that allegedly infringed code is covered by a valid registration.”). Courts  
26 have also recognized that one of the purposes of the deposit requirement is to provide “sufficient  
27 material to identify the work in which the registrant claims a copyright.” *Nicholls v. Tufenkian*  
28 *Import/Export Ventures*, 367 F. Supp. 2d 514, 520 (S.D.N.Y. 2005) (emphasis added). As the



1 First Circuit has recognized, “a key purpose of the Section 408(b) deposit requirement is to  
2 *prevent confusion about which work the author is attempting to register*” and protect under the  
3 registration. *Data General Corp. v. Grumman Systems Support Corp.*, 36 F.3d 1147, 1162 (1st  
4 Cir. 1994) (emphases added).

5 The Court should not allow Oracle to claim infringement of a subset of its registered  
6 work. In the summary judgment briefing, the sole case Oracle relied upon on this point was *Bean*  
7 *v. McDougal Littell*, 669 F. Supp. 1031 (D. Ariz. 2008). See Oracle Opp. to MSJ [Dkt. 339] at  
8 23-24. In *Bean*, the court noted that “[w]hen a claimant registers a collective work, the copyright  
9 protection can also extend to each constituent part of that work.” 669 F. Supp. at 1034 (emphasis  
10 added). Oracle did not register a collective work, see TX 464, 475, 476,<sup>6</sup> and thus *Bean* is  
11 inapplicable.<sup>7</sup>

12 There should be no confusion – and no issue – about the Oracle works that are the works  
13 at issue. The “works” are the complete Java 2 SE platforms that Oracle chose to register, which  
14 include over 160 API packages, a virtual machine, a compiler, documentation, the Java runtime  
15 environment, the NetBeans development environment, and all of the other elements of the Java  
16 Development Kit. This is, in fact, what Oracle pled in the operative complaint, where it alleged  
17 that “Google’s Android infringes Oracle America’s copyrights *in the Java platform.*” Am.  
18 Compl. [Dkt. 36] ¶ 39 (emphasis added). Oracle has not pled any copyright registrations for any  
19 lesser works, any portions of the works, or any works other than the entire platform.<sup>8</sup>

20 <sup>6</sup> Section 6 of the copyright registration form is titled “DERIVATIVE WORK OR  
21 COMPILATION,” and directs the copyright claimant to “[i]dentify any preexisting work or  
22 works that this work is based on or incorporates.” See TX 464, 475, 476. Sun identified versions  
23 1.4 and 5.0 of the Java 2 SE platform as derivative works of prior versions, but did not identify  
24 them as compilations. A collective work is a time of compilation. See 17 U.S.C. § 101 (“The  
25 term ‘compilation’ includes collective works.”).

26 <sup>7</sup> For the same reason, the reference in 37 C.F.R. § 202.3(b)(4)(i)(A) to “copyrightable elements  
27 that are otherwise recognizable as self-contained works” is inapplicable. The regulation allows a  
28 copyright owner to register as a single work *a collection of artistically related works* that have  
29 been published as a single *unit of publication*. However, if the copyright owner chooses not to  
30 register individual elements of that work, it cannot then assert infringement of the individual  
31 elements as separate works.

<sup>8</sup> Google agrees with Oracle that the issue of which of the Java platform registrations and works  
Oracle chose to rely on in this action is a separate issue from the “work as a whole” issue. See  
RT 920-21. For present purposes, the important fact is that all of Oracle’s relevant registrations  
for its works are for versions of the platform as a whole; none are for APIs, structure, sequence

1                   **2. Google’s “work” (i.e., the allegedly infringing work) is the Android**  
 2                   **platform as a whole.**

3                   It is, in the first instance, the plaintiff’s burden to identify the works of the defendant that  
 4                   it accuses to be infringements. In this case, Oracle made clear in the copyright count of its  
 5                   Amended Complaint that “Google’s Android” is the accused work. Am. Compl. [Dkt. 36] ¶ 39  
 6                   (“Google’s Android infringes Oracle America’s copyrights in the Java platform”; “Google  
 7                   infringes . . . by distributing Android . . . or the code contained within it.”).

