

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

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

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

17 *Attorneys for Plaintiff*
 ORACLE AMERICA, INC.

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

22 ORACLE AMERICA, INC.

23 Plaintiff,

24 v.

25 GOOGLE INC.

26 Defendant.

Case No. CV 10-03561 WHA

**ORACLE'S OPENING CLAIM
 CONSTRUCTION BRIEF**

Dept.: Courtroom 9, 19th Floor
 Judge: Honorable William H. Alsup

Tutorial: April 6, 2011, 1:30 p.m.

Hearing: April 20, 2011, 1:30 p.m.

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

TABLE OF CONTENTS

	Page
INTRODUCTION	1
LEGAL STANDARD	3
ARGUMENT	4
I. The Play Executing Step ('520 Patent)	4
II. Intermediate Form Code ('104 Patent) Intermediate Form Object Code ('104 Patent)	7
III. Resolve / Resolving ('104 Patent)	9
IV. Symbolic Reference ('104 Patent)	12
V. Reduced Class File ('702 Patent)	14
VI. Computer-Readable Medium ('104 Patent) Computer-Readable Storage Medium ('720 Patent) Computer-Readable Medium ('520 Patent) Computer Usable Medium ('702 Patent) Computer-Readable Medium ('447 Patent) Computer-Readable Medium ('476 Patent)	16
A. '104 patent	18
B. '720 patent	20
C. '520 patent	21
D. '702 patent	21
E. '447 and '476 patents	22

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

TABLE OF AUTHORITIES

Page(s)

CASES

Agilent Techs., Inc. v. Affymetrix, Inc.,
567 F.3d 1366 (Fed. Cir. 2009)..... 11

Astra Aktiebolag v. Andrx Pharms., Inc.,
222 F. Supp. 2d 423 (S.D.N.Y. 2002)..... 24

Ecolab, Inc. v. FMC Corp.,
569 F.3d 1335 (Fed. Cir. 2009)..... 25

Energizer Holdings, Inc. v. Int’l Trade Comm’n,
435 F.3d 1366 (Fed. Cir. 2006)..... 6

Festo Corp. v. Shoketsu Kinzoku Kogyo Kabushiki Co.,
535 U.S. 722 (2002)..... 25

In re Beauregard,
53 F.3d 1583 (Fed. Cir. 1995)..... 16

In re Lowry,
32 F.3d 1579 (Fed. Cir. 1994)..... 18

In re Nuijten,
500 F.3d 1346 (Fed. Cir. 2007)..... 18

Kathrein-Werke KG v. Radiacion y Microondas S.A.,
No. 07 C 2921, 2010 WL 2011939 (N.D. Ill. May 17, 2010)..... 5

Nystrom v. Trex Co.,
424 F.3d 1136 (Fed. Cir. 2005)..... 3

Phillips v. AWH Corp.,
415 F.3d 1303 (Fed. Cir. 2005) (*en banc*)..... passim

Rhine v. Casio, Inc.,
183 F.3d 1342 (Fed Cir. 1999)..... 24

Trading Techs. Int’l, Inc. v. eSpeed, Inc.,
595 F.3d 1340 (Fed. Cir. 2010)..... 25

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

TABLE OF AUTHORITIES
(continued)

Page(s)

OTHER AUTHORITIES

EXAMINATION GUIDELINES FOR COMPUTER-RELATED INVENTIONS (1996).....	23
THE IEEE STANDARD DICTIONARY OF ELECTRICAL AND ELECTRONICS TERMS (6th ed. 1996)	24
THE ILLUSTRATED DICTIONARY OF ELECTRONICS (7th ed. 1997)	24
MANUAL OF PATENT EXAMINING PROCEDURE § 2106.01 (8th ed. 6th rev. 2007)	16
MICROSOFT PRESS COMPUTER DICTIONARY (2d ed. 1994)	10, 13
MICROSOFT PRESS COMPUTER DICTIONARY (3d ed. 1997)	24
WEBSTER’S NEW WORLD DICTIONARY OF COMPUTER TERMS (5th ed. 1994).....	13

INTRODUCTION

1
2 The technology covered by Oracle's seven patents relates to the Java Platform.¹ Java was
3 originally developed by Sun Microsystems, which was acquired by Oracle in January 2010 and
4 was renamed Oracle America, Inc. In the early to mid-1990s, Java rose to popularity with the rise
5 of the Internet and the World-Wide Web. It was designed to achieve "**write once, run**
6 **anywhere**" in a new, networked computing environment that included the Internet.

