

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

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

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

16 *Attorneys for Plaintiff*
 ORACLE AMERICA, INC.

18 UNITED STATES DISTRICT COURT
 19 NORTHERN DISTRICT OF CALIFORNIA
 20 SAN FRANCISCO DIVISION

21 ORACLE AMERICA, INC.

22 Plaintiff,

23 v.

24 GOOGLE INC.

25 Defendant.

Case No. CV 10-03561 WHA

**ORACLE AMERICA, INC.'S
 RULE 50(A) MOTION AT THE
 CLOSE OF ALL EVIDENCE**

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

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28

I. INTRODUCTION

Google has infringed Oracle’s registered copyrights in Java 2 Standard Edition, Version 1.4 (“J2SE 1.4”) and Java 2 Standard Edition, Version 5.0 (“J2SE 5.0”) in three distinct ways: (1) Google copied into Android the structure, sequence, and organization (“SSO”) of 37 Java API packages; (2) Google copied Java code into twelve Android code files; and (3) Google copied material from Java API specifications into Android’s specifications. All three varieties of copying were submitted to the jury for resolution.¹ However, in light of Google’s admissions and the weight of the evidence, Google’s copyright infringement can be determined as a matter of law, as no reasonable jury could find for Google.

The SSO is the backbone of the 37 Java API packages, supplying both taxonomy and definition. Within the Android and Java code, the SSO is embodied in the names of the API elements (packages, classes, interfaces, methods, and fields), the organization and interrelationships among those elements, and the class and method declarations, and is expressed in more than 7000 lines of code. This SSO, as a core feature of the copyrighted Java software platform, is protected by copyright. *Johnson Controls, Inc. v. Phoenix Control Sys., Inc.*, 886 F.2d 1173, 1175 (9th Cir. 1989).

Assuming the Court confirms the copyrightability of the SSO under governing Ninth Circuit authority, Google’s infringement is manifest. Google and its witnesses admit that (1) the Java APIs are original; (2) Android engineers and contractors directly copied the SSO of the 37 Java API packages from Java specifications into Android; (3) the SSO is not *de minimis*; (4) Google copied Java code into twelve Android code files; and (5) Google paraphrased the Java specifications in writing the Android specifications. The unabashedly commercial, non-transformative nature of Google’s copying undermines its fair use defense. And Google has

¹ With respect to specification copying, the Court only submitted the issue of whether English-language descriptions were copied. (ECF No. 1018, Jury Instruction (“JI”) 21.) As discussed further below, Oracle reiterates its position that the descriptions should not be viewed in isolation and that the specifications also express the SSO of the 37 API packages which was copied into Android’s code and specifications. Judgment can be granted on this issue as a matter of law.

1 failed to meet its burden of proving its equitable defenses. For all of these reasons, Oracle is
2 entitled to judgment as a matter of law.

3 **II. LEGAL STANDARD FOR JUDGMENT AS A MATTER OF LAW**

4 Judgment as a matter of law is appropriate when “a party has been fully heard on an issue
5 during a jury trial and the court finds that a reasonable jury would not have a legally sufficient
6 evidentiary basis to find for the party on that issue.” Fed. R. Civ. P. 50(a)(1). “[T]he trial judge
7 must direct a verdict if, under the governing law, there can be but one reasonable conclusion as to
8 the verdict.” *Anderson v. Liberty Lobby, Inc.*, 477 U.S. 242, 250 (1986).

9 **III. NO REASONABLE JURY COULD FIND THAT GOOGLE DID NOT 10 INFRINGE ORACLE’S JAVA-RELATED COPYRIGHTS**

11 To establish copyright infringement, Oracle must show that: (1) Oracle is the owner of an
12 original copyrighted work; and (2) Google copied elements from the copyrighted work. (ECF
13 No. 1018, JI 23-24.) *Feist Publ’ns, Inc. v. Rural Tel. Serv. Co., Inc.*, 499 U.S. 340, 361 (1991).
14 Copying can be established either by (1) proving direct copying, or (2) showing that the accused
15 infringer had access to the copyrighted work and there are substantial similarities or virtual
16 identity between the copyrighted work and the accused work. (ECF No. 1018, JI 24.) *Range*
17 *Road Music, Inc. v. East Coast Foods, Inc.*, 668 F.3d 1148, 1154 (9th Cir. 2012) (“‘substantial
18 similarity’ is irrelevant in a case [involving] direct copying of copyrighted works”); *Jada Toys,*
19 *Inc. v. Mattel, Inc.*, 518 F.3d 628, 636-37 (9th Cir. 2008). Under the second approach, the Court
20 has determined that copying of the SSO and twelve code files should be evaluated using the
21 substantial similarity test, while copying of the API specifications should be evaluated using the
22 virtual identity test.² (ECF No. 1018, JI 24.)

23 As discussed below, all elements of copyright infringement have been established in this
24 case by Google’s own stipulated admissions, the clear admissions of Google witnesses, and/or the
25 overwhelming weight of the evidence.

27 ² Oracle maintains its objection to the Court’s use of the virtual identity test for evaluating
28 the copying of the API specifications. (*See* ECF No. 1016 at 2.)

1 **A. Oracle Owns The Asserted Copyrights³**

2 The issue of copyright ownership was not posed to the jury. (ECF No. 1018, JI 23.)
3 Oracle owns the copyrights to J2SE 1.4 and J2SE 5.0. The copyright registrations were admitted
4 into evidence (TX 464, 475, 476), along with the complete contents of both versions of the
5 platform. (RT 689:21-691:23 (Reinhold); TX 610.2, 622, 623, 1076.) Oracle is entitled to a
6 presumption that the facts stated on the front of the copyright registrations are correct. 17 U.S.C.
7 § 410(c) (certificate of registration “shall constitute prima facie evidence of the validity of the
8 copyright and of the facts stated in the certificate”). Google “has the burden of rebutting the facts
9 set forth in the copyright certificate.” *United Fabrics Int’l, Inc. v. C&J Wear, Inc.*, 630 F.3d
10 1255, 1257 (9th Cir. 2011). Google has submitted no evidence to rebut the facts reflected in
11 Oracle’s copyright registrations.

12 Accordingly, Oracle is entitled to judgment that it is the owner of valid copyrights to J2SE
13 1.4 and J2SE 5.0, covering all components of those works, including the individual code files and
14 the SSO of the API packages.

15 **B. Google Infringed By Copying The SSO Of The 37 Java API Packages**

16 Google’s admissions and the weight of the evidence prove that (1) the Java API packages
17 are original, (2) Google engineers directly copied the SSO of the 37 API packages into Android,
18 and (3) the SSO of the 37 packages is not *de minimis*. If the Court confirms that the SSO of the
19 API packages is copyrightable (addressed below), infringement will follow as a matter of course.

20 **1. Originality**

21 Google has not challenged the originality of the copyrighted Java API packages, and,
22 accordingly, this issue was not posed to the jury. Google has admitted that the Java APIs meet
23 the required threshold: “The Java APIs as a whole meet the low threshold for originality required
24 by the Constitution.” (ECF No. 938 (Order deeming issues undisputed based on Google’s
25 admission) at 1.) Moreover, because Oracle holds registered copyrights covering the asserted

26
27 ³ This issue will be addressed in more detail in Oracle’s Findings of Fact and Conclusions
28 of Law regarding issues for the Court to resolve, to be filed on May 2, 2012. Oracle incorporates
that discussion herein by reference.

