DONALD F. ZIMMER, JR. (SBN 112279)
fzimmer@kslaw.com
CHERYL A. SABNIS (SBN 224323)
csabnis@kslaw.com
KING & SPALDING LLP
101 Second Street - Suite 2300
San Francisco, CA 94105
Telephone: (415) 318-1200
Facsimile: (415) 318-1300

SCOTT T. WEINGAERTNER (*Pro Hac Vice*)
sweingaertner@kslaw.com
ROBERT F. PERRY
rperry@kslaw.com
BRUCE W. BABER (*Pro Hac Vice*)
bbaber@kslaw.com
KING & SPALDING LLP
1185 Avenue of the Americas
New York, NY 10036-4003
Telephone: (212) 556-2100
Facsimile: (212) 556-2222

IAN C. BALLON (SBN 141819)
ballon@gtlaw.com
HEATHER MEEKER (SBN 172148)
meekerh@gtlaw.com
GREENBERG TRAURIG, LLP
1900 University Avenue
East Palo Alto, CA 94303
Telephone: (650) 328-8500
Facsimile: (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

Honorable Judge William Alsup

GOOGLE INC.'S OPENING CLAIM CONSTRUCTION BRIEF

TABLE OF CONTENTS

TABLE OF AUTHORITIES

CASES

STATUTES

OTHER AUTHORITIES

I. INTRODUCTION

Pursuant to Patent Local Rule 4-5 and the Court's November 29, 2010 *Order Entering Joint Proposed Claim Construction Schedule* (Dkt. 59), Defendant Google Inc. ("Google") submits this claim construction brief.

Oracle America, Inc. ("Oracle") asserts the following **7** patents and **132** claims:

The "Asserted Patents"	The "Asserted Claims"
U.S. RE38,104 ("the '104 patent")	Claims 11-41 (31 claims)
U.S. 6,910,205 ("the '205 patent")	Claims 1, 2, 3, 4, and 8 (5 claims)
U.S. 5,966,702 ("the '702 patent")	Claims 1, 5-7, 11-13, 15, and 16 (9 claims)
U.S. 6,125,447 ("the '447 patent")	Claims 1-24 (24 claims)
U.S. 6,192,476 ("the '476 patent")	Claims 1-21 (21 claims)
U.S. 6,061,520 ("the '520 patent")	Claims 1-4 and 6-23 (22 claims)
U.S. 7,426,720 ("the '720 patent")	Claims 1-8, 10-17, and 19-22 (20 claims)

The six terms selected by the parties for claim construction briefing are used in claims of six of the seven Asserted Patents. The first "term" – the various formulations of "computer-readable medium," "computer-usable medium" and "computer-readable storage medium" used in the patents – appears in claims of six of the seven Asserted Patents (i.e., all but the '205 patent). The other five terms are used in claims of the '104 patent, the '520 patent and the '702 patent. No terms unique to the '205 patent, the '447 patent, the '476 patent or the '720 patent were selected for briefing.

II. BACKGROUND AND CONTEXT

The Court's claim construction rulings may affect summary judgment issues in the following ways:

"computer-readable/usable (storage) medium": The dispute over these related phrases affects thirty asserted claims in six of the seven Asserted Patents, and would be

dispositive of these claims if the Court adopts Google's construction. Under Google's construction, which adopts the explicit definitions set forth in or incorporated by reference in the Asserted Patents, these claim terms encompass transitory data signals. The Federal Circuit has held that claims to transitory signals are invalid under 35 U.S.C. § 101. *See In re Nuijten*, 500 F.3d 1346, 1353 (Fed. Cir. 2007).

"reduced class file": This term is recited in all nine asserted claims of the '702 patent. Adoption of Google's construction would be dispositive because the Accused Instrumentalities do not "obtain a plurality of reduced class files" as required by all of the asserted claims.

"symbolic data/field reference": This term is recited in all thirty-one asserted claims of the '104 patent. Adoption of Google's construction would be dispositive of these claims because the Accused Instrumentalities do not include "instructions containing one or more symbolic references" as required by all of the asserted claims.

"intermediate form (object) code" and "resolve" / "resolving": Oracle chose these terms for construction. Although it is unclear from Oracle's discovery responses how the disputes are dispositive of any currently asserted claims of the '104 patent, Google plans to seek summary judgment of invalidity of the claims of the '104 patent. The construction of "resolve / resolving" may be dispositive on issues of infringement as well as the adequacy of the specification of the '104 patent.

"the play executing step": Oracle chose this term, which affects only 2 of the 21 asserted claims of the '520 patent. Adoption of Google's construction would be dispositive because the affected claims would be invalid under 35 U.S.C. § 112, ¶2.

III. LAW OF CLAIM CONSTRUCTION

This Court has previously summarized the basic tenets of claim construction:

> Courts must determine the meaning of disputed claim terms from the perspective of one of ordinary skill in the pertinent art at the time the patent was filed. *Chamberlain Group, Inc. v. Lear Corp.*, 516 F.3d 1331, 1335 (Fed. Cir. 2008). While claim terms "are generally given their ordinary and customary meaning," the "claims themselves provide substantial guidance as to the meaning of particular claim terms." *Phillips v. AWH Corp.*, 415 F.3d 1303, 1312, 1314 (Fed. Cir. 2005) (en banc) (quoting *Vitronics Corp. v. Conceptronic, Inc.*, 90 F.3d 1576, 1582 (Fed. Cir. 1996)). The specification of a patent is also highly relevant to claim construction. Indeed, claims "must be read in view of the specification, of

> which they are a part." *Markman v. Westview Instruments, Inc.*, 52 F.3d 967, 979 (Fed.Cir.1995) (en banc), *aff'd*, 517 U.S. 370, 116 S. Ct. 1384, 134 L. Ed. 2d 577 (1996). Finally, courts should give due consideration to a patent's prosecution history, which "can inform the meaning of the claim language by demonstrating how the inventor understood the invention and whether the inventor limited the invention in the course of prosecution, making the claim scope narrower than it would otherwise be." *Phillips*, 415 F.3d at 1318 (citations omitted). These components of the intrinsic record are the primary resources in properly construing claim terms. *Id*. at 1317-18.

MShift, Inc. v. Digital Insight Corp., No. 10-00710-WHA, 2010 U.S. Dist. LEXIS 107946 at *43-46 (N.D. Cal. Oct. 8, 2010) (Alsup, J.); *see also Sandisk Corp. v. LSI Corp.*, No. 09-02737-WHA, 2010 U.S. Dist. LEXIS 24973 at *3-6 (N.D. Cal. Mar. 17, 2010) (Alsup, J.).