7 The basic idea behind Java is that a software developer writes application source code
8 once, compiles it into an intermediate form known as Java "**bytecode**", and distributes the
9 bytecode in the form of "**class files**" or "**jar files**." A user wanting to run the application can get
10 a copy of the application bytecode through a variety of mechanisms, including by downloading it
11 from the Internet. The user runs the code using a Java "**virtual machine**" that was written for the
12 user's particular computer architecture. The same application code can be run on any computer
13 that has a Java Virtual Machine. This is unlike other computer languages like C or C++, which
14 are compiled to a particular processor's instruction set, and so must be separately compiled for
15 each target platform. Consistency among each platform's Java virtual machine implementation is
16 achieved because each conforms to the Java Virtual Machine Specification so it will run the Java
17 bytecode as expected.

18 Because of its strengths, Java quickly became one of the most popular software platforms.
19 Oracle estimates that the Java Platform has attracted more than 6.5 million software developers,
20 and that more than 1.1 billion desktop computers and 3 billion mobile phones run Java. *See*
21 <http://www.java.com/en/about/> (last visited Mar. 17, 2011).

22 Oracle distributes the various components of the Java Platform, and permits others to do
23 so, under a variety of licensing terms. There are commercial, royalty-bearing Java licenses, and

24 ¹ We use "Java Platform" to distinguish from the Java *programming language*. Core pieces of the
25 Java Platform include:

- 26 • The Java programming language
- 27 • The Java Runtime Environment (including Java Virtual Machine)
- 28 • Extensive Java Class Libraries and APIs (application programming interfaces)
- The Java Development Kit (collection of programming tools, including a Java compiler)

1 there are Java licenses that are royalty-free (e.g., OpenJDK code² is available under the GNU
2 General Public License version 2). But not even the “free” versions of Java are licensed without
3 important restrictions. The various licenses covering Java share one common goal: protecting
4 compatibility. As the Java Community Process website explains: “A marketplace flooded with
5 proprietary code that varies from the official specification, or worse, parallel platform versions,
6 would ‘fork’ the platform, eliminating its basic ‘Write Once, Run Anywhere’ compatibility
7 foundation.” http://jcp.org/ja/press/news/licensing_update (last visited Mar. 17, 2011). Oracle,
8 and Sun before it, has gone to great lengths to prevent forks—*Sun v. Microsoft* probably being the
9 most famous example. The Java licensing model is an important part of that effort.

10 As Sun software engineers were developing the Java Platform, they faced many technical
11 challenges. By today’s standard, the desktops of the ’90s were constrained computing
12 environments, and in order to maximize performance, Sun engineers developed many innovative
13 techniques to increase execution speed while using memory more efficiently. Furthermore, since
14 Java was developed with the Internet in mind, security (e.g., protecting user data from being
15 stolen and misused by malicious software) was a primary concern and an active area for Sun
16 innovation.

17 Sun’s Java Platform development efforts to date have resulted in about two thousand
18 patents. The seven patents asserted in this case can be categorized as follows with respect to the
19 computing environment constraints discussed above:

20 Improving execution speed: RE38,104; 6,910,205; 7,426,720

21 Efficient memory utilization: 5,966,702; 6,061,520; 7,426,720

22 Improved security mechanisms: 6,125,447; 6,192,476

23 The Court ordered the parties to select no more than six phrases to brief and argue at the
24 claim construction hearing. Oracle selected two and Google selected four.

25 Oracle selected “the play executing step” from the ’520 patent because it is potentially
26 outcome-determinative for the claims in which it appears. Oracle contends that the meaning of

27 _____
28 ² See <http://openjdk.java.net/faq/> and <http://openjdk.java.net/legal/gplv2+ce.html>.

1 the phrase is readily determined from the intrinsic record; Google contends the phrase is not
2 amenable to construction and renders the claims indefinite because it lacks antecedent basis.

3 Oracle selected “intermediate form code”/“intermediate form object code” from the ’104
4 patent because the construction of that term will significantly streamline the validity issues with
5 respect to that patent. Google’s overbroad construction is designed to capture the disparaged
6 prior art disclosed in the specification as part of the background to the invention; Oracle’s
7 proposal will exclude most, if not all, of Google’s asserted prior art references.

8 Google selected “symbolic (data/field) reference” and “resolve”/“resolving” from the ’104
9 patent, and selected “reduced class file” from the ’702 patent. Oracle believes that Google’s
10 proposed constructions for these terms are intended to support motions for summary judgment of
11 non-infringement, although how precisely those motions might be framed we cannot say, as
12 Google has not explained the grounds for any such motions in its interrogatory responses.

13 Google selected “computer-readable medium”/“computer usable medium”/“computer-
14 readable storage medium” for construction. These phrases appear in six of the seven patents.
15 Google’s proposed construction for these terms are intended to support a motion for summary
16 judgment of invalidity under 35 U.S.C. § 101 for claiming nonstatutory subject matter.