8                   Based on Oracle’s pleadings, the accused work is Android—which Oracle defined to  
 9                   include an operating system software “stack” that included Java applications, an application  
 10                  framework, core libraries, a virtual machine and a software development kit. *Id.* ¶ 12. In its  
 11                  Amended Complaint, Oracle identified certain portions of Android that Oracle characterized as  
 12                  “infringing” and certain elements of the Java platform that it characterized as “infringed” (*id.*  
 13                  ¶ 40)—but the pleading in its entirety can only fairly be read to assert a claim that “Android”  
 14                  infringes Oracle’s copyright rights in “the Java platform.”

15                   **3. The jury must compare the works as a whole.**

16                   **a. In determining infringement (including consideration of the *de***  
 17                   ***minimis* doctrine), the jury must compare the works as a whole.**

18                  The Ninth Circuit has held that it is improper, in considering a claim of infringement  
 19                  based on a “selection, coordination and arrangement” of elements, to consider only selected  
 20                  excerpts such as individual pages of the parties’ works. In *Harper House, Inc. v. Thomas Nelson,*  
 21                  *Inc.*, 5 F.3d 536, 1993 WL 346546 (9th Cir. 1993) (“*Harper House II*”) (unpublished), the  
 22                  plaintiff attempted to prove infringement by relying on comparisons of individual pages from the  
 23                  parties’ works rather than the entire works. The court held that such a comparison is  
 24                  “inappropriate and misleadingly prejudicial,” and held that “[o]nly the unique selection,  
 25                  arrangement and coordination of the works *as a whole* may be compared.” 1993 WL 346546 at  
 26                  \*2 (emphasis added). The court specifically noted that there were sections in the defendant’s  
 27                  works that had “no counterparts” in the plaintiff’s work, and held that those elements had to be  
 28                  taken into account in the comparison of the works. *Id.*

and organization, documentation or any other subset, portion or element of the platform.

1 The Ninth Circuit has also held that the court *must* identify for the jury the unprotectable  
2 elements of the plaintiff's work. In *Harper House, Inc. v. Thomas Nelson, Inc.*, 889 F.2d 197 (9th  
3 Cir. 1989) ("*Harper House I*"), the Ninth Circuit held that the jury instructions must "adequately  
4 distinguish between protectable and unprotectable elements." 889 F.2d at 207-08. Relying on  
5 *Harper House I*, the court in *Dream Games of Arizona, Inc. v. PC Onsite*, 561 F.3d 983, 989 (9th  
6 Cir. 2009), reaffirmed that, under *Apple Computer, Inc. v. Microsoft Corp.*, 35 F.3d 1435 (9th Cir.  
7 1994), proper consideration of the works requires that the unprotectable elements are identified.  
8 The district court in *Dream Games* had identified for the jury the specific elements of the  
9 plaintiff's work that were unprotectable. The court therefore affirmed.

10 Thus, while the Ninth Circuit has made plain – and the parties have agreed – that the  
11 issues of copyrightability are questions for the court, Ninth Circuit case law appears to allow the  
12 court two ways to discharge that responsibility. The court could make specific findings that  
13 elements X, Y and Z of plaintiff's work are not copyrightable – and then advise the jury that it is  
14 not to consider those elements in its comparison of the works. Alternatively, the court could  
15 instruct the jury in categorical terms regarding the elements of the plaintiff's work that are not  
16 copyrightable, and then allow the jury to apply those tests – and exclude such elements – before  
17 conducting its comparison of the works for infringement purposes. In this case, that would  
18 require the court to instruct the jury on at least: (1) the statutory exclusions from copyright under  
19 section 102(b), i.e., ideas, systems, methods of operation, etc.; (2) functional requirements for  
20 compatibility; (3) the programming equivalents of scenes a faire; (4) the merger doctrine; and (5)  
21 any other applicable uncopyrightability doctrines that have evidentiary support. While such a  
22 course of action would require additional instructions and place an additional burden on the jury,  
23 it would be consistent with the Ninth Circuit's observation in *Harper House I* that the jury "was  
24 not told that blank forms, common property, or utilitarian aspects of useful items are not  
25 protectable." 889 F.2d at 208.