"Although courts have the discretion to consider extrinsic evidence, including expert and inventor testimony, dictionaries and scientific treatises, such evidence is 'less significant than the intrinsic record in determining the legally operative meaning of claim language.'" *Netflix, Inc. v. Blockbuster, Inc.*, 477 F. Supp. 2d 1063, 1066 (N.D. Cal. 2007) (Alsup, J.) (quoting *Phillips* at 1317); *Quantum Corp. v. Riverbed Tech., Inc.*, No. 07-04161-WHA, 2008 U.S. Dist. LEXIS 116831 at *8 (N.D. Cal. Aug. 5, 2008) (Alsup, J.) (same). However, the use of the term in the art may inform one of skill in the art as to the common meaning of a term. *Laryngeal Mask Co. Ltd. v. Ambu*, 618 F.3d 1367, 1373 (Fed. Cir. 2010).

"The purpose of claim construction is to determine the *meaning* and *scope* of the patent claims asserted to be infringed." *O2 Micro Int'l Ltd. v. Beyond Innovation Tech. Co.*, 521 F.3d 1351, 1360 (Fed. Cir. 2008) (quotation omitted, emphasis added). Claim construction is often dispositive, because "most patent cases turn on the meaning of only a few phrases." November 18, 2010 Case Management Order (Dkt. 56) at ¶ 5. The *Markman* process is not, however, a mechanism for *fixing* problematic claim language. *See Chef America v. Lamb-Weston, Inc.*, 358 F.3d 1371, 1374 (Fed. Cir. 2004) ("courts may not redraft claims, whether to make them operable or to sustain their validity").

IV. DISPUTED CLAIM TERMS

A. The "Computer-Readable Medium" Terms

Although slightly different language is used in different patents, the parties agree that the terms relating to "computer readable media" – namely, "computer-readable medium,"

"computer-usable medium" and "computer-readable storage medium" – should all have the same construction. The dispute regarding the construction of these related phrases affects thirty asserted claims in six of the seven Asserted Patents, and would be dispositive of these claims if Google's construction is adopted by the Court.

1. Google's Proposed Construction

Google's proposed construction simply asks the Court to adopt the *explicit* definition of the term "computer readable medium" that appears in the majority of the patents and is consistent with these terms as used in the other patents in which they appear.

Four of the seven patents explicitly and consistently define the relevant terms to encompass both "storage devices" and other media – including transmission media such as carrier waves – that can convey information to a processor. Two of the three others contain intrinsic evidence in the form of preferred embodiments that embrace and implicitly provide the same broad definition. The remaining patent, i.e., the '104 patent, which is the earliest filed of the Asserted Patents, does not contradict the explicit and implicit definitions of the other patents, and its prosecution history demonstrates that the same meaning was intended.

Highly probative extrinsic evidence – in the form of literally hundreds of other Oracle patents that embrace virtually this same definition encompassing wave transmissions – further supports Google's construction. That evidence reflects a conscious decision by Sun Microsystems (Oracle's predecessor) to seek patent rights that encompassed both storage media and "storage devices" as well as transmission media such as carrier waves.

2. Disputed Claim Terms: "computer-readable medium" ('104, '447, '476, '520 and '720 patents), "computer usable medium" ('447, '476 and '702 patents) and "computer-readable storage medium" ('720 patent)

Google's Proposed Construction	Oracle's Proposed Construction
any medium that participates in providing instructions to a processor for execution, including but not limited to, optical or magnetic disks, dynamic memory, coaxial cables, copper wire, fiber optics, acoustic or light waves, radio-waves and infra-red data communications	a storage device for use by a computer

a. The Intrinsic Evidence and Explicit Definitions in the '447, '476 and '520 Patents

The specification of the '447 patent includes the following definition:

> The term "computer-readable medium" as used herein refers to *any medium that participates in providing instructions to processor 104 for execution*. Such a medium may take many forms, *including but not limited to*, non-volatile media, volatile media, and transmission media. Non-volatile media includes, for example, *optical or magnetic disks*, such as storage device 110. Volatile media includes *dynamic memory*, such as main memory 106. Transmission media includes *coaxial cables*, *copper wire and fiber optics*, including the wires that comprise bus 102. Transmission media can also take the form of *acoustic or light waves*, such as those generated during *radio-wave and infra-red data communications*.

'447 patent at 5:4-16 (emphasis added). Google's proposed construction (italicized in the block quote above) uses key language from this explicit definition.[1] Because this is the definition the patentee explicitly ascribed to the term, Google's construction is the only proper one. *See Markman*, 52 F.3d at 979 (holding that the specification acts as a dictionary when it expressly defines terms used in the claims); *Vitronics Corp. v. Conceptronic, Inc.*, 90 F.3d 1576, 1582 (Fed. Cir. 1996) (same). Indeed, when a disputed term "is set off by quotation marks [as it is here, it is] often a strong indication that what follows is a definition. . . . [T]he patentee ***must be bound by the express definition***." *Sinorgchem Co. v. ITC*, 511 F.3d 1132, 1136 (Fed. Cir. 2007) (emphasis added).

In addition to this explicit and broad-reaching definition, the specification further describes specific embodiments of the "computer-readable media" that plainly and unambiguously include wireless signals:

- "The received code may be executed by processor 104 as it is received, and/or stored in storage device 110, or other non-volatile storage for later execution. In this manner, computer 100 may obtain application code in the form of a carrier wave." *Id*. at 6:11-15.

See also id. at 5:32-35 (data on a telephone line), 5:52-56 (electrical, electromagnetic or optical signals), 5:67-6:4 carrier waves).

These embodiments "throw light on the meaning" of the "computer-readable medium" term, and further support Google's construction. *See Nazomi Commc'ns, Inc. v. Arm Holdings, PLC*, 403 F.3d 1364, 1369 (Fed. Cir. 2005) (holding that "trial court should also consider that

---

[1] For convenience, Google's proposed construction paraphrases the specification's express definition. However, Google would readily agree to a construction for the disputed phrases that incorporates the entire quoted paragraph.

possible embodiments of the invention may throw light on the meaning of" a disputed claim term).

Further, the express claim language confirms that the "computer-readable medium" of the '447 patent encompasses transient transmission mediums such as a carrier wave. Independent claims 10 and 16 each recite "A computer-readable medium *carrying* one or more sequences of one or more instructions . . . ." '447 patent at 14:34-35; 15:10-11 (emphasis added). This is consistent with language used in the specification to describe transmission via carrier wave. *Id*. at 5:67-6:4 ("The signals through the various networks and the signals on network link 120 and through communication interface 118, *which carry the digital data* to and from computer 100 are exemplary forms of carrier waves transporting the information." (emphasis added)).