17 **LEGAL STANDARD**

18 This Court is well aware of applicable claim construction law. Claim construction is a
19 question of law for the Court. *Nystrom v. Trex Co.*, 424 F.3d 1136, 1141 (Fed. Cir. 2005). A
20 claim term is to be given “the meaning it would have to a person of ordinary skill in the art at the
21 time of the invention.” *Phillips v. AWH Corp.*, 415 F.3d 1303, 1313 (Fed. Cir. 2005) (*en banc*).³
22 Disputed terms are to be interpreted in light of the claims themselves, the patent specification, and
23 the prosecution history. *See id.* at 1312-14. While extrinsic evidence “may be useful to the court,
24 . . . it is unlikely to result in a reliable interpretation of patent claim scope unless considered in the
25 context of the intrinsic evidence.” *Id.* at 1319.

26 _____
27 ³ Unless otherwise indicated, all citations and internal quotations have been omitted, and all
28 emphasis has been added. Patents are referred to by their last three numbers, and citations to
patent specifications are in the form “patent, column:line”, such as “’104, 2:60-62”.

ARGUMENT

I. THE PLAY EXECUTING STEP ('520 PATENT)

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

The phrase "the play executing step" appears in the preambles of asserted dependent claims 3 and 4 in the '520 patent. Each begins: "The method of claim 1 wherein the play executing step includes the steps of" Google argues that the claim is indefinite because "the play executing step" has no antecedent basis. But in fact it does—it is the "simulating execution" step:

1. A method in a data processing system for statically initializing an array, comprising the steps of:

compiling source code containing the array with static values to generate a class file with a clinit method containing byte codes to statically initialize the array to the static values;

receiving the class file into a preloader;

simulating execution of 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;

storing into an output file an instruction requesting the static initialization of the array; and

interpreting the instruction by a virtual machine to perform the static initialization of the array.

3. The method of claim 1 **wherein the play executing step includes the steps of:**

allocating a stack;

reading a byte code from the clinit method that manipulates the stack; and

performing the stack manipulation on the allocated stack.

'520, 9:47-10:4. Claims 4 and 5 have the same preamble as claim 3.

1 The intrinsic evidence makes clear that the “play executing step” in claim 3 is a reference
 2 to the “simulating execution” step of claim 1. In the ’520 patent, “simulating execution” and
 3 “play executing” are synonymous. Indeed, in the “Summary of the Invention” section, “play
 4 executes” is defined as “simulates executing”: “the preloader identifies all <clinit> methods and
 5 simulates executing (‘play executes’) these methods to determine the static initialization
 6 performed by them.” ’520, 2:65-3:1. In the detailed description, the reverse can be found:
 7 “When play executing, discussed below, the preloader simulates execution of the byte codes
 8 contained in the <clinit> method by the virtual machine.” ’520, 4:64-66. The specification uses
 9 the phrases interchangeably to refer to the same act:

’520, 4:36-41	’520, 6:23-27
Processing consistent with the present invention is performed by the preloader 220 searching for a <clinit> method, and when it is found, the preloader (1) simulates execution of the <clinit> method to determine the effects it would have on memory if it was interpreted by the virtual machine 222	The preloader receives as a parameter a method information data structure that defines the <clinit> method, described in the Java™ Virtual Machine Specification at pp. 104-106, and play executes the byte codes of this <clinit> method.

16 Given this disclosure, there can be no doubt that “play executing” in claim 3 refers to
 17 “simulating execution” in claim 1. See *Kathrein-Werke KG v. Radiacion y Microondas S.A.*, No.
 18 07 C 2921, 2010 WL 2011939, at *5 (N.D. Ill. May 17, 2010) (construing “stripline sections”,
 19 “stripline segments”, and “stripline elements” to be the same, because the intrinsic evidence
 20 showed patentee “obstinately” used the terms to mean the same thing).

21 The prosecution history supports this construction. The patent examiner also viewed
 22 “play execution” and “simulated execution” as being synonymous. In the examiner’s summary of
 23 an October 13, 1999 interview, he wrote: “The general background of the invention was
 24 discussed. In particular, the disclosure of the ‘pre-loader’ and ‘simulation’ function as disclosed
 25 in claims 1, 6, and 12.” ’520 file history, 10/13/1999 Examiner Interview Summary (Declaration
 26 of Marc David Peters in Support of Oracle’s Opening Claim Construction Brief (“Peters Decl.”)
 27 Ex. 1). On that date, Claims 1, 6, and 12 all recited “play executing,” which the examiner
 28 referred to as “simulation.”