1 works, it is entitled to a presumption of originality. *Swirsky v. Carey*, 376 F.3d 841, 851 (9th Cir.
2 2004) (“Because *One* has a valid certificate of registration with the copyright office, however,
3 Swirsky is entitled to a presumption of originality”).

4 The evidence of originality is overwhelming and uncontroverted. The threshold for
5 establishing originality is low. *Feist*, 499 U.S. at 345 (“Original, as the term is used in copyright,
6 means only that the work was independently created by the author (as opposed to copied from
7 other works), and that it possesses at least some minimal degree of creativity”). Numerous
8 witnesses confirmed that API design is highly creative and challenging. (RT 513:14-18
9 (Screven); 627:21-628:1 (Reinhold).) Dr. Bloch stated that designing an API is a work of
10 craftsmanship, requiring the API author to exercise creativity and aesthetic judgment. (RT 751:5-
11 752:14; TX 624 at p. 47.) Dr. Reinhold testified that it took him and a team of engineers two
12 years to develop just one API package, java.nio. (RT 623:17-624:1.) Based on the evidence and
13 Google’s admission, the originality of the 37 API packages is beyond dispute.

14 2. Direct Copying

15 Where there is evidence of direct copying, there is no need to apply the substantial
16 similarity test. *Range Road*, 668 F.3d at 1154 (“‘substantial similarity’ is irrelevant in a case
17 [involving] direct copying of copyrighted works”). In this case, the evidence shows that Google
18 directly copied the SSO of the 37 Java API packages into Android.

19 Former Google engineer Bob Lee was the core libraries lead for Android in the 2006-2007
20 time period. (RT 980:14-981:6.) Mr. Lee testified that Android implements parts of the Java
21 APIs and that such implementation was based on consulting the Java API specifications:

22 Q. You consulted the Java API specifications to make sure that the Android code for the
23 corresponding core libraries would be consistent with those specifications, correct?

A. Yes.

24 (RT 982:25-983:3; *see also* RT 981:7-21.) Mr. Lee also testified that Google’s outside
25 contractor, Noser, was hired to implement core libraries based on the Java API specifications:

26 Q. Noser was an outside contractor hired by Google to implement core libraries according
27 to the Java API specifications, correct?

A. Yes.

1 (RT 985:3-6.) Former Google engineer Dan Bornstein confirmed that his team had used the Java
2 specifications to derive information for implementing the APIs in Android:

3 Q. Now, you testified on direct, sir, that you and your team looked at the Java
4 specifications for these application programming interfaces while you were doing your
5 work on Android; correct, sir?

6 A. That's right.

7 Q. And you did that in order to derive from these specifications the information you
8 needed in order to write that code; correct, sir?

9 A. Some of the information.

10 (RT 1836:19-1837:2.) Because the Java specifications are directly derived from the underlying
11 source code (RT 606:14-608:3 (Reinhold); TX 1046 at p.19), the Android engineers effectively
12 had access to the SSO of the source code itself. The fact that Google literally copied some Java
13 code (discussed below) demonstrates that they had *direct* access to the code as well.

14 By deriving the Android class libraries from the Java API specifications, the Android
15 engineers wound up copying the entire SSO of the 37 API packages into Android. Mr. Bornstein
16 confirmed that “the names and declarations” are the same between Java and Android. (RT
17 1791:21-1792:6 (Bornstein).) Google’s expert, Dr. Astrachan, admitted that the SSO of the 37
18 packages is “virtually identical” between Java and Android. (RT 2214:3-9 (Astrachan).) Based
19 on the evidence of direct copying, Oracle has proven infringement.

20 3. Access and Substantial Similarity

21 Oracle has proven copying under the substantial similarity test as well. That test requires
22 a showing that the infringer had access to the copyrighted work and that the accused and
23 copyrighted works are substantially similar. *See Jada Toys*, 518 F.3d at 636-37. The Java API
24 specifications that detail the SSO of the API packages were publicly available on Sun’s website.
25 (RT 983:4-6 (Lee).) The testimony of Messrs. Lee and Bornstein cited above demonstrates that
26 Google had access to the Java specifications. As for substantial similarity, Google admits that the
27 SSO of the 37 API packages is substantially the same between Java and Android:

28 Google agrees that the structure, sequence and organization of the 37 accused API
packages in Android is substantially the same as the structure, sequence and organization
of the corresponding 37 API packages in Java.

1 (ECF No. 1018, JI 19; *see also* ECF No. 946 (Stipulation); RT 1337:21-24.) Dr. Astrachan went
2 further, agreeing that the SSO in Java and Android is “virtually identical.” (RT 2214:3-9.)

3 During its closing argument, Google compared the code that embodies the SSO of the 37
4 Java API packages in Android to the complete code for the 166 packages in J2SE 5.0 and argued
5 that the two are not substantially similar because the latter is far more voluminous than the
6 former. This approach is wrong as a matter of law. First, as already noted, since the evidence
7 shows that Google engineers and contractors directly copied the SSO from Java specifications,
8 there is no need to apply the substantial similarity test at all. Second, even under the substantial
9 similarity test, an infringer cannot escape liability by arguing that it only took a small portion of
10 the overall copyrighted work that is qualitatively significant. *See Warner Bros. Entm’t Inc. v.*
11 *RDR Books*, 575 F. Supp. 2d 513, 535 (S.D.N.Y. 2008) (finding that Harry Potter Lexicon
12 “copie[d] a sufficient quantity of the Harry Potter series to support a finding of substantial
13 similarity,” even if the copied elements “amount to only a fraction of the seven-book series”);
14 *Paramount Pictures Corp. v. Carol Publ’g Group*, 11 F. Supp. 2d 329, 334 (S.D.N.Y. 1998)
15 (“Copying only small portions of a series of copyrighted works offers no protection for a
16 defendant”).

17 Under Google’s theory, a copier who takes a portion of a piece of software could never be
18 liable, even when the copied portion is qualitatively significant, as is the case with the SSO of the
19 37 API packages. While the number vastly understates its significance, the SSO is embodied in
20 more than 7,000 lines of code (names and declarations), without which the Android platform
21 would not function as designed. (RT 2212:3-19 (Astrachan).) The copied SSO in Android is
22 qualitatively substantial even as compared to the 166 Java API packages, and the two are
23 substantially similar in that regard.

24 **4. Google Admits That The SSO Is Not *De Minimis***

25 The Court instructed the jury that to find copyright infringement it must also find that the
26 amount of copying was not *de minimis*. (ECF No. 1017, JI 24.) However, Google dropped this
27 argument as to the SSO of the 37 API packages on the last day of trial. As reflected in the jury
28

1 instructions, the “parties are in agreement that the structure, sequence, and organization of the
2 API packages is more than *de minimis*.” (ECF No. 1018, JI 27.)

3 Between Google’s admissions and the evidence in the case, Oracle has proven that Google
4 infringes Oracle’s copyrights by copying the SSO of the 37 Java API packages into Android.

5 **C. Google Infringed By Copying Java Code Into Twelve Android Code**
6 **Files**

7 Google admits that it copied protected Java code into twelve Android code files. The only
8 issue here is whether the copying was *de minimis*. As explained below, it was not.