The '476 patent and the '477 patent share the same inventor, disclose similar subject matter, were filed on the same day, and even incorporate each other by reference. *See* '447 patent at 1:11-15; '476 patent at 1:16-20. The intrinsic evidence in the '476 patent is virtually identical to the intrinsic evidence in the '477 patent, and includes the following:

- The specification provides a definition of "computer-readable medium" that is *identical* to the definition found in the '447 patent. '476 patent at 5:4-16.
- Like the '447 patent, the '476 patent describes many embodiments that utilize various forms of wireless signals. *See id.* at 5:32-35 (same as in the '447 patent at 5:32-35); *id*. at 5:52-56 (same as in the '447 patent at 5:52-56); *id*. at 5:67-6:4 (same as in the '447 patent at 5:67-6:4); *id*. at 6:14-18 (same as in the '447 patent at 6:11-15).
- Independent claims 10 and 15 of the '476 patent claim "A computer readable medium *carrying* one or more sequences of one or more instructions," thus evidencing that a carrier wave is within the scope of the claims. *Id.* at 19:25-26, 20:6-7 (emphasis added).

Finally, the '520 patent provides a similar definition of "computer-readable medium" that explicitly includes carrier waves. *See* '520 patent at 4:48-56 (defining "computer-readable media" to include "such as secondary storage devices like hard disks, floppy disks, or CD-Rom; *a carrier wave received from the Internet 204*; or other forms of RAM or ROM" (emphasis added)). According to this definition, any construction of phrases like "computer-readable media" in the context of this patent must necessarily encompass carrier waves. *See Sinorgchem*, 511 F.3d at 1136 (definitional language binds the patentee).

b. The Intrinsic Evidence in the '720, '702 and '205 Patents

The '720 patent includes, by express incorporation, a definition that is even more broadly encompassing of carrier waves than the construction Google has offered. The '720 patent incorporates by reference the "commonly-assigned U.S. patent application Ser. No. 09/970,661, filed Oct. 5, 2001." '720 patent at 3:4-6. This incorporated patent application, subsequently issued to Oracle as U.S. Patent No. 7,213,240 ("the '240 patent"), expressly states that "[a]lthough aspects of the present invention are described as being stored in memory, one skilled in the art will appreciate that these aspects can also be stored on or read from other types of computer-readable media, such as secondary storage devices, like hard disks, floppy disks, or CD-ROM; *a carrier wave, optical signal or digital signal from a network, such as the Internet*; or other forms of RAM or ROM either currently known or later developed." Ex. A, '240 patent at 13:24-32 (emphasis added). Disclosures of a patent or patent application incorporated by reference should be viewed as – and are intended by the patentee as – the equivalent of the disclosures of the patent itself. *See Arlington Indus. v. Bridgeport Fittings, Inc.*, No. 2010-1025, 2011 U.S. App. LEXIS 1118 at *20; 97 U.S.P.Q.2d (BNA) 1811 (Fed. Cir. Jan. 20, 2011) (adopting a construction conforming with another patent incorporated by reference).

Other evidence intrinsic to the '720 patent confirms the broad definition of "computer-readable media" (and similar terms) to encompass carrier waves. The only relevant disclosure of the '720 patent specification is to storage device 19, storage device 15, and storage device 35:

> [A] server 18 is operatively coupled to a storage device 19 in which globally shareable class libraries 20 are maintained. Each of the heterogeneous devices 11 can interface via a network 12, which includes conventional *hardwired and wireless network configurations*. Other types of heterogeneous devices 11 and various network configurations, arrangements, and topologies are possible.

'720 patent at 4:40-46 (emphasis added); *see also id.* at figs. 1-2. Accordingly, "computer-readable storage medium" must be construed to include transient signals such as the incorporated and disclosed carrier wave. *See Nazomi Commc'ns*, 403 F.3d at 1369 (disclosed embodiments throw light on the meaning of a disputed claim term).

The '702 patent similarly encompasses carrier waves in the context of computer readable media in a manner that is wholly consistent with and fully supports Google's proposed construction based on the patentees' own explicit definition of the term:

- "The received code may be executed by CPU 213 as it is received, and/or stored in mass storage 212, or other non-volatile storage for later execution. *In this manner, computer 200 may obtain application code in the form of a carrier wave*." *Id*. at 7:10-14 (emphasis added).

See also id. at 6:48-52 (electrical, electromagnetic or optical signals); 6:60-67 (carrier waves). These disclosed embodiments in the '702 patent further emphasize that the term "computer-readable medium" properly includes transient signals such as the disclosed carrier waves. *See Nazomi Commc'ns*, 403 F.3d at 1369 (disclosed embodiments throw light on the meaning of a disputed claim term).

The '205 patent was filed in the same time frame as many of the other Asserted Patents, is also consistent with Google's proposed construction, and provides further evidence of the patentees' intention to draft claims to encompass software transmitted via the internet or otherwise via a carrier wave. In particular, a "carrier wave" is explicitly included among the embodiments disclosed in the '205 patent specification. *See* '205 patent at 4:48-54 ("Although the CD-ROM 15 is shown as an exemplary computer readable storage medium, other computer readable storage media including floppy disk, tape, flash memory, system memory, and hard drive may be utilized. Additionally, a data signal embodied in a *carrier wave* (e.g., in a network including the Internet) may be the computer readable storage medium." (emphasis added).

c. The Intrinsic Evidence in the '104 Patent.

The '104 patent also contains intrinsic evidence consistent with Google's proposed construction that is based on the repeated explicit definition of "computer readable media" encompassing waves. In fact, the intrinsic evidence reveals that Sun intentionally sought the perceived benefit of this definition when it ***added*** the language "computer-readable medium" to the claims of the '104 patent for the first time nearly four years after the original application was filed. The original application for the '104 patent, filed in 1992, did not include the term

"computer-readable medium" at all. It was not until 1996 – in an attempt to broaden the '104 patent – that Sun used the reissue procedure to add claims reciting this term.

In view of the above, Google's proposed construction should be adopted. A complete summary of the prosecution and reissue history of the '104 patent is attached as Exhibit F. As that summary shows, disclosures of a "computer-readable medium" in connection with the '104 patent were not submitted to the patent office until the first reissue application was filed on November 21, ***1996***. Because the term "computer-readable medium" was never disclosed in the claims or specification of the original 1992 application, the correct timeframe for interpreting this phrase in accordance with one of ordinary skill in the art would start no earlier than 1996, when this phrase was first introduced into the related application.[2] That later time frame, of course, is when Sun filed and prosecuted the applications discussed above, which contain the explicit definitions described above.

d. Extrinsic Evidence

Moreover, in the relevant 1996-97 timeframe – when "computer-readable medium" was defined and claimed by the inventors of the Asserted Patents – one of ordinary skill in the art would consider that phrase to broadly include carrier waves. As discussed above, at least four of the Asserted Patents explicitly defined this term to include a carrier wave. *See* discussions of intrinsic evidence in the '447, '476, '520, and '205 patents, *supra*. Sun's (now Oracle's) patent portfolio contains literally hundreds of instances in which Oracle represented to the Patent Office and to the public that its claims addressed to "computer readable medium" were explicitly defined to encompass information carried by waves.