1 The reason that claim 1 says “simulates execution” and claim 3 says “play executing” is
2 because of an amendment to claim 1 that the examiner agreed would “further clarify the
3 distinctions between the claim and the cited art.” ’520 file history, 10/18/1999 Amendment at 2
4 (Peters Decl. Ex. 2). That amendment changed the claim as follows:

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

8 *Id.* The point of the amendment was to include in claim 1 the concept of “without executing the
9 code” that was present in other claims that had already been allowed by the examiner (e.g., claim
10 6: “play executing the code without running the code”; claim 18: “simulating execution of the
11 code without running the code”). When the amendment to claim 1 was made, no corresponding
12 amendment to claims 3, 4, or 5 was made, so they still recited “wherein the play executing step.”
13 But of course the understanding of which step in claim 1 was the “play executing” step did not
14 change—it was the same step it was before.

15 The meaning of the claim is reasonably ascertainable, particularly given the way in which
16 “play execution” and “simulated execution” were used interchangeably both in the specification
17 and in the prosecution history. The use of the phrase “wherein the play executing step” does not
18 render the claim insolubly ambiguous or incapable of construction. “When the meaning of the
19 claim would reasonably be understood by persons of ordinary skill when read in light of the
20 specification, the claim is not subject to invalidity upon departure from the protocol of
21 ‘antecedent basis.’” *Energizer Holdings, Inc. v. Int’l Trade Comm’n*, 435 F.3d 1366, 1370-71
22 (Fed. Cir. 2006) (holding that claim was not indefinite because “anode gel” provided the
23 antecedent basis for “said zinc anode”).
24
25
26
27
28

1 **II. INTERMEDIATE FORM CODE ('104 PATENT)**
 2 **INTERMEDIATE FORM OBJECT CODE ('104 PATENT)**

Claim Phrase	Oracle's Proposed Construction	Google's Proposed Construction
<p>3 intermediate form code 4 intermediate form object code</p>	<p>5 executable code that is 6 generated by compiling 7 source code and is 8 independent of any computer 9 instruction set</p>	<p>10 code that is generated by 11 compiling source code and 12 is independent of any 13 computer instruction set</p>

14 The phrase “intermediate form object code” appears in asserted claims 11, 12, 17, 22, and
 15 23; and “intermediate form code” appears in asserted claims 19, 20, 21, 27, 28, 29, 30, 31, 32, 33,
 16 34, 35, 39, 40, and 41. In the '104 patent, “intermediate form object code” and “intermediate
 17 form code” are synonymous; that is not in dispute. Nor do the parties dispute that “intermediate
 18 form code” is the output of a compiler or that it need not be tied to any particular computer
 19 architecture or instruction set. The fundamental dispute is whether each phrase refers to code that
 20 is executable. It does.

21 Plain and ordinary meaning controls this dispute. *See Phillips*, 415 F.3d at 1312-13. As
 22 recited in the claims, each phrase means executable code that is generated by compiling source
 23 code and is independent of any computer instruction set. For one example, claim 11 recites:

- 24 11. An apparatus comprising:
- 25 a memory containing **intermediate form object code constituted by a set of instructions**, certain of said instructions containing one or more symbolic references; and
- 26 **a processor configured to execute said instructions** containing one or more symbolic references by determining a numerical reference corresponding to said symbolic reference, storing said numerical references, and obtaining data in accordance to said numerical references.

27 According to claim 11, “intermediate form object code” is a set of instructions that a processor is
 28 configured to execute. *Phillips*, 415 F.3d at 1314 (“To begin with, the context in which a term is
 used in the asserted claim can be highly instructive.”). In other words, “intermediate form object
 code” is executable. Every other claim in which “intermediate form code” or “intermediate form
 object code” appears consistently refers to it as executable: claim 12 (“interpreting said

1 instructions in accordance with a program execution control”); claim 17 (“a method for executing
2 said program”); claim 19 (“memory for use in executing a program”); claim 20 (“computer-
3 implemented method for executing a compiled program containing instructions in an intermediate
4 form code”); claim 21 (“such that when the program is executed”); claim 22 (“a processor
5 configured to execute the instruction”); claim 23 (“a method for interpreting a compiled program
6 in intermediate form object code”); and claims 27-35, 39, 40, and 41 (“execute the program” or
7 “executing the program”). The qualifier of “intermediate form” applied to “code” and “object
8 code” is to indicate that the code is not dependent on the instruction set of a specific type of
9 machine, as is the typical case for compiled object code. *Compare* ’104, 1:25-29 (“In a compiled
10 programming language, a computer program (called a compiler) compiles the source program and
11 generates executable code for a specific computer architecture.”) *with* ’104, 1:58-61 (“In an
12 interpreted language, a computer program (called a translator) translates the source statements of
13 a program into some intermediate form, typically independent of any computer instruction set.”).

