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

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

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

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

Attorneys for Defendant
GOOGLE INC.

UNITED STATES DISTRICT COURT

NORTHERN DISTRICT OF CALIFORNIA

SAN FRANCISCO DIVISION

ORACLE AMERICA, INC., Plaintiff, v. GOOGLE INC., Defendant.	Case No. 3:10-cv-03651 WHA **GOOGLE'S APRIL 22, 2012 COPYRIGHT LIABILITY TRIAL BRIEF** Dept.: Courtroom 8, 19th Floor Judge: Hon. William Alsup

1

653387.01

I. INTRODUCTION

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

II. ARGUMENT

A. Structure, sequence and organization can be patented.

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

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

> Whoever invents or discovers any new and useful process, machine, manufacture, 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.

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

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

> There is no doubt that a work on the subject of book-keeping, though only explanatory of well-known systems, may be the subject of a copyright; but, then, it is claimed only as a book. Such a book may be explanatory either of old systems, 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

653387.01

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

Baker, 101 U.S. at 101-02 (emphases added).[1]

The core of Oracle's copyright claims—everything except its documentation claims and its claims about portions of 12 files—is barred by *Atari* and *Baker.* Oracle claims that "the selection and arrangement" of the elements of the 37 APIs is copyrightable. *See* Oracle 4/5/12 Br. [Dkt. 859] at 10. But the selection and arrangement of APIs elements cannot, standing alone, support a copyright infringement verdict. If, for the 37 APIs at issue, Google had substituted APIs that had *exactly* the same structure, selection and organization as the Oracle APIs, but that *did* different things, the resulting work as a whole would not be substantially similar. For example, if every method always returned the same result, regardless of what inputs were 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

---

[1] 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).

653387.01

the Oracle APIs *do*—that the Google APIs perform the same *functions* as the Oracle APIs. In

short, in contradiction to Supreme Court precedent, Oracle is claiming that it owns the "the

exclusive right to the art or manufacture described" in its specifications. *See Baker,* 101 U.S. at

102.

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

Counsel for Oracle has admitted, "I don't think there's any—I don't think there's any

imprimatur on the theory of copyrightability. I think there is an imprimatur on the registerability

of the underlying material." RT 884:10-13. Here, the "underlying material" was the J2SE 5.0

platform *as a whole.* Nothing in the registration application offered any hint to the Copyright

Office that copyright protection was sought specifically for the 166 API packages that were a part

of that platform, let alone for the 37 API packages specifically at issue in this case—and certainly

no notice was given that Sun sought protection for the structure, sequence and organization of

those 37 API packages. *See* TX 475, 476; *see also* RT 884:3-7 (registration application did not

give the Copyright Office a "heads up" that Sun was "seeking copyright protection on the

structure, sequence, and organization" of the 37 API packages).

Any contrary assertion would border on the absurd. If, for example, an author seeks to

register copyright in a biography of John F. Kennedy, the Copyright Office does not examine the

book to determine whether the book includes quotations of President Kennedy's speeches over

which the biographer cannot claim copyright. The Copyright Office does not note that the

copyright does not extend to the individual words in the book. The Copyright Office does not

add a disclaimer that the registration does not preclude others from "copying" the idea of writing

a biography of our thirty-fifth president. The Copyright Office does not consider whether the

structure, sequence and organization of the biography is copyrightable. Instead, the registration

for that biography means only that the Copyright Office has concluded that the biography *as a

whole* contains copyrightable subject matter. *See* 3-12 Nimmer § 12.11[B][3] (other than

examining the work *as a whole* to determine copyrightability, "unlike a patent claim, a claim to

copyright is not examined for basic validity before a certificate is issued").

3

653387.01

C. The plaintiff must prove that an alleged derivative work infringes the plaintiff's work.

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

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

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

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

653387.01

decided by the court, not the jury. *See Lotus,* 788 F. Supp. at 95-96.[2]

E. What are the relevant "works" for copyright purposes?

1. Oracle's "work" (i.e., the work allegedly infringed) is the work it registered—the Java platform as a whole.

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

The Copyright Act pervasively refers to the concept of a "work" as the "thing" that is subject to copyright protection. Section 408 of the Act provides for the registration, by the owner of copyright in a work, of the owner's "copyright claim" in the work. *See* 17 U.S.C. § 408. While such registration is not "a condition for copyright protection," *see id.,* no infringement

---

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

[3] Section 411's requirement that the work be registered prior to suit applies only to "United 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.