On July 1, 1996, for example, Sun filed an application, which issued as U.S. Patent No. 5,953,522, wherein claim 15 recites: "The computer program product of claim 14 wherein the computer readable medium is a data signal embodied in a carrier wave."

---

[2] *See Reiffin v. Microsoft Corp.*, 214 F.3d 1342, 1346 (Fed. Cir. 2000) ("[C]laims to subject matter in a later-filed application not supported by an ancestor application in terms of § 112 ¶1 are not invalidated [for that reason]; they simply do not receive the benefit of the earlier application's filing date.").

Similarly, on December 12, 1997, Sun filed an application that issued as U.S. Patent No. 5,946,489 and discloses that:

> [c]ommon forms of computer-readable media include, for example, . . . *a carrier wave* as described below. . . . The network 86 uses electrical, *electromagnetic or optical signals that carry digital data streams*. The signals through the various networks and the signals on the network link 84 and through the communication interface 82, which carry the digital data to and from the computer system 20, are *exemplary forms of carrier waves transporting the information*. . . . The received code may be executed by the processor 22 as it is received and/or stored in the storage device 80, or other non-volatile storage for subsequent execution. In this manner, *the computer system 20 may obtain application code in the form of a carrier wave*.

U.S. Patent No. 5,946,489 at 11:1-64 (emphasis added).

Numerous other Oracle patents from the same time frame support Google's construction. *See* Ex. C (list of Oracle patents directed to computer readable media and carrier waves). Accordingly, compelling extrinsic evidence shows that, at the time the Asserted Patents and these other patents were filed, the term "computer readable medium" was widely recognized as a legal term of art that included wireless transmission. *Cf. PIN/NIP v. Platte Chem.*, 304 F.3d 1235 (Fed. Cir. 2002) (noting that the basic definition of a term was "well-established" and "well-expressed" in other patents and thus "applicable to [the present] case.").

B. The '702 Patent

1. Overview

The '702 patent purports to optimize the data storage scheme inherent in Oracle's Java platform. While typical applications for Windows or Macintosh computers are stored as a small number of relatively large files, Java applications are typically stored as a large number of relatively small "class files," each containing data and instructions related to a particular "class" of objects.[3]

The well-known object-oriented programming concept of a "class" is described generally in the background section of the patent and elsewhere. '702 patent at 2:11-61. The "class file" described in the '702 patent is a digital file on a computer with a very specific file format, as

[3] One commentator alluded to this difference by comparing the ability for certain tools to compress the relatively large executable files used on other systems with the large number of relatively small files created by Java compilers. Ex. E, Horspool at 1254.

illustrated in figure 3 of the '702 patent (reproduced below) and described at great detail starting at column 11 and ending at column 35.

As depicted and described, a class file is defined in the '702 patent as a specifically-formatted file containing the definition of exactly one class or interface.[4] *Id.* at 7:31-32, 7:20-9:7, 11:23-35:42, fig. 3. A class file may be structured in a standard-compliant Java class file format containing such structures as a "constant pool table," "fields," and "methods," which are defined with great precision in the patent. *Id.* at 7:20-9:7, 11:23-35:42, fig. 3. Class files contain data such as constant pool entries, including *names* by which the class file references other class files, etc. *Id.* at fig. 3, 8:9-13. Class files also contain instructions, organized as methods. *Id.* at fig. 3, 8:52-56. This detailed specification of the class file is necessary so that the Java Virtual Machine (which executes Java applications) can properly interpret the contents of each class file.

Another aspect of the Java platform design is the requirement that a Java application reference all data and functionality by name. Thus, if a Java program needs to print a message for the user using a method named printMessage, the Java program must include the name of that method, "printMessage," in a table of such names. This table is called the "constant pool table." *Id.* at fig. 5, numeral 305. Often, many different class files will each contain a copy of that string constant, "printMessage." This results in a duplication of data across the class files for a given Java application. *Id.* at 1:39-40.

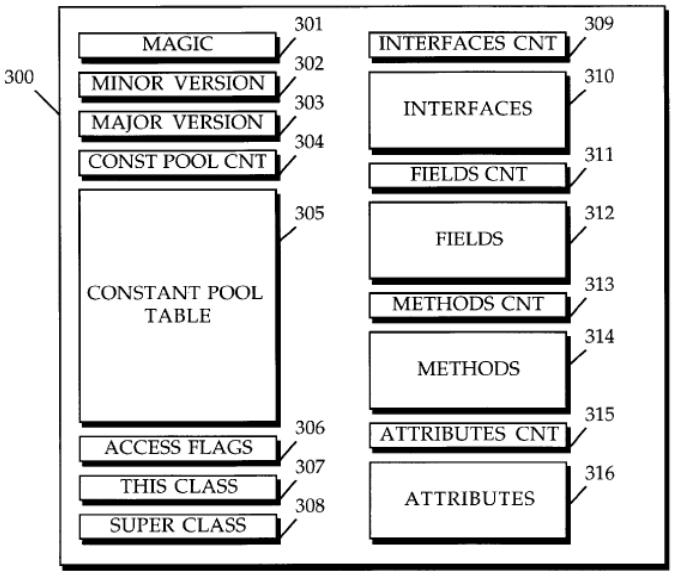

FIGURE 3

[4] An "interface" is a mechanism for generically discussing classes without talking about any specific class. By analogy, one might talk about wanting to buy a car, but one cannot actually purchase a "generic" car on a car lot – only specific models of cars. One can talk about a car generically, and people understand that it is a motor vehicle that has three or more wheels, doors, seats, a trunk, etc. A "car" is analogous to an interface, while a Chevy Volt is analogous to a class that may be said to implement the "car" interface.

The claims of the '702 patent are directed to reducing the duplication of information across a plurality of class files, such as those in a Java class file format. *Id.* at 1:29-40. The claims of the '702 patent recite *removing duplicated elements* in a plurality of class files, thus creating a plurality of *reduced class files*, and forming a *shared table* comprising the duplicated elements. The reduced class files and the shared table are then combined to form a *multi-class file*.

One embodiment described in the '702 patent is a pre-processor that examines each class file in a set of class files to identify duplicate information such as redundant constants. *Id.* at 9:17-21. All occurrences of the redundant constant are removed from the constant pools of the individual class files, which therefore become *reduced* class files. *Id*. at 9:23-25, 9:39-40. The redundant constants are collected and placed in a shared table. *Id.* at 9:21-23. The pre-processor determines memory allocation requirements for the class files and packages together the (1) shared table, (2) memory allocation requirements, and (3) reduced class files to form what is called a *multi-class* file. *Id.* at 5:12-17, 9:35-40.