14 The specification supports the conclusion that intermediate form code is executable. The
15 Summary of the Invention explains that the claimed invention covers a “method and apparatus for
16 generating *executable code* and resolving data references in the generated code” ’104, 2:25-
17 29. The specification consistently reinforces that “intermediate form [object] code” is executable:

18 Therefore, under the present invention, the “compiled” intermediate form object
19 code of a program achieves *execution performance* substantially similar to that of
20 the traditional compiled object code, and yet it has the flexibility of not having to
21 be recompiled when the data objects it deals with are altered like that of the
22 traditional translated code, since data reference resolution is performed at the first
23 execution of a generated instruction comprising a symbolic reference.

24 ’104, 5:41-49.

25 Google’s proposed construction is inconsistent with the claims and the specification.
26 Google seeks to embrace what the ’104 specification calls an “intermediate representation,”
27 which is an internal state of a compiler rather than executable code output by a compiler. *See*,
28 *e.g.*, ’104, 4:12-32 & Figs. 4-5. The specification explains that “intermediate representation” and
“intermediate form code” are not the same:

Shown in FIG. 4 is one embodiment of the compiler 42 comprising a lexical
analyzer and parser 44, an intermediate representation builder 46, a semantic

1 analyzer 48, and a code generator 50. These elements are sequentially coupled to
2 each other. Together, they transform program source code 52 into tokenized
3 statements 54, intermediate representations 56, annotated intermediate
4 representations 58, and *ultimately intermediate form code 60* with data references
5 made on a symbolic basis.

6 '104, 4:15-23. Figures 4 and 5 show that source code transformed to an intermediate
7 representation as part of the compilation process is later output as intermediate form object code
8 after it goes through a code generator. The claims reinforce the conclusion that an “intermediate
9 representation” is different from “intermediate form code”. Claim 21 recites “intermediate form
10 code containing symbolic field references associated with an intermediate representation of
11 source code for the program, the intermediate representation having been generated by lexically
12 analyzing the source code and parsing output of said lexical analysis.” Unsurprisingly, this
13 corresponds to the description of Figure 4 reproduced above.

14 By leaving out the requirement that intermediate form code be executable, Google’s
15 proposal is an effort to broaden the '104 claims to include within the scope of “intermediate form
16 code” the intermediate representations employed by prior art compilers. This broadening is not
17 supported by the specification, file history, or the claim language, and should be rejected.

18 **III. RESOLVE / RESOLVING ('104 PATENT)**

Claim Phrase	Oracle’s Proposed Construction	Google’s Proposed Construction
Resolve / Resolving	No construction necessary. “Resolving” a symbolic reference is determining its corresponding numerical reference.	replace/replacing at least for the life of the process

19 The term “resolve” or “resolving” appears in a majority of the asserted claims of the '104
20 patent. “Resolution” is the determination of the location that corresponds to a given name. The
21 plain language of the claims provides the meaning of “resolving” such that no construction is
22 necessary. Claim 20,⁴ for example, states: “resolving the symbolic reference in the instruction by
23 determining a numerical reference corresponding to the symbolic reference.” The idea behind
24

25
26
27 ⁴ The text of claims 20, 21, and 23 appears in the April 29, 2003 certificate of correction to the
28 '104 patent.

1 “resolution” is that, although computers store and access data by location (e.g., memory locations
2 or addresses), people like to refer to data by name (e.g., x, y, name) when writing code, so it can
3 be better understood. For a computer to run a program, it needs to “resolve” the names (the
4 symbolic references) in the program to determine the corresponding locations (the numeric
5 references). This process is reflected in a dictionary definition that Google offered in support of
6 “symbolic reference”: “Symbolic address: A memory address that can be referred to in a program
7 by name rather than by number. The interpreter, compiler, or assembler translates the name into
8 the number that specifies the address.” MICROSOFT PRESS COMPUTER DICTIONARY 379 (2d ed.
9 1994) (Peters Decl. Ex. 3); *see also* Joint Claim Construction Statement (“JCCS”) (Dkt. 91) at 9.
10 The translation referred to in the definition is a form of resolution.⁵

11 If the Court does construe this term, the Court should adopt Oracle’s proposed
12 construction because it is compatible with all the specification’s uses of the word in connection
13 with different types of programming languages, and it provides meaning to all relevant claim
14 terms. Google’s proposed construction, by contrast, is contrary to both the claims and the
15 specification.