9 **1. Direct Copying**

10 There is no dispute that Google copied Java code and comments into twelve Android code
11 files. (TX 1072.) As reflected in the jury instructions, Google admits copying the code:

12 With respect to the infringement issues concerning the rangeCheck and other similar files,
13 Google agrees that the accused lines of code and comments came from the copyrighted
14 material.

15 (ECF No. 1018, JI 27.) With respect to the rangeCheck code within the TimSort.java and
16 ComparableTimSort.java files, Josh Bloch, who authored the code while working at Sun,
17 acknowledged the likelihood that he copied that code from Sun copyrighted code. (RT 827:5-17.)
18 Dr. Mitchell confirmed that the accused code and comments in the twelve Android files are
19 identical to the copyrighted Java code and comments (RT 1254:1-1255:15, 1258:24-1260:18,
20 1262:2-1263:10 (Mitchell)), and Google did not offer any testimony to rebut this.

21 **2. Google’s Code Copying Was Not *De Minimis***

22 Copying can only be *de minimis* “if it is so meager and fragmentary that [compared to the
23 work as a whole] the average audience would not recognize the appropriation.” (ECF No. 1018,
24 JI 28.) *Fisher v. Dees*, 794 F.2d 432, 434 n.2 (9th Cir. 1985). The extent of the copying “is
25 measured by considering the qualitative and quantitative significance of the copied portion in
26 relation to the plaintiff’s work as a whole.” *Newton v. Diamond*, 388 F.3d 1189, 1195 (9th Cir.
27 2003); *see also Merch. Transaction Sys., Inc. v. Nelcela, Inc.*, 2009 U.S. Dist. LEXIS 25663, at
28 *61 (D. Ariz. Mar. 17, 2009) (“Thus, Nelcela will not escape liability unless it can show that the
protectable elements in the Lexcel software constitute an insignificant (quantitatively and

1 qualitatively) portion or aspect of the Lexcel software”). Even if the copied material is a
2 “quantitatively very small part” of the work as a whole, “[t]he smallness alone is not enough by
3 itself to avoid liability.” *Mktg. Tech. Solutions, Inc. v. Medizine LLC*, 2010 U.S. Dist. LEXIS
4 50027, at *10 (S.D.N.Y. Apr. 23, 2010).

5 Under the Court’s jury instructions, the copied code and comments in the twelve Android
6 files must be compared to the contents of the corresponding Java files from which the copied
7 material came. (ECF No. 1018, JI 29.) Professor Mitchell provided detailed and uncontested
8 testimony that Google literally copied either the entirety of the accused files or a qualitatively
9 significant portion. Google has made no showing that the copied code was “so meager and
10 fragmentary” that it would not be recognized as an appropriation. Under the appropriate
11 comparison, no reasonable jury could find that Google’s code copying was *de minimis*.

12 **a. Google’s Copying of Eight Decompiled Files Was Not *De***
13 ***Minimis*.**

14 During trial, Professor Mitchell explained that Google decompiled eight Java files and
15 copied the decompiled source code into certain Android files. (RT 1258:7-1259:15 (Mitchell);
16 TX 1072, 896.1-896.8, 1031-1038.) Google does not dispute that these files were copied. (ECF
17 No. 1018, JI 27.)

18 When compared on a file-to-file basis (per the Court’s jury instructions), it is clear that
19 Google’s copying of the eight files was not *de minimis*. Google copied each entire file, so by
20 definition, the copying is both quantitatively and qualitatively significant. (*See* RT 1260:11-
21 1262:1 (Mitchell); *compare* TX 896.1-896.8 (Sun decompiled files) *with* TX 1031-38 (Android
22 files).) No reasonable jury could find that the copying of the decompiled files was *de minimis*.

23 **b. Google’s Copying of the RangeCheck Method Was Not**
24 ***De Minimis*.**

25 Professor Mitchell testified that the rangeCheck method is qualitatively significant and
26 “useful” to Android. (RT 1316:17-19.) He explained that the rangeCheck method operates on
27 Android devices, including on Samsung phones. (RT 1255:22-25, 1264:19-23.) To determine
28 how useful the rangeCheck method is on devices, Professor Mitchell experimented with an

1 Android device and found that it called the rangeCheck method no less than 2,600 times during
2 start up. (RT 1329:9-21.) Google did not present any testimony to rebut this.

3 **c. Google’s Copying Of Comments Was Not *De Minimis***

4 Professor Mitchell compared Oracle’s CodeSource.java file (TX 623.9) against Android’s
5 CodeSourceTest.java file (TX 1039) and concluded that, except for some HTML commands,
6 certain English-language comments are “syntactically . . . identical.” (RT 1262:13-1263:4.) He
7 reached the same conclusion with respect to Android’s CollectionCertStoreParametersTest.java
8 file. (RT 1253:9-10; *compare* TX 1040 (Android) to TX 623.10 (Java)). These literally copied
9 comments are quantitatively significant: they amount to about 25% of Oracle’s
10 CollectionCertStoreParameters.java file (TX 623.10), and about 2.90% of Oracle’s
11 CodeSource.java file (TX 623.9).

12 **D. Google Infringed By Copying From Java Specifications Into Android**
13 **Specifications**

14 The Court submitted to the jury the question of whether Google has infringed Oracle’s
15 copyrights by copying the English-language descriptions of API elements from the Java
16 specifications into the Android specifications. (ECF No. 1018, JI 21.) The Court did not submit
17 the issue of whether the SSO of the 37 API packages, as expressed in the Java specifications, was
18 also copied into the Android specifications. These issues should be considered together. Oracle
19 is entitled to judgment on both of these issues, based on the weight of the evidence.

20 **1. The Evidence Shows That Android’s English-Language**
21 **Descriptions Were Copied From The Java API Specifications.**

22 Android developer Bob Lee testified that the English-language descriptions within the
23 Android specifications were paraphrased from Sun’s specifications and were therefore
24 “substantially similar.” (RT 1191:4-13.) He was shown three examples of such paraphrasing,
25 and he acknowledged that the same level of similarity evident in those examples exists across the
26 full documentation for the 37 Java API packages within Android. (RT 1175:25-1176:3; TX
27 610.2, 767 (full Android and Java documentation).) One of the examples is shown below:
28

J2SE 5.0	Android
<p>A pair of channels that implements a unidirectional pipe.</p> <p>A pipe consists of a pair of channels: A writable sink channel and a readable source channel. Once some bytes are written to the sink channel they can be read from source channel in exactly the order in which they were written.</p>	<p>A pipe contains two channels, forming a unidirectional pipe. One is the writable sink channel, and the other is the readable source channel. When bytes are written into the writable channel they can be read from the readable channel. Bytes are read in the order in which they were written.</p>

Mr. Lee expressed regret that such paraphrasing had occurred:

I actually wasn't even a big fan of including these. I would have preferred that we just point people to Sun's site for this specific documentation because you shouldn't really be rewriting a contract. And in doing so they are going to be substantially similar.

(RT 1175:25-1176:3.) Mr. Lee's admission that the English-language descriptions in the Android specifications were paraphrased from the Java specifications is evidence of direct copying, which entitles Oracle to judgment on this point.