2. Disputed Claim Term: "reduced class file"

Google's Proposed Construction	Oracle's Proposed Construction
a class file containing a subset of the data and instructions contained in a corresponding original class file	Oracle's position is that no construction is necessary, but if the Court agrees that Construction is necessary, the parties are essentially in agreement. A "reduced class file" contains a subset of the code and data contained in a class file

Contrary to Oracle's assertions, the parties' proposals differ in important aspects that require this Court's construction. Google's proposed construction properly focuses on the unambiguous claim language in view of the specification, and makes clear that a "reduced class file" is, by its own terms, *a class file*. Claim 1 recites "removing said duplicated elements from said plurality of class files to obtain a plurality of reduced class files." In other words, the original class files are reduced when duplicated elements are removed, but are still of the class file format. *See also* '702 patent at claim 7 ("computer readable program code configured to cause a computer *to remove said duplicated elements* from said plurality of class files *to obtain a plurality of reduced class files*" (emphasis added)). *See Phillips*, 415 F.3d at 1314 ("claims

themselves provide substantial guidance as to the meaning of particular claim terms"). The removal of duplicated elements from a class file does not change that class file into something other than a class file. It is still a class file – albeit a "reduced" one. *E.g.*, '702 patent at 5:9-17, fig. 5.

This is further supported by the specification – particularly the "Summary of the Invention" – which explains that "all occurrences of the [duplicated] constant are removed from the respective constant pools of the individual class files," and that "[t]he shared table, the memory allocation requirements and *the reduced class files* are packaged as a unit in a multi-class file." *Id.* at 5:9-17 (emphasis added). When the Java virtual machine processes the multi-class file, it loads the "individual classes, with respective reduced constant pools" from the multi-class file (*Id.* at 5:23-27, 9:35-43), confirming that the classes are stored as individual class files that are "reduced" versions of the original class files.

In view of the claim language and the clear language of the specification, Google believes it is important that the Court's construction of "reduced class file" reflect: (1) that the "reduced class file," by its own terms, *must be a class file;* (2) that the relevant elements of the class files at issue are "data" and operations (i.e., "instructions") (*e.g.*, '702 patent, fig. 3, numerals 305 and 314); and (3) that the reduced class file must be a subset of *the original class file* that has been reduced by the claimed process. Google also believes it is important that the construction not include unnecessary ambiguities such as references to the undefined term "code," which does not appear in the discussion of "reduced class files" in the intrinsic evidence.

C. The '104 Patent

1. Overview

The '104 patent discloses two different approaches to converting a software program written in a human-readable programming language ("source code") into a form that can run on a computer. In the first "compiled" approach, a program called a compiler converts the source code into "executable code for a specific computer architecture." '104 patent at 1:25-28. The resulting code, often called "machine code" or "native code" is in a form that is specific to the computer hardware (e.g., a microprocessor or CPU) so that the hardware knows how to run the

software program. In the second "interpreted" approach, a translator first converts the source code "of a program into an intermediate form, typically independent of any computer instruction set." *Id.* at 1:58-61. Subsequently, a program called an interpreter "scans through the code in intermediate form, and performs the indicated actions." *Id.* at 1:67-2:3. In other words, the interpreter essentially converts (or "interprets") the intermediate form code to machine code at run time so that the hardware knows how to run the software program.

Intermediate form code may take the form of a sequence of instructions that may contain references to data stored in a data object elsewhere in memory. *Id.* at 1:31-43, 1:61-6, figs. 1A-1B. Generally, a data reference may be numeric as depicted in Figure 1A (i.e., the number "2" in the "LOAD 2" instruction), or symbolic as depicted in Figure 1B (i.e., the string or character "y" in the "LOAD 'y'" instruction). According to the '104 patent, data references in instructions in intermediate form "are made on a symbolic basis" and, thus, are not "not fully resolved." *Id.* 1:61-64. Accordingly, the interpreter must resolve a symbolic reference "each time the instruction comprising the symbolic reference is interpreted." *Id.* at 2:3-6.

The '104 patent describes this resolution as "an extra level of interpretation at execution time," which causes execution to slow each time instructions comprising symbolic references are interpreted. *Id.* at 2:10-15. Accordingly, the '104 patent discloses a "hybrid compiler-interpreter" that resolves symbolic references only once during the first pass of the program execution. *Id.* at 5:39-49.

2. Disputed Claim Term: "symbolic [data/field] reference"

Google's Proposed Construction	Oracle's Proposed Construction
a dynamic reference to data that is string- or character-based	No construction necessary. The ordinary meaning is "a reference by name"

This claim term is important because the specification repeatedly distinguishes between references that are "symbolic" and references that are "numeric." Oracle's construction has no support in the specification, and also fails to make clear this important distinction. On the other hand, Google's proposed construction is fully supported by the intrinsic evidence, and directly addresses this issue.

Google's construction relies on the specification to determine the meaning of the word "symbolic." "Symbolic" was repeatedly distinguished from "numeric" in the specification, and Google's construction makes this distinction clear by identifying what references are – and what references are not – "symbolic" within the meaning of the '104 patent. Indeed, the '104 patent clearly and consistently recites the distinction between two different mechanisms by which program instructions may reference data values: static numeric references and dynamic symbolic references. For example, in the "Summary of the Invention," the '104 patent states that there are "two data reference handling routines, a static field reference routine for handling numeric references and a *dynamic* field reference routine for handling *symbolic references*. The dynamic field reference routine resolves a symbolic reference and rewrites the symbolic reference into a numeric reference." *Id.* at 2:41-47 (emphasis added). In another example, "[a]s shown in FIG. 7, upon receiving a data reference byte code, block 86, the main interpretation routine determines if the data reference is static, i.e. numeric, or *dynamic, i.e. symbolic*, block 88." *Id.* at 5:9-12 (emphasis added).

In the lone example provided in the '104 specification, a program references a data coordinate point, which may represent the coordinates of a point on a two-dimensional graph, i.e., an (x, y) coordinate. *Id.* at 1:32-41; *see also* Figs. 1A and 1B *below*.

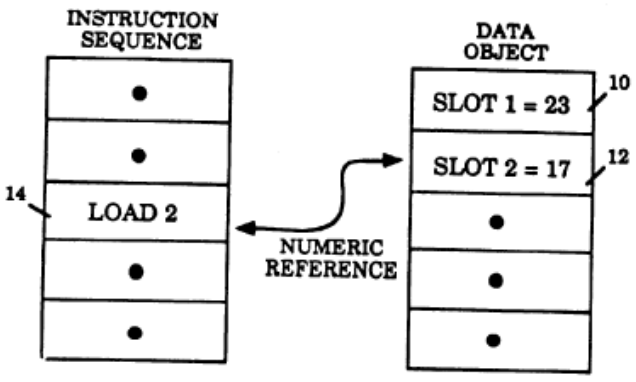

Figure 1A
Prior Art

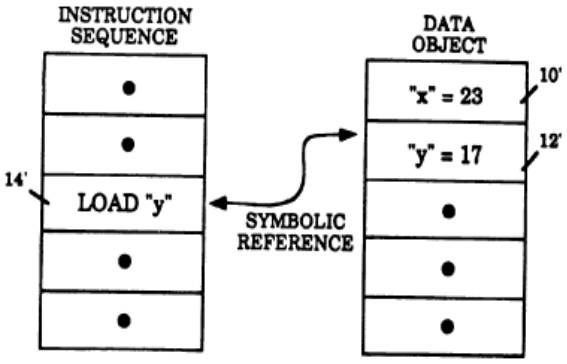

Figure 1B
Prior Art

In this example, the data values representing the coordinate pair may be referred to symbolically by the single character strings "x" and "y" (*see* Fig. 1B) or directly using the numeric addresses, or slot numbers, 1 and 2 (*see* Fig. 1A). Symbolic referencing is

advantageous as it allows the program to correctly access the desired data values even if those values are at some point moved in memory or reassigned to different slots. *Id.* at 1:51-56. This advantage must be balanced against the additional processing required to locate the data in its numeric slot when the program is executed. *Id.* at 2:10-15.