16 The different ways that the specification uses the word “resolve” or “resolution” shows
17 that a symbolic reference is resolved by determining a corresponding numerical reference.
18 *Phillips*, 415 F.3d at 1313 (claim terms must be read “not only in the context of the particular
19 claim in which the disputed term appears, but in the context of the entire patent, including the
20 specification.”). According to the ’104 specification, in some programming languages, a
21 compiler resolves references when it generates executable code, and before it is executed:

22 In a compiled programming language, a computer program (called a compiler)
23 compiles the source program and generates executable code References to
24 data in the generated code are *resolved prior to execution* based on the layout of
the data objects that the program deals with, thereby, allowing the executable code
to reference data by their locations.

25 ’104, 1:25-32. Thus a compiler may resolve references even for code that never comes to “life”
26 by being executed. So Google’s proposed construction that “resolving” is for the life of a

27 _____
28 ⁵ The converse is not true, however. Resolution does not require translation.

1 particular process is contradicted by the specification.

2 The specification also discloses that, in some interpreted programming languages, “[e]ach
3 of the symbolic references is resolved during execution *each time* the instruction comprising the
4 symbolic reference is interpreted.” ’104, 2:3-6. This means the resolved symbolic reference is not
5 *replaced*—otherwise it would not need to be resolved again. So Google’s proposed construction
6 that “resolving” means “replacing” is contradicted by the specification.

7 Oracle’s proposed construction encompasses determining the numerical reference
8 corresponding to a symbolic reference, regardless of when or how often resolution occurs. Thus
9 it fits equally well with compiled languages, interpreted languages, and languages that share
10 characteristics of both, as with the ’104 invention. ’104, 2:16-24.

11 In contrast, Google’s proposed construction should be rejected because it would render
12 claim language superfluous. *Agilent Techs., Inc. v. Affymetrix, Inc.*, 567 F.3d 1366, 1378 (Fed.
13 Cir. 2009) (“A claim construction that gives meaning to all the terms of the claim is preferred
14 over one that does not do so.”). In several instances, the claims and the specification include
15 “resolve” in close proximity to “replace” or to synonyms of replace, indicating that “resolve” and
16 “replace” have separate meanings. For example, claim 30 recites: “*replacing* each instruction in
17 the program with a symbolic data reference with a new instruction containing a numeric reference
18 *resulting from* invocation of a . . . *routine to resolve* the symbolic data reference.” Claims 31 and
19 32 follow an almost identical pattern. The Summary of the Invention, moreover, describes the
20 acts of resolving and rewriting separately within the same sentence: “The . . . routine . . . *resolves*
21 a symbolic reference and *rewrites* the symbolic reference into a numeric reference.” ’104, 2:44-
22 47. In yet another example, the original claim 1 of the 1992 application—part of the patent
23 specification—recites, within a single step (c), “*resolving* said symbolic references to
24 corresponding numeric references, *replacing* said symbolic references with their corresponding
25 numeric references, and continuing interpretation.” U.S. Patent App. No. 07/994,655 Original
26 Claims at 14 (Peters Decl. Ex. 4); *see also* ’104, claim 1. Had the inventor intended resolving and
27 replacing to be synonymous, roughly half of the language quoted above would be unnecessary.
28

1 This analysis is further supported by the doctrine of claim differentiation, in which “the
 2 presence of a dependent claim that adds a particular limitation gives rise to a presumption that the
 3 limitation in question is not present in the independent claim.” *Phillips*, 415 F.3d at 1315. A
 4 construction that equated “resolving” and “replacing” would render claim 14 identical in scope to
 5 claim 13, from which it depends. Claim 13 includes the limitation “resolving a symbolic
 6 reference in an instruction, said step of *resolving* . . . including the substeps of: determining a
 7 numerical reference corresponding to said symbolic reference, and storing said numerical
 8 reference in a memory.” Claim 14 adds the limitation: “wherein said substep of storing said
 9 numerical reference comprises the substep of *replacing* said symbolic reference with said
 10 numerical reference.” If “resolving” meant “replacing,” as Google proposes, claim 14 would be
 11 completely superfluous. Such a construction must be rejected. *Id.* at 1324-25.

