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
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

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

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-03561 WHA

**GOOGLE'S MAY 24, 2012 COPYRIGHT LIABILITY TRIAL BRIEF**

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

## I. The exceptions counts in Google's May 23, 2012 brief

Due to errors in the program used, Google's May 23, 2012 Brief misstated the number of exceptions thrown by J2SE 5.0 and Android 2.2. As summarized in Exhibit A to this brief, for the 37 packages at issue, the public methods in J2SE 5.0 throw 2,400 exceptions, while the public methods for those packages in Android 2.2 ("Froyo") throw 2,316 exceptions. For the 37 packages at issue, the public methods in the two platforms throw 2,304 exceptions that are the same in the two platforms, while J2SE throws 107 exceptions that are not thrown by Android—84 of which are thrown by methods that are not implemented in Android—and Android throws 12 exceptions that are not thrown by J2SE. For most of the packages at issue, the public methods in Android and J2SE throw exactly the same exceptions.[1]

## II. Oracle's "compatibility" arguments

Oracle argues in its May 23 brief that the Android and J2SE platforms are not compatible for three reasons, but these reasons are irrelevant to determining whether the SSO of the 37 API packages is copyrightable. Specifically, Oracle argues that Android and J2SE applications are not compatible because Android applications (1) typically use a different "entry point" than J2SE applications, (2) once compiled, are in Dalvik bytecode instead of Java bytecode, and (3) once compiled, typically are stored in "apk" files instead of "jar" files.

First, Oracle's arguments proceed from an erroneous premise—that the "compatibility" analysis should be based on whether the Android platform is compatible *in all respects* with the J2SE platform. That misses the point. The purpose of implementing the Java language APIs in Android is to allow developers writing in the freely-available Java language to use the familiar, established and basic APIs that Java language developers all learn. *See* RT 762:13-23 (Bloch); RT 961:13-962:3 (Swetland); RT 1018:4-23 (Morrill); RT 1769:11-17 (Bornstein). Implementing those APIs in Android allows Java language developers to use the skills and experience they have, and ensures that they can reuse source code that they have written using the APIs in the 37 API packages. RT 2183:2-11 (Astrachan). Without these basic APIs, the Java

[1] These numbers are not identical to the numbers reported by Oracle, which may be due to slight differences in how the exceptions were counted.

language is largely useless. RT 683:14-684:4 (Reinhold); RT 782:9-14 (Bloch); RT 1477:2-13 (Schmidt); RT 1960:4-8 (Schwartz).

The relevant compatibility analysis, therefore, is whether Google's implementation of the APIs in the 37 packages is compatible with J2SE's implementation of those APIs from a technical, computer science perspective—and it is. RT 2172:6-11 (Astrachan); RT 2292:25-2293:14 (Mitchell). It is irrelevant whether Android is compatible, in its entirety, with a specific product of Oracle, with Oracle's licensing or business model, or with definitions of "compatibility" that Oracle has chosen to adopt for self-serving business reasons.[2]

Second, each of Oracle's arguments also points only to superficial distinctions that ignore that source code that uses the APIs that are common to the two platforms is interoperable with both platforms. With respect to the entry point used, Professor Astrachan explained that while J2SE encloses applications in a method called "main," Android uses a procedure that is "a little different." RT 2221:11-2222:3. But Professor Astrachan also explained that, aside from this slightly different startup procedure, "[o]therwise *nothing else would change*." RT 2221:18 (emphasis added).[3] The other two points that Oracle raised—the differences in the bytecode used and the differences between the "apk" and "jar" file formats—are relevant only to the *compiled* versions of applications. Neither of the latter two points affects whether *source code* written for the Android platform will function on the J2SE platform or whether source code written for the J2SE platform can be re-used for the Android platform—the salient question for compatibility.

Oracle further argues that these three arguments represent only the "tip of the iceberg," in that J2SE and Android applications are not fully compatible because Google did not implement all 168 of the J2SE 5.0 API packages. But as Google explained in its May 23 brief, even if source

[2] Oracle implicitly acknowledges in its May 23 brief that its "interoperability" arguments are grounded in its concerns about its "for-charge licensing model" and that it is wedded to the erroneous notion that the only compatibility that matters is the type it prefers, namely that a set of Java APIs is not "compatible" unless it is compatible with "a particular edition of Sun's Java." Dkt. 1191 at 9:4-20.

[3] Although Professor Astrachan testified that the launch point for applications in Android is "a little different," there is nothing in the record that states that Android *cannot* use "main" as an entry point for an Android application. Indeed, a "command line" Android application *can* use "main" as its entry point.

code written for one platform uses APIs not available on the other platform, the portions of the source code that rely on common APIs will run on both platforms, which means the platforms are, with respect to those APIs, compatible. *See* Dkt. 1192 at 6:5-10. Professor Mitchell conceded that this is useful. RT 2289:21-23; *see also* RT 1787:23-1788:4 (Bornstein).

Oracle's brief illogically assumes that it would for some reason be better for Android to be incompatible in every sense rather than being, as it is, compatible with the APIs in the 37 packages.[4] Jonathan Schwartz—Sun's CEO at the time that Android was launched—testified to the contrary:

> Q. And did you actually give interviews in which you said you thought Android was helping Java?
>
> A. I did. . . . .
>
> At least if they picked an Open Source Java implementation, they could be a part of the community. If they had picked something that was completely variant, it would have had no utility to us whatsoever.

RT 1992:2-12.