653387.01

action may be instituted until registration of the owner's copyright claim has been made in

accordance with the Act. *Id.* § 411(a). Registration, in turn, requires compliance with the Act's

deposit requirement, set forth in section 411(b). Section 411(b) requires that two complete copies

of the best edition of a published work (or other identifying or other material acceptable to the

Copyright Office) be submitted to the Copyright Office, and section 411(a) requires that those

copies be submitted together with the application for registration.

Because of the registration requirement, only "copyrighted works"—i.e., works in which

the owner's copyright claim has been registered—can be the subject of a claim of infringement.

The Act uses the phrase "copyrighted works" throughout section 106—which defines the

exclusive rights of the owner of a "copyrighted work"—and in other sections, including section

107, which provides that the fair use of a copyrighted work is not an infringement. 17 U.S.C.

§§ 106, 107.

In the three copyright registrations identified in Oracle's Amended Complaint, Sun

Microsystems registered with the Copyright Office two separate "works":

    1.   a work entitled "Java Standard Edition 1.4," *see* Am. Complaint, Ex. H [Dkt. 36-

        1];[4] and

    2.   a work entitled "Java 2 Standard Edition, Version 5.0," *see id.*[5]

In accordance with normal Copyright Office practice, the registration forms do not provide a

description or more specific identification of the works. As part of indicating the "nature of

authorship" for Sun and the other authors whose materials were included in the works and the

material that was added to prior versions of the works, Sun stated that the works included both

"computer code" and "accompanying documentation and manuals." *See* TX 464, 475, 476. The

registrations do not suggest that the "work" being registered was just the APIs (however defined),

---

[4] The registration for this work, No. TX 6-196-514, states that the work is also known as "J2SE 1.4," "Java 2 Platform, Standard Edition, v 1.4," "Java 2 Standard Edition Software Development Kit 1.4" and "SDK 1.4."

[5] 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."

653387.01

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

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

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

7

653387.01

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

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

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

---

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

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

[8] 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

8

GOOGLE'S APRIL 22, 2012 COPYRIGHT LIABILITY TRIAL BRIEF
Case No. 3:10-CV-03561 WHA

653387.01

2. Google's "work" (i.e., the allegedly infringing work) is the Android platform as a whole.

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

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

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

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

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

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

9

653387.01

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

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

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

10

653387.01

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,[9] 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.

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 [9] *Hustler Magazine, Inc. v. Moral Majority, Inc.,* 796 F.2d 1148 (9th Cir. 1986), is not to the
contrary. The "work" at issue in Hustler was an issue of a magazine – which is a classic
25 "collective work" under the Copyright Act. *See* 17 U.S.C. § 101 (definition of "collective
work"). A "collective work" is, by definition, a work consisting of a number of contributions that
26 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
27 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
28 collective work.

653387.01

requires that the portions of the plaintiff's work used by the defendant be analyzed compared to the plaintiff's copyrighted (i.e., registered) work "as a whole."

F. The Effect of Factual Copying on Substantial Similarity Analysis

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

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

The law in the Ninth Circuit is no different:

> For an unauthorized use of a copyrighted work to be actionable, the use must be significant enough to constitute infringement. *See Ringgold v. Black Entm't Television, Inc.,* 126 F.3d 70, 74-75 (2d Cir. 1997). This means that ***even where the fact of copying is conceded, no legal consequences will follow from that fact 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.

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

Thus, the issues presented by the presence of isolated identical or very similar elements in two works remain the same as in all copyright cases—namely, whether those similarities relate to protectable elements and, if they do, whether they are sufficient to support a finding that the two works are, in their entireties, "substantially similar" or, in cases of thin copyright protection, "virtually identical" as to copyrightable elements.[10]

---

[10] Indeed, one Seventh Circuit decision in 1990 held to the contrary, and stated that "Establishing substantial similarity is necessary only when direct evidence of copying is unavailable." *Illinois 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*

653387.01

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

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

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

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

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

H.   The implementing code for the Android API packages do not "borrow from other APIs" in the same pattern as the implementing code for the Oracle API packages.

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

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

653387.01

of the HashTable class to provide that functionality.

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

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

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

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

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

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

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

14

653387.01

such organization "would be really hard to use from the developer's, the software developer's standpoint." RT 619:20-21 (Reinhold).

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

> Q.   On any of the occasions while you were at Sun when you worked on reimplementing an existing API, did you ever change any of the elements of the method declaration for an existing method?
>
> A.   No. We couldn't.
>
> Q.   Why couldn't you?
>
> A.   Because it wouldn't work any more, because programs that had been written to use that API would no longer work. You would compile them and 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.

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

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

653387.01

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   /s/ Robert A. Van Nest
                                                      By: ROBERT A. VAN NEST
13
                                                             Attorneys for Defendant GOOGLE INC.
14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

653387.01