2. Evidence Also Shows That Google Copied The SSO Of The Java API Packages From Java Specifications Into Android Specifications

While the Court only submitted the issue of English-language description copying to the jury, Google is also liable for copying the SSO of the Java API packages into the Android documentation. The facts are uncontroverted and Oracle is entitled to judgment on this issue.

There is no question that Google copied the SSO of the 37 Java API packages into the Android documentation. As explained by Dr. Reinhold, the structure expressed in the API documentation is the same as the structure within the compilable code because the code is run through the Java Documentation Extractor (or "Javadoc") to pull out the structure and English language comments to produce a webpage that reflects the same SSO of the API that is in the code. (RT 606:14-608:3 (Reinhold); TX 1046 at p.19.) The Android documentation is created the same way from the Android code. (RT 1169:8-15 (Lee agreeing that like Java documentation, Android documentation was "created by a tool that actually reads portions of the source code and then places it in a kind of template that's available on the web as a source of documentation."))

1 It is undisputed that the SSO of the 37 Java API packages is identical in both the Java and
2 Android code and documentation. (ECF No. 984 at 10 (Google’s JMOL motion: “As such, if
3 based on the same starting point – the names of the 37 API packages at issue – the structure,
4 sequence, and organization of the Android and Java documentation inevitably will be the same.”))

5 Regardless of whether the SSO is expressed in the compilable code or the API
6 documentation, it is protectable expression in both cases. *See, e.g., Situation Mgmt. Sys., Inc. v.*
7 *ASP Consulting Group*, 560 F.3d 53, 61 (1st Cir. 2009) (“creative choices [in training manuals] in
8 describing those processes and systems, including the works’ overall arrangement and structure,
9 are subject to copyright protection” even when describing uncopyrightable system); *CDN Inc. v.*
10 *Kapes*, 197 F.3d 1256, 1262 (9th Cir. 1999) (prices in guide for collectible coins); *Jacobsen v.*
11 *Katzer*, 2009 U.S. Dist. LEXIS 115204, at *9-10 (N.D. Cal. Dec. 10, 2009) (text files reflecting
12 decoder information from model railroad manufacturers).

13 As currently presented to the jury, it is possible Google will be found liable for copying
14 the SSO into the Android code, but not liable for copying the very same SSO into Android
15 specifications. That makes no sense. Oracle is entitled to judgment on both forms of copying.

16 **E. Google’s Copying Is Not Fair Use**

17 As the Court instructed the jury, Google had the burden to prove its fair use defense based
18 on the four factors identified in 17 U.S.C. § 107(1). (ECF No. 1018, JI 26.) As demonstrated
19 below, consideration of those factors weighs overwhelmingly against a finding of fair use.
20 Because no reasonable juror could find based on the trial evidence that Google has met its burden,
21 Oracle is entitled to judgment as a matter of law on Google’s fair use defense.

22 **1. Google’s Use Of The Copyrighted Work Is Purely Commercial 23 And Not “Transformative”**

24 The first fair use factor—“the purpose and character of the use, including whether such
25 use is of a commercial nature or is for nonprofit educational purposes” (17 U.S.C. § 107(1))—
26 weighs entirely against a finding of fair use. Google exploits Oracle’s copyrighted work for
27 significant commercial gain, and its use of the work is not “transformative.”
28

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28

a. Google’s Use Is Purely Commercial

“Although not controlling, the fact that a new use is commercial as opposed to non-profit weighs against a finding of fair use.” *Elvis Presley Enters. Inc. v. Passport Video*, 349 F.3d 622, 627 (2003). As explained by the Supreme Court:

“[Every] commercial use of copyrighted material is presumptively an unfair exploitation of the monopoly privilege that belongs to the owner of the copyright.” *Sony Corp. of America v. Universal City Studios, Inc.*, 464 U.S. [417, 451 (1984)]. ... The crux of the profit/nonprofit distinction is not whether the sole motive of the use is monetary gain but whether the user stands to profit from exploitation of the copyrighted material without paying the customary price.

Harper & Row Publishers, Inc. v. Nation Enters., 471 U.S. 539, 562 (1985) (citations omitted); see also *Passport Video*, 349 F.3d at 627 (“[T]he degree to which the new use exploits the copyright for commercial gain—as opposed to incidental use as part of a commercial enterprise—affects the weight we afford commercial nature as a factor”).

In *Harper & Row*, the Court found that the first factor weighed against a news magazine’s use of quotes from Gerald Ford’s unpublished presidential memoir because of the magazine’s “stated purpose of scooping the forthcoming hard cover and Time abstracts” and despite its “not purely commercial” purpose of “news reporting.” 471 U.S. at 562. Similarly, in *Passport Video*, the Ninth Circuit upheld the district court’s finding that the first factor weighed against fair use for television performance excerpts in a broadcast Elvis Presley biography, despite acknowledging that “[i]t would be impossible to produce a biography of Elvis without showing some of his most famous television appearances for references purposes.” 349 F.3d at 629.

Here, unlike in *Harper & Row* or *Passport Video*, it is undisputed that Google’s use of the Java APIs is purely for commercial purposes. Google has “profit[ed] from exploitation of the copyrighted material without paying the customary price” (*Harper & Row.*, 471 U.S. at 562) and “exploit[ed] the copyright for commercial gain” (*Passport Video*, 349 F.3d at 627).

Google’s distribution of the Android platform is to increase use of Google services, which generate advertising revenue for Google. (RT 1458:12-16 (Schmidt)). Google copied the 37 Java API packages in order to capture a large developer community and penetrate the market more quickly. (See RT 1783:15-22 (Bornstein)) (“The goal of the project was to provide something that

1 was familiar to developers”). Indeed, the evidence showed that Android was hugely profitable.
2 (RT 1458:12-16; 1456:15-19 (Schmidt); 2225:18-2226:24 (Agrawal).) One Google internal
3 document described how Android and Chrome were “critical” platforms for five different Google
4 business units, each one identified as a \$10 billion opportunity for Google. (TX 431 at 3.) This
5 evidence of Google’s purely commercial use of the Java APIs weighs heavily against fair use.

6 **b. Google’s Use Was Not Transformative**

7 Google argues that even though its use of the Java APIs was purely commercial and
8 highly exploitative, its use was “transformative.” This argument fails as a matter of law because
9 Google has presented no evidence to support a conclusion that its use of Java APIs is
10 “transformative” within the meaning of controlling case law.

11 The leading case on “transformative” use is *Campbell v. Acuff-Rose Music, Inc.*, 510 U.S.
12 569 (1994). *Campbell*, which involved a claim of fair use in a song parody, held that “parody,
13 like other comment or criticism, may claim fair use under § 107.” 510 U.S. at 579. Ninth Circuit
14 cases following *Campbell* have emphasized that “transformative” is intended to refer to a work—
15 like a criticism or a parody—that has a purpose entirely different from the original and is not
16 intended to apply to a competing work with a parallel object or purpose. For example, in *Kelly v.*
17 *Arriba Soft Corp.*, 336 F.3d 811 (9th Cir. 2003), in holding that a search engine operator’s use of
18 “thumbnail” pictures of copyrighted images was “transformative” fair use, the court explained
19 that,

20 [T]he thumbnails were much smaller, lower-resolution images that served *an*
21 *entirely different function* than Kelly’s original images. Kelly’s images are
22 artistic works intended to inform and to engage the viewer in an aesthetic
23 experience Arriba’s use of Kelly’s images in the thumbnails is *unrelated to*
24 *any aesthetic purpose*.