12 **IV. SYMBOLIC REFERENCE (’104 PATENT)**

13

Claim Phrase	Oracle’s Proposed Construction	Google’s Proposed Construction
symbolic (data/field) reference(s)	No construction necessary. The ordinary meaning is “a reference by name”	a dynamic reference to data that is string- or character-based

14

15

16

17 Variants of the phrase “symbolic reference” appear in each asserted claim of the ’104
 18 patent. The plain and ordinary meaning of “symbolic reference,” “symbolic data reference,” and
 19 “symbolic field reference” is a reference to something by name, so it need not be construed. The
 20 ’104 patent contrasts a situation in which “references to data are made on a symbolic basis,” such
 21 as when an instruction “references the variable y by the symbolic name ‘y’”, with a situation in
 22 which those references are resolved and code can “reference data by their locations.” ’104, 1:31,
 23 1:64-67. The ’104 patent specification has three examples of symbolic references (x, y, and name
 24 (’104, 1:51-54 & Fig. 1B)), but these examples are not limiting. *Phillips*, 415 F.3d at 1323; ’104,
 25 5:34-36 (“*As illustrated*, a data referencing instruction, such as the LOAD instruction 14”, is
 26 initially generated with a symbolic reference, *e.g.* “y”). So, references in the ’104 patent are
 27 either by location (a numeric reference) or by name (a symbolic reference).

28

1 Extrinsic evidence confirms that a “symbolic reference” is a reference by name.
2 According to Google’s own proffered dictionaries, a “symbolic address” is a “memory address
3 that can be referred to in a program by name rather than by number. The interpreter, compiler, or
4 assembler translates the name into the number that specifies the address” and a “symbol table” is
5 a “list of names used in a program with brief descriptions and storage addresses.” MICROSOFT
6 PRESS COMPUTER DICTIONARY 379 (2d ed. 1994) (symbolic address) (Peters Decl. Ex. 3);
7 WEBSTER’S NEW WORLD DICTIONARY OF COMPUTER TERMS 561 (5th ed. 1994) (symbol table)
8 (*id.*); JCCS at 9. No construction of this term is really necessary.

9 Google asks the Court to construe “symbolic reference” to mean something that is not
10 supported by the intrinsic evidence. First, Google proposes to construe “symbolic reference” as
11 one that is string- or character-based. The significant problem with Google’s approach is that
12 “string” does not appear in the claims, specification, prosecution history, or Google’s proffered
13 extrinsic evidence. Nor is there any mention of “character” in a way that is germane to “symbolic
14 reference.” There’s no reason to import these limitations into the claim.

15 Second, Google also proposes that the Court construe “symbolic” as “dynamic.” How
16 will this help the jury? How shall they know what makes a reference “dynamic”? Accepting
17 Google’s proposal threatens to require the Court to “construe the construction” should the
18 meaning of “dynamic” be disputed later. Google’s proposal again imports a limitation from the
19 specification into the claim. The specification teaches:

20 As shown in FIG. 7, upon receiving a data reference byte code, block 86, the main
21 interpretation routine determines if the data reference is static, i.e. numeric, or
22 *dynamic, i.e. symbolic*, block 88. If the data reference is a symbolic reference,
branch 88b, the main interpretation routine invokes the dynamic field reference
routine, block 90.

23 ’104, 5:10-15. The specification defines “dynamic” as symbolic, but not the other way around.
24 In other words, a “dynamic reference” is a “symbolic reference,” but there is no clear indication
25 in the specification that a “symbolic reference” has to be a “dynamic reference.”

26 Surely Google’s proposed construction is designed to set up a non-infringement argument
27 of some kind or another. Because Google still has not identified or described its argument in
28 response to Oracle’s interrogatories, Oracle cannot discuss how it relates to the proposed

1 construction. Regardless, Google’s construction is divorced from the ’104 patent specification
2 and should be rejected.

3 **V. REDUCED CLASS FILE (’702 PATENT)**

4

Claim Phrase	Oracle’s Proposed Construction	Google’s Proposed Construction
reduced class file	No construction necessary. A “reduced class file” contains a subset of the code and data contained in a class file	a class file containing a subset of the data and instructions contained in a corresponding original class file

5
6
7
8

9 The phrase “reduced class file” appears in every asserted claim of the ’702 patent. So far
10 as Oracle has been able to determine through the meet and confer process, the dispute between
11 the parties is about whether the “reduced class file” is itself a “class file.” The parties agree that a
12 “reduced class file” contains a subset of the code and data contained in a corresponding class file.
13 No construction is necessary because this idea is inherent in the word “reduced,” a word that the
14 jury can understand. The general idea of the invention is to eliminate code and data duplicated
15 among class files, and place only one copy of duplicated code and data in a multi-class file along
16 with the non-duplicated code and data. ’702, Abstract, 5:6-17. However, the ’702 patent is
17 specific about what a “class file” is, and a reduced class file does not qualify. (The parties dispute
18 the meaning of “class file” as well, and it’s not clear to Oracle why Google selected one phrase
19 for construction but not also the other, since the two phrases are so intimately related.)