26 The same analysis applies to the *de minimis* doctrine. In deciding whether alleged  
27 copying is *de minimis*, the qualitative and quantitative significance of the taken material must be  
28 measured "in relation to the plaintiff's work *as a whole*." *Newton v. Diamond*, 388 F.3d 1189,

1 1195 (9th Cir. 2004). In this case, that means the jury must consider whether the portions of the  
2 12 files are qualitatively or quantitatively significant when compared to the *whole* of Oracle’s  
3 work, i.e., *the Java platform*, including all of the J2SE APIs and libraries, the Java Virtual  
4 Machine, the compiler and the SDK.

5 Because the plaintiff’s work for purposes of its claim is, as a matter of law, the work that  
6 is the subject of its registration,<sup>9</sup> it would be error for the court to leave it to the jury to decide  
7 what “works” are at issue. The Court must identify the works as a whole to the jury. For the  
8 plaintiff, the “work” is the work that is the subject of the registration—the Java platform. For the  
9 defendant, it is the accused work – Android. There is no factual or other standard the jury could  
10 apply to determine that some other “works” are at issue or are the “works as a whole” that must  
11 be compared. There is also no burden of proof on identifying the “entire work”; the only burden  
12 is on the plaintiff to identify the work and registration on which it relies.

13 Because of these general principles, it is improper and would be error to allow the jury to  
14 find infringement based on any comparison other than a comparison of the works in their  
15 entireties.

#### 16 **b. Fair Use**

17 For purposes of fair use analysis, the statute—17 U.S.C. § 107—is clear and  
18 unambiguous. In assessing fair use, one of the factors that must be included in the analysis is “the  
19 amount and substantiality of the portion used *in relation to the copyrighted work as a whole.*” 17  
20 U.S.C. § 107(3) (emphasis added); see also 17 U.S.C. §§ 107(2) (referring to the “nature of the  
21 copyrighted work”), 107(4) (referring to the “value of the copyrighted work”). Nothing in section  
22 107 suggests or permits fair use analysis based on a portion of the copyrighted work. The statute  
23

---

24 <sup>9</sup> *Hustler Magazine, Inc. v. Moral Majority, Inc.*, 796 F.2d 1148 (9th Cir. 1986), is not to the  
25 contrary. The “work” at issue in *Hustler* was an issue of a magazine – which is a classic  
26 “collective work” under the Copyright Act. See 17 U.S.C. § 101 (definition of “collective  
27 work”). A “collective work” is, by definition, a work consisting of a number of contributions that  
28 constitute “separate and independent works in themselves.” The “stand alone” test used in  
*Hustler* is appropriate for use only with respect to collective works or, as the court called the  
magazine, a “copyrighted composite work.” 796 F.2d at 1154-55; see 1909 Copyright Act § 3  
(providing for copyright in “composite works”). Oracle did not register the Java platform as a  
collective work.

1 requires that the portions of the plaintiff’s work used by the defendant be analyzed compared to  
 2 the plaintiff’s copyrighted (i.e., registered) work “as a whole.”

3 **F. The Effect of Factual Copying on Substantial Similarity Analysis**

4 Even if the structure, sequence and organization of the 37 API packages and  
 5 “declarations” are held to be copyrightable, substantial similarity of the works as a whole is still  
 6 an issue for the jury.

7 Courts have reiterated many times that “[n]ot all copying . . . is copyright infringement.”  
 8 *See, e.g., Feist Publications, Inc. v. Rural Tel. Serv. Co.*, 499 U.S. 340, 361 (1991); *Jackson v.*  
 9 *Booker*, 2012 U.S. App LEXIS 3024 at \*8 (3rd Cir. Feb. 16, 2012) (“even if actual copying is  
 10 proven, the court must decide, by comparing the allegedly infringing work with the original work,  
 11 whether the copying was unlawful”; unlawful copying requires substantial similarity with respect  
 12 to protected expression); *Intervest Construction, Inc. v. Canterbury Estate Homes, Inc.*, 554 F.3d  
 13 914, 921 (11th Cir. 2008) (substantial similarity must be shown as to “similarity of expression,  
 14 i.e., material susceptible of copyright protection”).

