MORRISON & FOERSTER LLP
MICHAEL A. JACOBS (Bar No. 111664)
mjacobs@mofo.com
KENNETH A. KUWAYTI (Bar No. 145384)
kkuwayti@mofo.com
MARC DAVID PETERS (Bar No. 211725)
mdpeters@mofo.com
DANIEL P. MUINO (Bar No. 209624)
dmuino@mofo.com
755 Page Mill Road, Palo Alto, CA 94304-1018
Telephone: (650) 813-5600 / Facsimile: (650) 494-0792

BOIES, SCHILLER & FLEXNER LLP
DAVID BOIES (Admitted *Pro Hac Vice*)
dboies@bsfllp.com
333 Main Street, Armonk, NY 10504
Telephone: (914) 749-8200 / Facsimile: (914) 749-8300
STEVEN C. HOLTZMAN (Bar No. 144177)
sholtzman@bsfllp.com
1999 Harrison St., Suite 900, Oakland, CA 94612
Telephone: (510) 874-1000 / Facsimile: (510) 874-1460

ORACLE CORPORATION
DORIAN DALEY (Bar No. 129049)
dorian.daley@oracle.com
DEBORAH K. MILLER (Bar No. 95527)
deborah.miller@oracle.com
MATTHEW M. SARBORARIA (Bar No. 211600)
matthew.sarboraria@oracle.com
500 Oracle Parkway, Redwood City, CA 94065
Telephone: (650) 506-5200 / Facsimile: (650) 506-7114

*Attorneys for Plaintiff*
ORACLE AMERICA, INC.

UNITED STATES DISTRICT COURT

NORTHERN DISTRICT OF CALIFORNIA

SAN FRANCISCO DIVISION

ORACLE AMERICA, INC.

Plaintiff,

v.

GOOGLE INC.

Defendant.

Case No. CV 10-03561 WHA

**ORACLE'S COMMENTS REGARDING THE COURT'S DRAFT SPECIAL VERDICT FORM**

Dept.: Courtroom 8, 19th Floor
Judge: Honorable William H. Alsup



Dockets.Justia.com

At the Court's request, Oracle America, Inc. ("Oracle") provides the following comments on the Court's draft special verdict form distributed on April 25, 2012. As portions of this verdict form depend on the Court's jury instructions (including the definitions of "implementation," "documentation," and "work as a whole"), Oracle wishes to reserve further comments until after the jury instructions have been circulated.

## I. API "PACKAGES"

For clarity and accuracy, Oracle suggests adding the term "packages" after "APIs" in questions 1 ("37 API *packages*"), 1.A (same), 2 (same), 2.A ("API *package* implementation"), and 2.C ("individual API *package*"). As established through witness testimony, the "37 Java APIs" asserted in this case are most accurately described as "packages," each of which contain numerous sub-elements. Dr. Reinhold, Chief Java Architect at Oracle, explained: "Packages are the highest level concept. Packages can contain classes and interfaces." (RT at 593:8-10.) As an example, Dr. Reinhold described the "java.nio.channels package" (one of the 37 in this case) and its constituent parts, including classes, interfaces, methods, and fields, organized in a hierarchical structure. (RT at 593:14-597:8.) The term "API package" properly conveys the multi-element nature of the 37 copied works.

Because some witnesses have used "API" to refer to a single method within a package (RT at 1770:16-24 (Bornstein) (describing "max API")), the jury may misunderstand the term "37 APIs" to mean only 37 methods (or other sub-elements), rather than 37 "packages" of elements. For these reasons, Oracle proposes that the "37 APIs" be referred to as "API packages."