25 336 F.3d at 818 (emphasis added).

26 Similarly, in *Perfect 10, Inc. v. Amazon.com, Inc.*, 487 F.3d 701 (9th Cir. 2007), which
27 followed *Kelly* and held that Google’s search engine “thumbnail” photographs were
28 “transformative” uses of the plaintiff’s photographs and hence fair use, the Ninth Circuit
emphasized that the search engine used the photographs for an entirely different function:

1 Although an image may have been created originally to serve an entertainment,
2 aesthetic, or informative function, a search engine transforms the image into a
3 pointer directing a user to a source of information [A] search engine provides
4 social benefit by incorporating an original work into a new work, namely, an
5 electronic reference tool In other words, a search engine puts images in a
6 different context so that they are transformed into a new creation.

487 F.3d at 721 (emphasis added) (internal citations and quotations omitted).

5 In *Leadsinger, Inc. v. BMG Music Publ'g*, the Ninth Circuit started its analysis by
6 considering whether the allegedly transformative use of copying song lyrics for karaoke fell
7 within the statutory examples. 512 F.3d 522, 530 (9th Cir. 2008). The Court concluded it did
8 not, and emphasized that “Leadsinger’s basic purpose remains a commercial one — to sell its
9 karaoke device for a profit. And commercial use of copyrighted material is ‘presumptively an
10 unfair exploitation of the monopoly privilege that belongs to the owner of the copyright.’” *Id.* at
11 530 (citation omitted).

12 Here, unlike the parody in *Campbell*, Google’s use of the copied materials in Android is
13 nothing like “the examples given in the preamble to § 107.” *See* 17 U.S.C. § 107. Nor is its use
14 for “an entirely different function” (*Kelly*, 336 F.3d at 818) or “in a new context to serve a
15 different purpose” (*Perfect 10*, 487 F.3d at 722). It is for the same intrinsic purpose—supplying
16 core library APIs in a software platform—as used in the Java platform, but in a competing
17 product. Google simply copied the SSO of the 37 Java API packages over into Android, to serve
18 the same purpose that it does in Java. (RT 2184:22-2185:9 (Astrachan) (“that structure of the
19 names of the classes, packages, and methods needs to be the same so that the code will work on
20 both platforms”).) There is nothing “transformative” about that use.

21 Nor can the fact that Android is a smart phone platform render Google’s use
22 “transformative.” There is nothing new or “transformative” in the fact that Android is a
23 smartphone platform: the uncontroverted evidence at trial established that Java technology is
24 used in the RIM Blackberry smartphones, and was used in the Danger Sidekick/Hiptop
25 smartphones and Nokia’s Series 60 phones. (RT 959:20-23 (Swetland); 1585:21-23 (Rubin);
26 300:18-19 (Ellison); 383:6-9 (Kurian); 1102:3-10 (Cizek); 1922:22-25 (Gering).)

1 If Google’s argument were accepted, the idea of “transformation” would swallow up
2 copyright protection: anyone claiming to have a better business model for distributing the
3 copyrighted work would be able to copy it, sell it, and claim “fair use.” Movie makers would be
4 hard-pressed to enforce their copyrights against infringing distributors where there were no
5 previous distributors; book publishers could not enforce against e-Book publishers if they were
6 not already distributing e-Books; musicians could not enforce against FM or satellite radio
7 stations if their songs were broadcast only on AM stations. This is not the law. As the Ninth
8 Circuit noted in *Perfect 10*:

9 [D]uplicating a church’s religious book for use by a different church was not
10 transformative. *See Worldwide Church of God v. Phila. Church of God, Inc.*, 227
11 F.3d 1110, 1117 (9th Cir. 2000). Nor was a broadcaster’s simple retransmission of
12 a radio broadcast over telephone lines transformative, where the original radio
13 shows were given no “new expression, meaning, or message.” *Infinity Broad.*
14 *Corp. v. Kirkwood*, 150 F.3d 104, 108 (2d Cir.1998).

15 487 F.3d at 722.

16 Here, Google took the SSO of 37 Java API packages that are implemented by Oracle, or
17 by others under license, from the Java documentation and computer software and copied it into
18 the Android documentation and computer software. This is not “an entirely different function”
19 (*Kelly*, 336 F.3d at 818); it is the same function and “the same intrinsic purpose” (*Worldwide*
20 *Church of God*, 227 F.3d at 1117). Google’s parallel use of the Java APIs is thus not
21 “transformative” as a matter of law.

22 **2. The Copyrighted Work Is Creative In Nature**

23 The second factor—“the nature of the copyrighted work”—favors creative expression.
24 Though computer programs are generally “utilitarian” and “functional,” “[t]o the extent that there
25 are many possible ways of accomplishing a given task or fulfilling a particular market demand,
26 the programmer’s choice of program structure and design may be highly creative and
27 idiosyncratic.” *Sega*, 977 F.2d at 1524.

28 Witness after witness testified to the highly creative nature of API design. Larry Ellison
testified that API design is “arguably, its one of the most difficult things we do at Oracle. . .done
by our most senior experienced and talented software engineers.” (RT 289:25-291:16.)

1 Numerous other witnesses confirmed that API design is very complex, requiring significant
2 design choices and creativity. (RT 513:12-18 (“[designing API’s] is a very creative process”);
3 513:21-514:12; 515:14-23 (Screven); 627:21-628:1 (Reinhold); 741:9-742:3; 747:5-9; 748:7-13;
4 752:5-14; 831:17-832:4 (Bloch); 1220:6-12; 1238:11-1239:12; 1240:17-20 (Mitchell); 1775:3-16
5 (Bornstein); 2209:7-8 (Astrachan) (“Q: Is it difficult to write good APIs? A: Yes.”)).

6 Google’s chief Java architect, Joshua Bloch, testified that:

7 API design is really a creative process. There are many, many kinds of decisions
8 that go into API design. Often, the people who do API design are called software
architects.

9 (RT. 1238:13-15; *see also* TX 624 at 2, 20-21, 47 (Bloch presentation entitled “How to Design a
10 Good API and Why it Matters.”).)

11 3. Google Uses Valuable, Core Portions Of Copyrighted Work

12 The third statutory factor is “the amount and substantiality of the portion used in relation
13 to the copyrighted work as a whole.” 17 U.S.C. §107(3). “This factor evaluates both the quantity
14 of the work taken and the quality and importance of the portion taken.” *Passport Video*, 349 F.3d
15 at 630.

16 In *Harper & Row*, the Supreme Court overruled the Court of Appeals and found that the
17 use of 300 words out of 200,000 from Gerald Ford’s unpublished presidential memoir was not
18 fair use, emphasizing that “[t]he portions actually quoted were selected . . . as among the most
19 powerful passages in those chapters.” 471 U.S. at 565-66. The Court also cited favorably *Roy*
20 *Export Co. Establishment v. Columbia Broad. Sys., Inc.*, 503 F. Supp. 1137, 1145 (S.D.N.Y.
21 1980), which held that a jury could reasonably have held that the defendant’s use of short excerpts
22 from Charlie Chaplin films chosen for a broadcast program about Chaplin because they were
23 among the best moments, was both “quantitatively substantial” and “qualitatively great” (503
24 F.Supp. at 1145). *Harper & Row*, 471 U.S. at 565 (“taking of 55 seconds out of 1 hour and 29-
25 minute film deemed qualitatively substantial”).