15 The law in the Ninth Circuit is no different:

16 For an unauthorized use of a copyrighted work to be actionable, the use must be  
 17 significant enough to constitute infringement. *See Ringgold v. Black Entm’t*  
 18 *Television, Inc.*, 126 F.3d 70, 74-75 (2d Cir. 1997). This means that ***even where***  
 19 ***the fact of copying is conceded, no legal consequences will follow from that fact***  
 20 ***unless the copying is substantial.*** *See Laureyssens v. Idea Group, Inc.*, 964 F.2d  
 131, 140 (2d Cir. 1992); 4 Melville B. Nimmer & David Nimmer, Nimmer on  
 Copyright § 13.03[A], at 13-30.2.

20 *Newton*, 388 F.3d at 1192-93 (emphasis added).

21 Thus, the issues presented by the presence of isolated identical or very similar elements in  
 22 two works remain the same as in all copyright cases—namely, whether those similarities relate to  
 23 protectable elements and, if they do, whether they are sufficient to support a finding that the two  
 24 works are, in their entirety, “substantially similar” or, in cases of thin copyright protection,  
 25 “virtually identical” as to copyrightable elements.<sup>10</sup>

26 <sup>10</sup> Indeed, one Seventh Circuit decision in 1990 held to the contrary, and stated that “Establishing  
 27 substantial similarity is necessary only when direct evidence of copying is unavailable.” *Illinois*  
 28 *Bell Tel. Co. v. Haines & Co.*, 905 F.2d 1081, 1086 (7th Cir. 1990). That decision, however, was  
 vacated by the Supreme Court following its decision in *Feist*, and, on remand, the Seventh Circuit  
 remanded to the district court with instructions to enter judgment in favor of the defendant. *See*

1 In this case, even if the structure, sequence and organization of the 37 packages is found  
2 to be copyrightable, the jury must still decide whether the protected elements of that structure,  
3 sequence and organization—*excluding* the names, the ideas, the scenes a faire, the programming  
4 conventions, the functional elements necessary for compatibility and any other uncopyrightable  
5 elements—are sufficient to make Android, in its entirety, substantially similar to the Java  
6 platform. That analysis must properly take into account all the elements of the two works,  
7 including those that are different.

8 **G. The source code in the Sun compiled lines in the 37 APIs calls upon other  
9 APIs.**

10 The implementations of the Sun compiled lines in the 37 APIs are not self-contained.  
11 Instead, each of the 37 APIs calls upon methods and classes in other API packages. This is done  
12 for the same reasons third-party programs call on the APIs—to efficiently reuse pre-written code.

13 Indeed, as but one example, *every* single package in the Java 2 SE platform *requires*  
14 *java.lang*, because every single class in every single package in the Java APIs directly or  
15 indirectly inherits the characteristics of the Object class, which is part of the *java.lang* package.  
16 Put another way, the compiled versions of *every single package* in the Java 2 SE platform are  
17 inoperable unless the compiled version of the *java.lang* package is present.

18 As another example, Sun’s implementation of the URL class in the *java.net* API package  
19 needs to keep a “table” of certain information. Having a separate implementation of this  
20 functionality would be inefficient and unnecessary, and so the *java.net* API calls on the  
21 HashTable class in the *java.util* API package to provide it.

22 **H. The implementing code for the Android API packages do not “borrow from  
23 other APIs” in the same pattern as the implementing code for the Oracle API  
24 packages.**

25 Because both the Oracle and Android API implementations implement similar  
26 functionality, both implementations often call upon *similar* APIs in a *similar* pattern to implement  
27 that functionality. For example, Android’s implementation of the URL class in the *java.net* API  
28 package also must track a table of information, and it too calls on the Android’s implementation

---

932 F.2d 610 (7th Cir. 1991).



1 of the HashTable class to provide that functionality.

2           However, in many of the classes in the 37 API packages, the Oracle and Android  
3 implementations of the class call upon different sets of APIs. For example, both the Oracle and  
4 Android implementations of the URL class call upon the java.io API package, but, the Oracle  
5 implementation uses the OutputStream class from the java.io package, while the Android  
6 implementation uses the ObjectOutputStream class from the java.io package.

7           **I. Source code in both the Oracle and Android implementations of the 37 APIs  
8 call upon APIs outside of the 37 APIs.**

9           The cross-referencing is not all within the 37 API packages at issue. For example,  
10 Oracle’s implementations of the 37 API packages at issue call upon at least 29 API packages *that*  
11 *are not even present in Android*. Similarly, Android’s implementations of the 37 APIs at issue  
12 call upon at least 28 packages *that are not even present in Java 2 SE*.