Thus, it is clear from the intrinsic evidence that a "symbolic reference" is a dynamic reference to data that is string- or character-based.

3. Disputed Claim Term: "intermediate form (object) code"

Google's Proposed Construction	Oracle's Proposed Construction
code that is generated by compiling source code and is independent of any computer instruction set	executable code that is generated by compiling source code and is independent of any computer instruction set

The parties have nearly reached agreement on the construction of the term "intermediate form (object) code," with the only dispute being whether the term requires the limitation that the code *must* be executable, as Oracle proposes. The intrinsic evidence firmly establishes that this term should not be so limited. The Court can therefore accept Google's proposed claim construction and focus its attention on the propriety of Oracle's suggestion that "executable" should be included in the construction as a further limitation on "code."

Nothing in the intrinsic evidence supports reading the term "executable" as a further narrowing of the term "code." The specification contains no such definition. Further, nothing in the specification amounts to a clear disavowal of claim scope that would narrow the term's ordinary meaning. Rather, the specification explicitly states that "a variety of well known tokens, intermediate representations, annotations, *and intermediate forms* may also be used to practice the present invention." '104 patent at 4:29-32 (emphasis added).

It is a bedrock principle of patent law that "the claims of a patent define the invention to which the patentee is entitled the right to exclude," and that the words of the claim "are generally given their ordinary and customary meaning." *Phillips v. AWH Corp.*, 415 F.3d at1312. The term "intermediate form (object) code," as used in the '104 patent claims, is never modified by the word "executable." *See* '104 patent claims 11, 12, 17, 19-23, 27-35, and 39-41. Thus, the explicit claim language does not require the "executable" limitation proposed by Oracle.

The claims and prosecution history both reinforce this conclusion. The original claims of the '685 patent (before reissue) explicitly recited "*executable code* in intermediate form." *See* '104 patent at 5:63 (original claim 1) (emphasis added); *id.* at 6:36-37 (original claim 6). Upon reissue, however, the patentee decided to drop the "executable" modifier, and claim only "intermediate form (object) code." The absence of this modifier strongly implies that the term "intermediate form (object) code" is not limited to "executable" code. *Cf. Phillips*, 415 F.3d at 1314 (finding that the use of the term "steel baffles" "strongly implies that the term 'baffles' does not inherently mean objects made of steel"). Indeed, when claims use different terms, those differences are presumed to reflect a difference in the scope of the claims. *Bd. of Regents v. BenQ Am. Corp.*, 533 F.3d 1362, 1371 (Fed. Cir. 2008).

Accordingly, this Court should adopt Google's construction of "intermediate form (object) code," which does not improperly import a limitation that the code be "executable." *See, e.g., Ventana Med. Sys., Inc. v. Biogenex Labs., Inc.*, 473 F.3d 1173, 1181 (Fed. Cir. 2006) (claims are not limited to the disclosed embodiment and need not cover every feature disclosed in the specification).

4. Disputed Claim Term: "resolve" / "resolving"

Google's Proposed Construction	Oracle's Proposed Construction
replace/replacing at least for the life of the process	No construction necessary. "Resolving" a symbolic reference is determining its corresponding numerical reference.

a. Overview.

The vast majority of the '104 specification discloses admittedly well-known techniques of the prior art. The sole allegedly distinguishing feature of the '104 patent is the manner in which the claimed system resolves symbolic references. Indeed, the '104 patent admits that prior art interpreters had long been capable of resolving a symbolic reference in intermediate code at run time. *See, e.g.*, '104 patent at 2:3-9. However, the '104 patent claims that the prior art was inefficient because resolution of a symbolic reference was performed *each time the instruction was interpreted*. Accordingly, the '104 patent purports to resolve symbolic references so that an instruction containing a symbolic reference need only be resolved *at the first execution of the*

instruction. In other words, when a process (or program) is run by an interpreter, the symbolic reference is replaced during resolution with a more efficient numeric reference, after which the numeric reference is used. This replacement must remain at least for the duration of execution – i.e., the life – of the process that is being interpreted in order to satisfy the stated objective of the '104 patent and to distinguish the '104 patent from the prior art disclosed therein. *See Nystrom v. Trex Co.*, 424 F.3d 1136, 1144-45 (Fed. Cir. 2005) (construing term "board" as limited to wood cut from log because the intrinsic evidence consistently described "board" as such); *3M v. J&J Orthopaedics, Inc.*, 976 F.2d 1559, 1566 (Fed. Cir. 1992) (holding that the lower court was "entirely proper" to use the specification—including the stated objective of the patent—to construe disputed claim terms).

b. The Term Should be Construed

Oracle's position that this phrase does not need to be construed is without merit because – as evidenced by the proposed constructions – the "plain meaning" of this term is disputed by the parties. The word "resolve" has non-technical meanings as well as many possible technical meanings. Construction is therefore necessary. *See Conceptus, Inc. v. Hologic, Inc.*, No. 09-02280-WHA, 2010 U.S. Dist. LEXIS 24247 at *19 (N.D. Cal. Mar. 16, 2010) (Alsup, J.) (citing *O2 Micro*, 521 F.3d at 1361 ("A determination that a claim term 'needs no construction' or has the 'plain and ordinary meaning' may be inadequate when a term has more than one 'ordinary' meaning or when reliance on a term's 'ordinary' meaning does not resolve the parties' dispute.")). Moreover, when presented with a term having multiple ordinary meanings – such as "resolve" – the court should adopt a meaning that includes the disclosed examples in the specification. *Conceptus*, 2010 U.S. Dist. LEXIS 24247 at *26 (citing *Verizon Servs. Corp. v. Vonage Holdings Corp.*, 503 F.3d 1295, 1305 (Fed. Cir. 2007). Finally, the claims and specification show that the patentee applied a particular technical meaning for the term "resolve" that needs to be reflected in the definition of this term. *See Edwards Lifesciences LLC v. Cook Inc.*, 582 F.3d 1322, 1329 (Fed. Cir. 2009)(holding that court should adopt special meaning "if the intrinsic evidence shows that the patentee distinguished that term from prior art on the basis

of a particular embodiment . . . ."); *3M*, 976 F.2d at 1566 (holding that it is "entirely proper" to use the stated objective of the patent to construe disputed claim terms).