## II. QUESTIONS ON VICARIOUS AND CONTRIBUTORY INFRINGEMENT

In addition to Oracle's claim against Google for direct copyright infringement, Oracle also asserts claims for vicarious and contributory infringement by virtue of Google's distribution of infringing Android software to third parties. Oracle has presented evidence establishing that third parties have installed the infringing Android software (including the 37 Java API packages) on their devices. This evidence includes, *inter alia*, (1) the testimony of Rafael Camargo, Senior Director at device-maker Motorola Mobility, who testified that Android software is installed on Motorola's Droid and other devices (RT at 1047:21-24; TX 1064); and (2) the testimony of

Daniel Morrill, Google's Technical Program Manager for Android Compatibility, who testified that OEMs including Motorola, Samsung, HTC, and LG all make certified Android-compatible devices, which by definition include implementations of the 37 Java API packages. (RT at 1016:6-1017:16.)

The Court's draft verdict form does not include questions covering Oracle's indirect infringement claims. Oracle proposes adding the following questions to the verdict form:

*4. If you found in Questions 1, 2, or 3 that Google copied Oracle's works and that the copying was not excused as "fair use" or de minimis, then answer the following questions:*

*(a) Has Oracle proven that third-party mobile device manufacturers or end-users infringed Oracle's copyrights by copying or using Android software?*

*(b) (If you answered "yes" to 4(a)) Has Oracle proven that Google intentionally induced or materially contributed to the third-party infringement you found in Question No. 4(a)?*

*(c) (If you answered "yes" to 4(a)) Has Oracle proven that Google vicariously infringed Oracle's copyrights as a result of the third-party infringement you found in Question No. 4(a)?*

## III. THE JURY SHOULD BE INSTRUCTED ON DERIVATIVE WORKS

The Court's draft verdict form omits the question from the Court's earlier version regarding derivative works. If the issue of liability for derivative works is not presented in the verdict form, the jury should at least be instructed regarding derivative works. Google is arguing that it "transformed" the work under the doctrine of fair use. Jurors may incorrectly assume that Google's derivative works are fair use transformations if not instructed that infringement includes derivative works.

The Copyright Act separately protects the copyright owner's exclusive rights to "prepare derivative works based upon the copyrighted work." 17 U.S.C. § 106(2). A "derivative work" is:

> a work based upon one or more preexisting works, such as a translation, musical arrangement, dramatization fictionalization, motion picture version, sound recording, art reproduction,

> abridgement, condensation, or any other form in which a work may be *recast, transformed, or adapted.* A work consisting of editorial revisions, annotations, elaborations, or other modifications which, as a whole, represent an original work of authorship, is a "derivative work."

17 U.S.C. § 101 (emphasis added). Thus, creation of a work that "recast[s], transform[s], or adapt[s]" a copyrighted work is the basis for a finding of infringement even in the absence of actual copying. *See, e.g.*, *Apple, Inc. v. Psystar Corp.*, 673 F. Supp. 2d 931, 938-39 (N.D. Cal. 2009) (finding that modifications to software to enable Mac OS X to run on a non-Apple computer constituted infringing, derivative work: "[d]eletions, modifications, and additions to software result in an infringing derivative work"); *Dun & Bradstreet Software Servs., Inc. v. Grace Consulting, Inc.*, 307 F.3d 197, 208 (3rd Cir. 2002) (concluding that defendant's modification of a copy of plaintiff's software, including fixing bugs and adding features, created an infringing derivative work); *Midway Mfg. Co. v. Artic Int'l, Inc.*, 704 F.2d 1009, 1013-14 (7th Cir. 1983) (finding defendant's addition of circuit boards creating a speeded-up video game was a derivative work); *SAS Inst., Inc. v. S & H Computer Sys., Inc.*, 605 F. Supp. 816, 831 (M.D. Tenn. 1985) (finding defendant's duplication and conversion of software that would run on an IBM computer to run on a VAX computer constituted a derivative work).