26 The Court has ruled that for purposes of fair use analysis, “the ‘work as a whole’ means
27 the contents (including name, declarations and English-language comments) of the documentation
28 for all of the 166 API packages (not just the 37) in the registered work.” (ECF No. 1018, JI 29.)

1 Thirty-seven out of 166 packages—more than 22%—is a sizeable percentage of the work as a
2 whole and significant quantitatively. This 22% represents an extraordinary amount of work. (*See*
3 RT 617:2-15 (Reinhold) (testifying that the specifications for the 37 API packages amount to
4 about 11,000 pages when printed out), 1248:11-1249:7, 2279:16-2280:6 (Mitchell) (testifying that
5 the 37 API packages included “around 400 classes” and about “5,000 methods”).)

6 Google also cannot dispute that the APIs it took were qualitatively significant. Mr.
7 Bornstein testified that these 37 API packages in particular were the most suited for smartphones
8 and the ones that Google believed developers would use and expect. (RT 1782:6-9, 1783:23-
9 1784:1 (Bornstein); 981:22-982:21 (Lee); 1331:10-15 (Mitchell).) For these reasons, the third
10 statutory factor cuts strongly against fair use.

11 **4. Google’s Use Harms The Potential Market For And Value Of** 12 **The Copyrighted Work**

13 The final fair use factor—“the effect of the use upon the potential market for or value of
14 the copyrighted work” (17 U.S.C. § 107(4))—“is undoubtedly the single most important element
15 of fair use.” *Harper & Row*, 471 U.S. at 566. “Fair use, when properly applied, is limited to
16 copying by others which does not materially impair the marketability of the work which is
17 copied.” *Id.* at 566-67, quoting 1 Nimmer § 1.10[D], at 1-87. In assessing the fourth factor, it is
18 necessary to “consider not only the extent of market harm caused by the particular actions of the
19 alleged infringer, but also ‘whether unrestricted and widespread conduct of the sort engaged in by
20 the defendant . . . would result in a substantially adverse impact on the potential market’ for the
21 original.” *Campbell*, 510 U.S. at 590.

22 The first and fourth factors are powerfully connected because, as the Supreme Court has
23 held, the proof required to demonstrate present or future market harm (the fourth factor) varies
24 with the purpose and character of the use (the first factor):

25 A challenge to a noncommercial use of a copyrighted work requires proof either
26 that the particular use is harmful, or that if it should become widespread, it would
27 adversely affect the potential market for the copyrighted work. . . . *If the intended*
28 *use is for commercial gain, that likelihood [of market harm] may be presumed.*
But if it is for a noncommercial purpose, the likelihood must be demonstrated.

1 *Sony Corp. v. Universal City Studios, Inc.*, 464 U.S. 417, 451 (1984) (cited in *A&M Records,*
2 *Inc. v. Napster, Inc.*, 239 F.3d 1004, 1016 (9th Cir. 2001). *See also Leadsinger*, at 532 (“we have
3 not hesitated to apply this presumption”); *Passport Video*, 349 F.3d at 631. As noted above in
4 discussing the first factor, it is uncontested that Google’s use of the Java APIs is purely
5 commercial in nature and has greatly enhanced Google’s very substantial Android profits.

6 Undisputed evidence at trial also established that Google’s infringing use of the Java API
7 packages substantially impaired the value of the Java Platform. Google’s Technical Program
8 Manager for Android testified that the number of Android-compatible device activations per day
9 was on average 750,000, and each of those devices has the 37 API packages from Java. (RT
10 1017:4-16 (Morrill).) Google’s Android phones compete directly with Java smart phones (such
11 as the RIM Blackberry) that were on the market prior to Android’s introduction. (RT 1922:22-25
12 (Gering); *see also* RT 2062:5-12 (McNealy) (testifying that while he was the Chairman of the
13 Board at Sun, Sun had a “very lucrative revenue stream from J2ME, which was the handset
14 version of Java.”) 2057:24-2058:14 (McNealy).)

15 Google has also significantly harmed the value of the APIs by fragmenting Java and
16 undercutting its “write once, run anywhere” capability. (TX 172 (email from Bornstein to Rubin
17 describing Android as a “fork” of Java); RT 2287:13-2288:5 (Mitchell) (testifying that beyond the
18 37 API packages, Android and Java “are different and incompatible, and the way in which things
19 are prepared to run and execute is different”); *see also* RT 984:22-24; 981:19-21 (Lee); 1010:1-7
20 (Morrill) (testifying that Android is not Java compatible).

21 Google knows this fragmentation is harmful. Google itself imposes terms requiring
22 Android users to refrain from fragmenting its APIs. (TX 749 at 8-9 (Android Compatibility
23 Definition Document).) Moreover, both Larry Page, CEO and co-founder of Google, and Tim
24 Lindholm, Google engineer and former Sun Distinguished Engineer, testified that they knew that
25 Sun was concerned about fragmentation of Java. (RT 471:6-18 (Page); 844:3-7 (Lindholm).)
26 Google seeks to use “compatibility” to justify what was actually a deliberate fragmenting of a
27 compatible ecosystem that Sun, Oracle and many other companies spent years nurturing.
28

1 No reasonable jury could find that Google has met its burden of fair use, and Oracle is
2 therefore entitled to JMOL on that defense.

3 **IV. OTHER ISSUES THAT ARE NOT BEING PRESENTED TO THE JURY**

4 **A. Oracle Is Entitled To Judgment As A Matter Of Law On**
5 **Copyrightability.**

6 Oracle understands that the Court will be determining copyrightability of the 37 API
7 packages based upon the evidence submitted at trial. At the Court's request, the parties will be
8 submitting proposed findings of fact and conclusions of law to the Court on Wednesday, May 2.
9 Oracle will set forth the evidence at trial that relates to copyrightability in greater detail in that
10 submission, which it incorporates by reference here. Nonetheless, out of an abundance of
11 caution, and because Oracle believes it is entitled to judgment on copyrightability as a matter of
12 law, Oracle addresses the issue of copyrightability here as well.

13 The evidence on copyrightability presented at trial was overwhelming. Fact and expert
14 witnesses from both sides testified that designing APIs is a creative and challenging task. (*See*
15 *RT 2209:7-8 (Astrachan); 1238:13-16 (Mitchell); 627:21-629:5 (Reinhold); 513:14-18 (Screven);*
16 *751:14-18, 830:18-19, 831:7-12 (Bloch); see also TX 624 at 47 (Bloch presentation).*) Nobody
17 testified to the contrary.