The sparse disclosure of the '104 patent relating to the term "resolve" -- namely Figure 8 and its description ('104 patent at 3:26-27) -- tie "resolution and rewriting" together as two sides of the same coin. That Figure, the only one in the patent relating to this disputed term, shows a single step labeled "symbolic reference rewritten as numeric reference." This is not surprising, since this characterization is consistent with the stated objective of the patent, the remainder of the specification, and the purported distinction over the admitted prior art. Based on this limited disclosure, Google's proposed construction is correct.

c. The Specification

The '104 patent concedes that prior art interpreters such as the BASIC interpreter had long been able to resolve a symbolic reference by replacing it with a numeric reference at run time. '104 patent at 2:3-9. However, the '104 patent alleges that these prior art interpreters experience "execution [that] is slowed significantly" due to the "extra level of interpretation at execution time" (*i.e.*, resolving the symbolic reference) that must be performed "each time an instruction comprising a symbolic reference is interpreted." *Id.* at 1:67-2:15.

In contrast, the '104 patent discloses that a symbolic reference be resolved only once while a process is being executed. The "Summary of the Invention" states that the dynamic field reference routine "resolves a symbolic reference and rewrites the symbolic reference into a numeric reference . . . thereby allowing the rewritten instruction with numeric reference to be reexecuted." *Id*. at 2:44-51. As a result, the disclosed system was said to "achieve[] execution performance substantially similar to that of the traditional compiled object code . . . since data reference resolution is performed at the *first execution* of a generated instruction . . . ." *Id*. at 2:60-67 (emphasis added).

Similarly, the "Detailed Description" states unequivocally that resolution of a symbolic reference is performed only once while a process is being executed "under the present invention":

> ***Thus, except for the first execution, the extra level of interpretation to resolve the symbolic reference is no longer necessary.*** Therefore, under the present

> invention, the "compiled" intermediate form object code of a program achieves execution performance substantially similar to that of the traditional compiled object code, and yet it has the flexibility of not having to be recompiled when the data objects it deals with are altered like that of the traditional translated code, ***since data reference resolution is performed at the first execution of a generated instruction comprising a symbolic reference.***

Id. at 5:39-49 (underline and italics added); *see also* 5:32-38, fig. 8. *See Trading Techs. Int'l v. eSpeed*, 595 F.3d 1340, 1353 (Fed. Cir. 2010) (a reference to "'present invention' strongly suggests" that the invention is so limited).

The '104 patent discloses only this embodiment and does not suggest elsewhere that the claimed "resolution" can be practiced without replacing the subject of the resolution (i.e., the symbolic reference) at least for the duration of execution, i.e., the life of the process.

d. The Prosecution History

Google's definition also reflects the patentee's express statements in the prosecution history, where he explained that "[w]hen an unresolved symbolic reference is encountered, a numerical value corresponding to the reference is determined and stored in memory. When a resolved symbolic reference is encountered, the instruction is interpreted by reading the stored numeric value." Ex. B-3, '104 File History at OAGOOGLE0000059397–98[5] (in the context of claim 11 but referring to the patent more generally).)

It is therefore evident from the patent claims, figures, specification, prosecution history, and prior art that "resolving" requires (i) *replacing* a symbolic reference by rewriting it as a numeric reference, such that (ii) the numeric reference is used *at least for the life of the process*, so that the symbolic resolution is performed only "for the first execution" of an instruction containing a symbolic reference. '104 patent at 5:39-40. These distinctions are essential to the patent, its only disclosed embodiment, and the stated reason for the improved execution performance claimed by the patent. *See id*. at 2:60-67, 5:41-49. *See 3M*, 976 F.2d at 1566 (holding that the lower court was "entirely proper" to use the specification – including the stated objective of the patent – to construe disputed claim terms).

---

[5] Future citations to the file histories will be shortened to the last five digits of the document production number.

e. The Claim Language

The claims of the '104 patent further confirm that "resolve" and "resolving" mean "replace/replacing at least for the life of the process." Claim 12, for example, indicates that the patentee intended that claim for "resolving" to encompass and not be distinct from storing a numerical reference. *See* '104 patent at claim 12; *see also* claims 13 and 18-19. Other claims, such as claim 11, provide that a numeric reference is *determined* for use instead of the symbolic reference, and that it is *stored* for future use. *See id*. at claim 11. Thus, the claim language is consistent with the specification's apparent attempt to distinguish the admitted prior art and supports Google's proffered construction. *See Applera*, 2008 U.S. Dist. LEXIS 16712 at *15 ("While certain terms may be at the center of the claim construction debate, the context of the surrounding words of the claim also must be considered in determining the ordinary and customary meaning of those terms." *ACTV, Inc. v. Walt Disney Co.*, 346 F.3d 1082, 1088 (Fed. Cir. 2003)); *see also Hockerson-Halberstadt, Inc. v. Converse Inc.*, 183 F.3d 1369, 1374 (Fed. Cir. 1999) ('Proper claim construction, however, demands interpretation of the entire claim in context, not a single element in isolation')").

D. The '520 Patent

1. Overview

The '520 patent is directed at a way of improving the performance of a virtual machine. A virtual machine is a program that runs on particular hardware and that can run other programs not written for that hardware. '520 patent at 1:9-17. Virtual machines long predate the '520 patent. The '520 patent is directed to a method of receiving computer code, simulating the execution of computer code without actually running the code in order to identify the target operation, and then creating a single instruction allowing the processing component to perform the target operation. *Id.* at claim 6. This "optimized" single instruction is later processed by the virtual machine. *Id.* In this same manner, the '520 patent seeks to improve conventional systems for initialization of static arrays by reducing the amount of code executed by the virtual machine to statically initialize an array. *See id.* at Abstract; *see also id*. at claim 1.

2. Disputed Claim Term: "the play executing step"

Google's Proposed Construction	Oracle's Proposed Construction
Indefinite – cannot be construed	"The play executing step" in claims 3 and 4 is a reference to the "simulating execution" step in claim 1

The phrase "the play executing step" as used in claims 3 and 4 of the '520 patent is indefinite because it lacks proper antecedent basis. This deficiency is insoluble and renders the claims indefinite. *See Exxon Research & Eng'g Co. v. United* States, 265 F.3d 1371, 1375 (Fed. Cir. 2001).