Here, ample evidence shows that Google created infringing, derivative works using Oracle's copyrighted Java API specifications and implementations:

1. Former Google engineer Bob Lee testified that Google looked at the Java API specifications and transformed them into Android source code. (RT at 982:22-983:12.) Mr. Lee, who was the core library lead for the Android team, admitted that he consulted the Java API specifications available on Sun's website to make sure that the Android code for the corresponding core libraries would be consistent with those specifications. (*Id.*)

2. The jury heard testimony from Professor Mitchell that Google used a decompiler to create Android source code derived from the Java bytecode, a process that would not have been possible in a clean room implementation. (RT at 1257:14-1258:6 ("one of [the Java] class files that are widely distributed was used to produce source code that's in the Google library"); 1259:16-1260:7 (comparison of source code files shows "strong evidence that the Google code

was actually produced from the Oracle class file by a particular decompiler called JAD for Java Decompiler"); 1260:8-1261:3 (analysis shows that "developers had access and used the Oracle installation and copied from it . . .using a decompiler to produce source code"); 1261:14-1262:1.) Google's recasting of the bytecode into a different form created a derivative work.

This evidence could support a jury finding that Google created derivative works, in addition to copying the API packages. In the absence of a derivative work instruction, the jury may incorrectly assume that Google's derivative works constitute "transformative use" under the fair use doctrine. Google is arguing that Android is a "transformative use," under the first of the fair use statutory factors identified in recent case law. The word "transform" also appears in the Ninth Circuit Model Jury Instruction on derivative use, as part of the definition for derivative work. Ninth Circuit Manual of Model Jury Instruction No. 17.13 (2007). Acknowledging the risk of juror confusion between transformative use under the fair use doctrine and the derivative work doctrine, the Comment to Instruction No. 17.13 explains that "A derivative work is saved from being an infringing work 'only because the borrowed or copied material [in the derivative work] was taken with the consent of the copyright owner of the prior work, or because the prior work has entered the public domain." *Id.* at Comment (citing *Micro Star v. Formgen*, 154 F.3d 1107, 1112 (9th Cir. 1998)).

Given this risk of juror confusion, the Court should instruct the jury on derivative works, even if the issue is not included on the special verdict form. Both parties submitted jury instructions on derivative works; in fact, of all the copyright instructions proposed, this instruction was the closest to reaching full stipulation. (ECF No. 539, Joint Proposed Jury Instructions at 45-47 (Instruction No. 14).)

## IV. THE JURY SHOULD BE INSTRUCTED THAT SELECTION AND ARRANGEMENT OF NAMES WITHIN THE 37 JAVA API PACKAGES IS PROTECTABLE

During its opening statement and in proceedings before the Court, Google has repeatedly referenced the Court's earlier ruling that the individual names of Java API package elements ("packages, classes, and methods") are not protectable by copyright. (*See* ECF No. 433, 9/15/11 Order on Summ. J. at 7-8.) Omitted from these discussions is the fact that the Court specifically

left open "the possibility that the selection or arrangement of those names is subject to copyright protection." (*Id*. at 8.) The Court observed that "[c]opyright protection for the selection and arrangement of elements within a work is a separate question from whether the elements themselves are protected by copyright." (*Id.*) *See Lamps Plus, Inc. v. Seattle Lighting Fixture Co.*, 345 F.3d 1140, 1147 (9th Cir. 2003) ("a combination of unprotectable elements is eligible for copyright protection only if those elements are numerous enough and their selection and arrangement original enough that their combination constitutes an original work of authorship").

The Court's draft verdict form does not specify whether the selection and arrangement of names can be considered by the jury in evaluating the "structure, sequence, and organization" of the 37 Java API packages. Oracle does not propose modifications to the verdict form on this issue. Oracle requests, however, that the Court instruct the jury that the selection and arrangement of names within the Java API packages may be eligible for copyright protection (*see* ECF No. 539, Oracle's Proposed Jury Instruction No. 7).

Dated: April 25, 2012

MORRISON & FOERSTER LLP

By: *___/s/ Michael A. Jacobs___*

*Attorneys for Plaintiff*
ORACLE AMERICA, INC.