18 The API packages themselves are expressed in a detailed and complex structure, with
19 many hierarchies and interdependencies. These were illustrated in part in TX 1028, the Java API
20 package poster used by developers when programming for J2SE version 5.0. The intricate
21 structure of the API packages poster reflects only the high level class and interface relationships
22 for some of the API packages in version 5.0, because it would be impossible to fit a description of
23 all the relationships even on a large poster with tiny print. *RT 599:15-600:3 (Reinhold)*. These
24 relationships extend within and across the different packages. The types of relationships that
25 were shown at trial included the following: (1) classes can have one or more subclasses, each of
26 which inherits the methods and fields of the classes above it in the hierarchy (*RT 1225:10-16*
27 (*Mitchell*)); (2) interfaces are used to relate different classes that share common characteristics
28 among different classes located in the same, or different packages (*RT 589:13-17, 590:5-23,*

1 601:22-25 (Reinhold)); (3) methods can contain parameters that are defined in other classes
2 located within, or outside, the package in which the method is found (RT 1239:24-1240:8
3 (Mitchell)); (4) classes and subclasses can be contained within the hierarchy of one package but
4 defined in another (RT 601:14-24 (Reinhold)); (5) interfaces themselves are often arranged
5 hierarchically in a manner similar to classes (RT 1219:14-23 (Mitchell)).

6 The detailed expression of this structure cannot possibly be just an idea, as Google has
7 sometimes claimed. Nor is it driven or constrained by function as Google has also contended.
8 Dr. Reinhold testified that very little structure is required for the APIs to operate with the virtual
9 machine or computer. If function were the only concern, all of the classes could have been placed
10 in a one giant package. (RT 619:13-23). A primary purpose of the structure, sequence and
11 organization of the APIs is to make them easy to learn and easy for developers to use. (RT
12 619:24-620: 6 (Reinhold); RT 741:2-742:2 (Bloch); TX 624 at 4.)

13 Google has argued in the past that the doctrines of merger and *scenes a faire* apply to bar
14 copyrightability of APIs. The Court warned Google on summary judgment of the standard it
15 would have to meet if it hoped to establish this. The Court stated that:

16 If Google believes, for example, that a particular method declaration is a *scene a*
17 *faire* or is the only possible way to express a given function, then Google should
18 provide evidence and argument supporting its views as to that method
19 declaration. ... This approach ignores the possibility that some method
20 declarations (for example) may be subject to the merger doctrine or may be *scenes*
21 *a faire*, whereas other method declarations may be creative contributions subject to
22 copyright protection. Google has not justified the sweeping ruling it requests.
23 Google has not even identified which categories of specification elements it deems
24 unprotectable under these doctrines. This order declines to hold that API package
25 specifications, or any particular category of elements they contain, are
26 unprotectable under the merger or *scenes a faire* doctrines.

27 (ECF No. 433 at 9.) Google's evidence at trial was no better. Indeed, Google chose to not even
28 try to meet this standard; it submitted no evidence whatsoever to establish that any particular
method declaration was a *scene a faire* or was the only possible way to express a given function.

Nor could it have. It is obvious, given the complexity of the structure, that there are
countless ways to design and express the Java API packages, so the doctrine of merger does not
apply. *See Satava v. Lowry*, 323 F.3d 805, 812 n.5 (9th Cir. 2003) ("Under the merger doctrine,

1 courts will not protect a copyrighted work from infringement if the idea underlying the
2 copyrighted work can be expressed in only one way, lest there be a monopoly on the underlying
3 idea”). Dr. Reinhold testified that “In anything except the most trivial API design, there are so
4 many choices to be made that I wouldn’t even know how to start counting them.” (RT 627:21-
5 628:1 (Reinhold); *see also id.* at 628:22-629:6 (discussing design options for java.nio). Professor
6 Mitchell agreed, emphasizing that API design starts with a “clean slate.” (RT 1240:9-20.) No
7 Google witness disputed this.

8 Google also did not put on any evidence at trial to establish *scenes a faire*. “Under the
9 scenes a faire doctrine, when certain commonplace expressions are indispensable and naturally
10 associated with the treatment of a given idea, those expressions are treated like ideas and
11 therefore not protected by copyright.” *Swirsky v. Carey*, 376 F.3d 841, 850 (9th Cir. 2004).
12 Google never tried to establish this for any one of the 37 API packages, let alone all of them. The
13 evidence in the record is that APIs solving the same kinds of problems can be designed very
14 differently. Dr. Reinhold gave the example of the java.util.logging API package, which has
15 different relationships, interfaces, names and elements from a competing open source Java
16 logging API called Log4J. (RT 630:11-631:18 (Reinhold).) Dr. Mitchell discussed how data
17 collections are handled in different ways in APIs in Java, C++ and Smalltalk, and how even
18 within Java, the design of the Java.util package has changed significantly over time. (RT
19 1240:23-1244:16 (Mitchell).) In pretrial briefing, Google stated that it was “reserving the right to
20 present evidence that many aspects of the APIs are unoriginal.” (ECF No. 823 at 9.) It never did.

21 The many creative choices exercised by API designers extend not just to the structure, but
22 to the decision as to what to include in the API in the first place, and even whether to include a
23 particular API package in the library. Since the API packages are just pre-written code supplied
24 for the benefit of developers, there is no set requirement that any particular API be included. The
25 Java API packages have grown dramatically, from the seven API packages that were included in
26 the first release, to the 166 packages included with version 5.0, to 209 packages included with
27 version 7.0. (RT 631:19-25 (Reinhold).) Dr. Reinhold testified that Sun and Oracle did not have
28 to create so many Java API packages, but did so “in order to – to encourage the adoption of the

1 Java platform by adding more and more facilities to make it an attractive platform for developers
2 to use.” (*Id.* at RT 632:1-6.) Other software platforms have taken a different approach. For
3 example, C is one of the world’s most popular programming languages, but Dr. Reinhold testified
4 that “the C Standard Library is really very simple and primitive” and does not even contain basic
5 data structures.” (*Id.* at RT 632:7-20.)

6 The Court should also issue judgment as a matter of law in Oracle’s favor on the
7 copyrightability of the selection and arrangement of the names in the 37 Java APIs. The Court
8 expressly left this issue open on summary judgment. (ECF No. 433 at 8.) The evidence showed
9 that the choice of names is significant, and that API designers thoughtfully selected thousands of
10 names for aesthetic purposes and consistency. (RT 628:2-21 (Reinhold)); TX 624 (Bloch
11 presentation) at 7 (“Code should read like prose.”).) (RT 746:20-748:13 (Bloch).) The names, of
12 course, are organized within the same complex and creative structure as the API elements they
13 label, and Oracle’s selection and arrangement of those names should be found copyrightable. *See*
14 *Lamps Plus, Inc. v. Seattle Lighting Fixture Co.*, 345 F.3d 1140, 1147 (9th Cir. 2003 (“[A]
15 combination of unprotectable elements is eligible for copyright protection only if those elements
16 are numerous enough and their selection and arrangement original enough that their combination
17 constitutes an original work of authorship.”).