This deficiency – which affects only two claims out of the 132 claims asserted by Oracle – is, moreover, a claim drafting error that cannot be corrected or "fixed" by way of claim construction. As this Court has previously held, the proper way to remedy a claim drafting error is through proceedings before the Patent Office. *See Applera Corp.-Applied Biosystems Group v. Illumina, Inc.*, No. 07-cv-02845-WHA, 2008 U.S. Dist. LEXIS 16712 at *15 (N.D. Cal. February 21, 2008) (Alsup, J.); *see also Helmsderfer v. Bobrick Washroom Equip., Inc.*, 527 F.3d 1379, 1383 (Fed. Cir. 2008) ("Courts cannot rewrite claim language."). Even if this phrase could be construed, moreover, Oracle's proposed construction is contrary to the plain language of the claims and improperly attempts to regain claim scope that was surrendered during prosecution of the patent.

a. The Prosecution History of the '520 Patent

The application that led to the '520 patent was filed with original claim 1 reciting:

> 1. A method in a data processing system for statically initializing an array, comprising the steps of:
>
> compiling source code … ;
>
> receiving the class file … ;
>
> play executing the byte codes of the clinit method against a memory to identify the static initialization of the array by the preloader;
>
> storing into an output file … ; and
>
> interpreting the instruction … .

Ex. B-4, '520 File History at 57480. This claim was rejected as anticipated by the prior art publication "Briki: An Optimizing Java Compiler," by Cierniak. *See* Ex. B-4, '520 File History at 57505-06. In response to the rejection, the applicants amended claim 1 as follows:

> **simulating execution of [play executing]** the byte codes of the clinit method against a memory **without executing the byte codes** to identify the static initialization of the array by the preloader;

Ex. B-4, '520 File History at 57510-11 (underlining shows additions to claim; bracketing shows deletions from claim). Applicants stated that during an interview with the Examiner, "applicants' attorney and the Examiner agreed [that the] amendment to claim 1 . . . would further clarify the distinctions between the claim and the cited art" and that "this amendment rendered all of the pending claims allowable over the cited art." Ex. B-4, '520 File History at 57511.

Dependent claims 3 and 4 refer to "The play executing step" of claim 1. These two claims were not amended when claim 1 was amended. In response to the amendment of claim 1, the Examiner allowed all of the claims. Ex. B-4, '520 File History at 57513.

Because "[t]he prosecution history limits the interpretation of claim terms so as to exclude any interpretation that was disclaimed during prosecution," *Southwall Techs., Inc. v. Cardinal IG Co.*, 54 F.3d 1570, 1576 (Fed. Cir. 1995), it is clear that the phrase "play executing is not the same as "simulating execution." "Claims may not be construed one way in order to obtain their allowance and in a different way against accused infringers." *Id.* Claim 1 originally recited "play executing the byte codes" – but that language was rejected under 35 U.S.C. § 102 as anticipated by the prior art. *See* Ex. B-4, '520 File History at 57505–06. To secure the patent, the applicants amended claim 1, changing "play executing" to "simulating execution of" and adding the phrase "without executing the byte codes" to this limitation. Ex. B-4, '520 File History at 57510. The Examiner believed that this amendment was sufficient to narrow the claim and avoid the prior art. *See* Ex. B-4, '520 File History at 57511 ("Narrowing the claims to make clear the functions of these elements were also discussed."). And because no argument was made at the time of the amendment that the claim scope was unchanged, the patentability-based changes to the claim are limiting. "Where no explanation is established, . . . the court should presume that the patent applicant had a substantial reason related to patentability for

including the limiting element added by amendment." *Warner-Jenkinson Co. v. Hilton Davis Chem. Co.*, 520 U.S. 17, 33 (1997).

As a result, the applicants of the '520 patent surrendered any claim that the "play execution" step is synonymous with, or equivalent in scope to, the "simulating execution" step.

b. Judicial Correction of this Claim Drafting Error is Improper

The Federal Circuit has established a two-part test to determine whether "a district court can act to correct an error in a patent by interpretation of the patent." *Novo Indus., L.P. v. Micro Molds Corp.*, 350 F.3d 1348, 1354 (Fed. Cir. 2003). A district court may do so if (1) the prosecution history does not "suggest a different interpretation of the claims," and (2) the correction is not "subject to reasonable debate based on consideration of the claim language and the specification." *Id.* Failure to meet just one prong of the test is futile to the party seeking to correct a claim. *See id.* The proposed "correction" offered by Oracle fails to meet either prong.

First, as shown above, the prosecution history suggests a different interpretation of the language and demonstrates that "play executing" is not the same as "simulating execution" and the "simulating execution" step is different in scope from the originally-claimed "play executing" step.

Second, the plain language of the claims contradicts the correction that Oracle now seeks. Claim 1 recites "simulating execution of the byte codes of the clinit method against a memory *without executing the byte codes*." '520 patent at 9:34-36 (emphasis added). Claim 3, on the other hand, recites "the play executing step includes . . . reading a byte code from the clinit method that manipulates the stack; *and performing the stack manipulation on the allocated stack*." *Id*. at 9:66-10:4 (emphasis added). Similarly, claim 4 recites "the play executing step includes . . . reading a byte code from the clinit method that manipulates local variables of the clinit method; *and performing the manipulation of the local variables on the allocated variables*." *Id*. at 10:5-11 (emphasis added). The "simulating execution" step of claim 1 *prohibits* execution of the byte codes, whereas claims 3 and 4 indicate that the "play executing" step *requires* execution of the byte codes in order to perform the stack or variable manipulations.

As a result, the propriety of the proposed "correction" of claims 3 and 4 to simply equate "play execution" with "simulating execution" is "subject to reasonable debate based on consideration of the claim language." *Novo Indus.*, 350 F.3d at 1354 (holding that court cannot correct an error in a patent through interpretation where correction is subject to reasonable debate). This Court should refrain from correcting an error that Oracle should have sought to address in the Patent Office. *See id.*

As this Court has held, the claim construction process should not be misused for redrafting invalid claims:

> [T]he inventor, patent counsel, and the examiner all made a drafting error. While it is tempting to just fix it up in the claim construction process, that temptation would be dangerous course, for it should be up to the PTO in the first instance to amend claims.. . . It may be that, once redrafted, the examiner might recognize prior art problems that escaped attention before. The express language of the claim must govern.

Applera, 2008 U.S. Dist. LEXIS 16712 at *15.

V. CONCLUSION

In view of the above, Google respectfully asks the Court to enter an order adopting Google's proposed constructions for each of the disputed terms discussed above.

DATED: March 17, 2011

KING & SPALDING LLP

By: /s/ Scott T. Weingaertner

SCOTT T. WEINGAERTNER (*Pro Hac Vice*)
sweingaertner@kslaw.com
ROBERT F. PERRY
rperry@kslaw.com
BRUCE W. BABER *(Pro Hac Vice)*
bbaber@kslaw.com

ATTORNEYS FOR DEFENDANT GOOGLE INC.