Indeed, Oracle has itself benefitted from the compatibility between the J2SE and Android platforms. For example, Oracle admitted that it accepted Google's contribution of Josh Bloch's TimSort.java and ComparableTimSort.java source code and incorporated it into Oracle's OpenJDK 7, which is the current Oracle release of J2SE. RT 1865:11-20 (Oracle's Resp. to Google's RFA No. 170); RT 822:10-15 (Bloch); *see also* RT 823:3 (Bloch) (testifying that Dr. Reinhold of Oracle praised the performance of TimSort when J2SE 7 was released). It is undisputed that TimSort is compatible with both the J2SE and Android platforms.

## III. *Sony Comp. Entm't, Inc. v. Connectix Corp.*

In *Sony Comp. Entm't, Inc. v. Connectix Corp.*, the Ninth Circuit held that Connectix's copying and disassembly of the Sony PlayStation BIOS—the "basic input-output system" that was the software program that operated the Sony PlayStation video game console—was a fair use

[4] Oracle's quote from Judge Whyte's decision in *Sun v. Microsoft* is inapposite. "Write once, run anywhere was never a promise that if you wrote code for one Java platform that it would automatically/magically work on another." RT 725:10-12 (Reinhold). Unlike Microsoft, Google has never claimed that Android is an implementation of J2SE. Android is a different platform than J2SE.

because Connectix's purpose in doing so was "for the purpose of gaining access to the unprotected elements of Sony's software." 203 F.3d 596, 598, 602 (9th Cir. 2000).

The Ninth Circuit's opinion does not discuss in any detail precisely what the unprotected elements were; instead, the opinion refers to the "functional" and unprotected elements of the Sony software, without identifying specifically what they are. *See, e.g., id.* at 603 ("There is no question that the Sony BIOS contains unprotected functional elements."), 605 ("If Sony wishes to obtain a lawful monopoly on the functional concepts in its software, it must satisfy the more stringent standards of the patent laws.").

In its opening Ninth Circuit appellate brief, however, Connectix explained that its Virtual Game Station software ("VGS") emulated both the Sony PlayStation hardware, and the Sony PlayStation BIOS software. In developing VGS, Connectix first emulated the PlayStation's microprocessor in software. Ex. B (Connectix Br.) at 11.[5] Connectix also studied the "interaction between Sony's BIOS and the hardware" and then "wrote software to emulate the hardware functionality." *Id.* at 11-12. The Connectix code that emulated the Sony PlayStation hardware dwarfed the Connectix code that it subsequently wrote that emulated Sony's BIOS, *see id.* at 31 n.8, much as the Android source code that implements the 37 API packages is dwarfed by the rest of the Android source code.

After Connectix had written its hardware emulation code, it "reverse engineered Sony's BIOS by running PlayStation games in conjunction with the BIOS and its software emulator of the hardware." *Id.* at 12. This was necessary because "[o]perations systems, system interface procedures, and other programs like the Sony BIOS are not visible to the user when they are operating." *Sony,* 203 F.3d at 600. Connectix then "proceeded to write code to emulate the necessary BIOS functionality." Ex. B at 13. This final step was therefore analogous to the process by which Google wrote its own code implementing the 37 API packages. Connectix "began with an empty table consistent of the entry points into the BIOS." *Id.* In order to ensure that the VGS was compatible with PlayStation games, this table had to "contain the same entry

[5] Google previously filed a copy of Connectix's opening appellate brief as Exhibit GG to the Reply Declaration of Michael S. Kwun that was filed on August 29, 2011. *See* Dkt. 369-3.

points, and be in the same order and format, as the table in Sony's BIOS." *Id.*

It appears that when the software code in PlayStation games invoked the Sony BIOS, this process included the name of the BIOS function that was being accessed. *See id.* at 13-14 ("Connectix engineers could typically deduce the requisite BIOS functionality by examining the function name, the information sent to and from the BIOS, or the general grouping of functions requested by PlayStation games."). Through a variety of means, Connectix determined the purpose, parameters and return formats for 137 of the 242 functions implemented in the Sony BIOS,[6] and then "independently wrote code to implement the required functionality." *Id.* at 13-15 & n.3. A minority of these functions ("a third to a half") were standard C programming language functions, *id.* at 13, while the rest were not. In sum, Connectix's VGS implemented the Sony PlayStation BIOS interfaces. *See* JONATHAN BAND & MASANOBU KATOH, INTERFACES ON TRIAL 2.0 at 61 (MIT Press 2011) (in *Sony*, the Ninth Circuit found that intermediate copies were fair use where "they were necessary for the uncovering of elements not protected by Sony's copyright—specifically, the BIOS's interface specifications.").

Again, this process was analogous to the process by which Google implemented the 37 API packages. Google, like Connectix, implemented some but not all of the functions from the plaintiff's software system, choosing only those functions necessary to accomplish its purpose. For those functions, Google, like Connectix, ensured that it duplicated the same "entry points" and each function's functionality, including the information sent to and from the system. Many of the Sony functions, like many of the J2SE functions, performed standard functions familiar to any programmer. Google, like Connectix, wrote its own implementing code. Indeed, the key distinction between the present case and *Sony*, is that Connectix created intermediate copies of Sony's *implementing code*, for which it had to rely on fair use. Google did not copy Oracle's implementing code, and thus section 102(b) itself precludes copyright infringement liability.

[6] Connectix implemented only 137 of the 242 functions because those were the only functions invoked by the games that Connectix tested. *Id.* at 18. This parallels Google's decision to implement some but not all of the J2SE 5.0 API packages, and most but not all of the JS2E 5.0 exceptions. Moreover, this means that VGS likely was not "fully compatible" with the Sony PlayStation—and that full compatibility is not relevant to the section 102(b) analysis.

Dated: May 24, 2012

KEKER & VAN NEST LLP

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

Attorneys for Defendant
GOOGLE INC.