18 Google has contended in the past that the 37 API packages are simply a “functional
19 requirement for compatibility.” This is not the correct legal standard, but Google still failed to
20 meet it. As discussed above, the API package designs are not simply functional, and the
21 undisputed evidence at trial is that Android is not compatible with Java in any event. (*See, e.g.*,
22 RT 1007:6-11 (Morrill); TX 383 at 8; RT 1331:16-1332:2 (Mitchell).) The API packages are not
23 like a hardware interface that Google had to adopt if it wanted to use the Java programming
24 language. Google designed many of its own API packages, and the experts agreed that Google
25 could have designed its own corresponding 37 API packages if it wanted to. (RT 2212:25-
26 2213:19, 220:1-7 (Astrachan); 2288:6-13 (Mitchell).) Only about 60 classes must be present in
27 the APIs for the Java language to function, and for most of those classes there is no requirement
28 that the class contain any particular method or methods — the language simply expects that a

1 class by that name will exist. (RT 684:16-685:2 (Reinhold); 2196:1-4 (Astrachan); TX 1062; TX
2 984.)

3 Finally, as discussed in section II.D.2 above, numerous witnesses testified at the trial that
4 the structure, sequence and organization of the API packages is identical in the documentation
5 and the class libraries because both are derived from the source code. The SSO of the source
6 code falls squarely within the Ninth Circuit’s holding in *Johnson Controls*, 886 F.2d at 1175.
7 The written description of this highly expressive structure contained in the documentation is no
8 less protectable than the expression of the structure in the source code. The structure, sequence
9 and organization is copyrightable in both formats. It would be a perverse result if Google were
10 found to infringe one but not the other.

11 In sum, based on the uncontroverted evidence presented at trial, the Court can grant
12 judgment as a matter of law on copyrightability in Oracle’s favor.

13 **B. Google Infringed Oracle’s Copyrights By Deriving Its**
14 **Implementations Of The 37 API Packages From Java Specifications**

15 The Android source code implements the functions specified in the English descriptions
16 of Oracle’s copyrighted specifications, and it also incorporates the SSO of the 37 Java API
17 packages as described in the specifications. By copying the SSO *and* implementing the prose
18 descriptions of the Java API specifications, converting from English text into Java-language code,
19 Google has created an unauthorized derivative work.

20 A “derivative work” is defined as “a work based upon one or more preexisting works,
21 such as a translation . . . or any other form in which a work may be recast, transformed, or
22 adapted.” 17 U.S.C. § 101. Ninth Circuit law supports Oracle’s derivative works claim. In
23 *Micro Star v. Formgen, Inc.*, 154 F.3d 1107, 1112 (9th Cir. 1998), the seller of new levels for a
24 video game claimed it had not copied any protectable expression from the game’s creator because
25 its level files “reference the source art library, but do not actually contain any art files
26 themselves.” *Id.* at 1112. The Ninth Circuit disagreed, finding that “[i]n making this argument,
27 Micro Star misconstrues the protected work. The work that Micro Star infringes is the D/N-3D
28 story itself.” *Id.* The court thus reversed the district court’s denial of a preliminary injunction.

1 *See also Twin Peaks Prods. v. Pul'n Int'l Ltd.*, 996 F.2d 1366, 1373-74 (2nd Cir. 1993) (finding
2 detailed recounting of plot elements of television series was infringement)

3 Similarly, Google's argument that the implementing source code in Android copies only
4 unprotectable ideas from Oracle's specifications misconstrues Oracle's derivative works claim.
5 Professor Mitchell testified that the narrative in the Oracle API specifications is reflected in the
6 Android source code: "[T]he narrative is reflected in the source code because the source code is a
7 program that in a sense carries out that narrative, does what the explanation requires for this
8 method." (RT 1253:16-18 (Mitchell).) Google's expert, Dr. Astrachan, acknowledged that the
9 Android source code was "based on the specification." (RT 2219:7-18 (Astrachan).) That
10 admission tracks the very definition of a "derivative work" in the Copyright Act: "A 'derivative
11 work' is a work *based upon* one or more preexisting works" 17 U.S.C. § 101.

12 Oracle is not claiming that Google is liable for implementing any single method or even a
13 group of methods. Google is liable for implementing the entire SSO of the 37 Java API packages,
14 including the methods described therein. Even if Google were correct that the implementation
15 steps of a given method are an uncopyrightable idea—the same way that individual plot elements
16 in a film might be uncopyrightable—Google implemented Oracle's descriptions thousands of
17 times in the same structure, sequence, and organization in which Oracle wrote them. "[A] claim
18 of copyright infringement can be based on infringement of a combination of unprotected
19 elements." *Dream Games of Ariz., Inc. v. PC Onsite*, 561 F.3d 983, 988 (9th Cir. 2009). No
20 reasonable fact finder could conclude that Google did not derive the SSO, as well as the meaning,
21 of the code for the Android core libraries from Oracle's API specifications. Google's expert
22 Professor Astrachan testified that corresponding elements in Java and Android APIs are
23 structured, sequenced, and organized in the same way:

24 Q. It has to be in the same position in the application programming interface structure,
25 sequence and organization, correct?

A. That is correct.

26 (RT 2215:24-2216:2 (Astrachan).) He further testified that the method declarations in the
27 Android code are like the detailed headings in an outline:

1 Q. And the method declarations are like the sub-sub-sub-chapter headings in this
2 structure, sequence and organization; correct, sir?

3 A. I think that's one analogy that's reasonable.

4 (RT 2215:2-5 (Astrachan).) Google copied Oracle's detailed outline into the Android code.

5 Google's argument that it was required to copy to use the Java language fails. With the
6 exception of a very few classes, the Java APIs are not required to use Java at all. (RT 684:16-
7 685:2 (Reinhold).) Google could have written its own APIs that provided similar functionality, as
8 Professor Astrachan testified: "it would be possible to provide an API that performed similar
9 functionality; not maybe exactly the same but similar." (RT 2213:8-10 (Astrachan).) Google
10 actually did write many of its own APIs. (See RT 2213:17-19 (Astrachan) (noting that Android
11 developers wrote most of their own core libraries).) Google did not have to copy.

12 Nor did "compatibility" compel Google's copying. Android is not, in fact, compatible
13 with Java. (See TX 383 at 8 ("Q49. Is Android Java compatible? A. No.")) As Google engineer
14 Dan Bornstein testified, Google did not even attempt full compatibility with the Java platform:

15 Q. Did Android implement all the API packages present in any particular Java Platform?

16 A. No.

17 Q. All right. And why not?

18 A. That wasn't a goal of the project. The goal of the project was to provide something that
19 was familiar to developers. It wasn't to provide any particular preexisting set of packages.

20 (RT 1783:15-22 (Bornstein).) True compatibility with Java would have required Google to
21 incorporate all of the Java API packages, not just some. (RT 374:4-24 (Kurian).) Accordingly,
22 Google cannot use compatibility to excuse copying a select set of API packages into Android.

23 C. Google's Equitable Defenses Fail

24 Google has failed to carry its burden of establishing its equitable defenses of waiver,
25 estoppel, implied license, and laches. These defenses will be addressed in Oracle's Findings of
26 Fact and Conclusions of Law to be filed on May 2, 2012, which Oracle incorporates herein by
27 reference.

28 V. CONCLUSION

For the foregoing reasons, Oracle is entitled to judgment in its favor on its copyright
infringement claim and against Google on Google's defenses.

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28

Dated: May 1, 2012

MICHAEL A. JACOBS
MARC DAVID PETERS
DANIEL P. MUINO
MORRISON & FOERSTER LLP

By: /s/ Michael A. Jacobs
Michael A. Jacobs

Attorneys for Plaintiff
ORACLE AMERICA, INC.