13           Both implementations also call APIs in *other* packages that are present in both platforms,  
14 but not at issue in this case, such as org.w3c.dom, org.xml.sax, javax.xml.transform,  
15 javax.xml.parsers, and java.util.concurrent. For example, Oracle’s implementations of the 37 API  
16 packages often reference APIs in the “sun” namespace that are not present in Android. Similarly,  
17 Android’s implementations reference packages in other namespaces that are not present in  
18 Oracle’s implementation of Java platform, such as the “ibm” namespace.

19           **J. Efficiency and compatibility would be compromised if the interrelationships  
20 of methods and fields were changed by altering their grouping or inheritance.**

21           While it is technically possible to group all methods and fields into arbitrary classes, the  
22 system of APIs is easier for programmers to learn and more efficient for them to use if the APIs  
23 are organized in a predictable and practical fashion. The groupings of the methods and fields  
24 provide a helpful convention for programmers to follow to access and use the functionality of the  
underlying implementations.

25           The testimony of Dr. Mark Reinhold, Oracle’s Chief Java Architect’s, addressed this  
26 question directly, when he said that Sun “could have put all of the NIO—all of the new IO APIs  
27 into one package.” RT 634:11-12 (Reinhold). However, “[h]umans aren’t good at looking at  
28 very long lists of unstructured information,” RT 634:10-11 (Reinhold), and so packages lacking

1 such organization “would be really hard to use from the developer’s, the software developer’s  
2 standpoint.” RT 619:20-21 (Reinhold).

3         Once methods and fields are grouped, changes to that grouping would also lead to another  
4 type of inefficiency: the loss of compatibility. When learning an API, programmers learn how  
5 methods and fields are grouped into classes. They then in turn write software that relies on that  
6 grouping—e.g., that assumes that the “cos” method is in the Math class. As Dr. Josh Bloch  
7 testified, once the developers (and their software) rely on that structure, changes to the name  
8 (which includes information about the grouping into packages and classes) would cause  
9 incompatibility in existing software:

10                 Q.     On any of the occasions while you were at Sun when you worked  
11                 on reimplementing an existing API, did you ever change any of the elements of the  
                  method declaration for an existing method?

12                 A.     No. We couldn’t.

13                 Q.     Why couldn’t you?

14                 A.     Because it wouldn’t work any more, because programs that had  
15                 been written to use that API would no longer work. You would compile them and  
16                 there would be a mismatch. You would call a method name, the method name  
                  better be the same. If you change the name, the program won’t work any more. It  
                  would be an incompatible change.

17 RT 803:9-20 (Bloch). As a result, any alternative implementations of these methods and fields  
18 (and packages) must replicate that specific grouping in order to be compatible with the original  
19 implementation.

20         Inheritance (and the related concept of Interfaces) is another way to organize API  
21 elements in order to improve efficiency, in this case by removing redundancy. *See, e.g.*, RT 590-  
22 92 (Reinhold). Instead of having the equivalent of “dogs have hair and feed their young milk,”  
23 “cats have hair and feed their young milk,” and “humans have hair and feed their young milk,” an  
24 API can define “mammals” as a class, and state that mammals all have hair and feed their young  
25 milk. Dogs, cats and humans could then be defined as subclasses of the mammal class, and  
26 “inherit” the characteristics of having hair and feeding their young milk. The definitions of the  
27 dog, cat, and human subclasses could then focus on defining characteristics unique to each of  
28 those subclasses.

1           The grouping of methods and fields through the use of inheritance is the same as other  
2 groupings or organizations of methods and fields. First, as with other groupings, organization  
3 through inheritance is part of a functional method of operation. Programmers must know about  
4 and use parts of an API's inheritance structure in order to operate the underlying software  
5 libraries. Second, as with other groupings, organization through inheritance must be efficient,  
6 such that it will be easy and practical for programmers to learn and use. Finally, the organization  
7 and structure reflected by inheritance must be followed in all implementations to maintain  
8 compatibility with code written relying upon it.

9  
10 Dated: April 22, 2012

KEKER & VAN NEST LLP

11  
12 By: /s/ Robert A. Van Nest  
13           ROBERT A. VAN NEST

14           Attorneys for Defendant GOOGLE INC.  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28