20 The class file structure is outlined in Figure 3 and in column 7 of the ’702 patent:

21 Embodiments of the invention can be better understood with reference to aspects
22 of the class file format. Description is provided below of the Java class file format.
23 Also, enclosed as Section A of this specification are Chapter 4, “The class File
24 Format,” and Chapter 5, “Constant Pool Resolution,” of *The Java Virtual Machine Specification*,
by Tim Lindholm and Frank Yellin, published by Addison-Wesley in September 1996, ©Sun Microsystems, Inc.

25 ’702, 7:21-28. When source code is compiled, the definition of each class is stored by the
26 compiler in its own class file, even if more than one class was defined in one source code file.
27 See, e.g., ’702, 3:29-32, 7:31-32 (“A single class or interface file structure is contained in the
28 class file.”). This is why the ’702 patent mentions the disadvantages that result from “the fact that

1 a typical Java application can contain hundreds or thousands of small class files.” ’702, 4:2-4.
2 As mentioned above, the complete specification of the Java class file format is part of the ’702
3 specification, and may be found in columns 11 through 43.

4 “Reduced class files” do not satisfy the class file format specification. The key issue is
5 that a “class file is self-contained,” but examples of reduced class files described in the
6 specification violate that restriction. ’702, 4:3-4. As the ’702 patent explains, in one
7 embodiment, “duplicated constant elements are placed in the shared constant pool as a shared
8 element, and an element of the new constant type replaces the duplicated element in the reduced
9 pool to direct constant resolution to the shared element in the shared constant pool.” ’702, 9:58-
10 62. The “new constant type” contains an “index into the shared constant table.” ’702, 9:56-58.
11 So in this embodiment, the pre-processor places in the reduced class file an index that refers to a
12 shared constant pool entry *outside* the reduced class file. ’702, 9:63-65. But class files must be
13 self-contained, and do not—and cannot—contain indices to information stored elsewhere.
14 Indeed, this is one of the disadvantages of class files that the ’702 patent solves: each class file
15 has to contain its own constants, even though different class files may have the same constants,
16 creating duplication and inefficiency. ’702, 3:65-4:14; *see also* ’702, 4:45-60 (although a Java
17 archive (JAR) file encapsulates multiple class files, they “remain subject to storage inefficiencies
18 due to duplicated information between files”).

19 In contrast, reduced class files are not self-contained. Although of course reduced class
20 files can be very similar to class files, and their information can be stored similarly, embodiments
21 of the ’702 patent involve non-standard extensions to the class file format that permit code and
22 data from one class to index into a constant pool shared among classes. ’702, 9:65-10:5.

23 In the final analysis, the description of the embodiments in the specification makes clear
24 that a “reduced class file” has duplicated information removed (“duplicated constant elements are
25 placed in the shared constant pool as a shared element”) but also may have non-standard elements
26 added, such as new constant types and tags. ’702, 9:55-65. So if a construction of the term is
27 necessary, it should be just that a “reduced class file” contains a subset of the code and data
28 contained in a class file.

1 **VI. COMPUTER-READABLE MEDIUM ('104 PATENT)**
2 **COMPUTER-READABLE STORAGE MEDIUM ('720 PATENT)**
3 **COMPUTER-READABLE MEDIUM ('520 PATENT)**
4 **COMPUTER USABLE MEDIUM ('702 PATENT)**
5 **COMPUTER-READABLE MEDIUM ('447 PATENT)**
6 **COMPUTER-READABLE MEDIUM ('476 PATENT)**

Claim Phrase	Oracle's Proposed Construction	Google's Proposed Construction
computer-readable medium / computer-readable storage medium / computer usable medium	a storage device for use by a computer	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

14 Six of the seven asserted patents contain claims using the phrases “computer-readable
15 medium,” “computer-readable storage medium,” or “computer usable medium.” These claims
16 are often referred to as “Beauregard claims,” after *In re Beauregard*, 53 F.3d 1583 (Fed. Cir.
17 1995). This type of claim gained popularity after *Beauregard*, in which a patent applicant
18 challenged a Board of Patent Appeals’ rejection of software-related claims. Before the Federal
19 Circuit could rule, the United States Patent and Trademark Office changed its position, stating
20 that “computer programs embodied in a tangible medium, such as floppy diskettes, are patentable
21 subject matter under 35 U.S.C. § 101 and must be examined under 35 U.S.C. §§ 102 and 103.”
22 *Id.* at 1584. Based on this changed position, the PTO moved to dismiss the appeal, and the
23 Federal Circuit vacated the Board’s decision and remanded. *Id.*

24 Following *Beauregard*, a software invention claimed as a program embodied in a tangible
25 medium has been consistently recognized as statutory subject matter under § 101. *See* MANUAL
26 OF PATENT EXAMINING PROCEDURE § 2106.01 (8th ed. 6th rev. 2007) (explaining that “a claimed
27 computer-readable medium encoded with a computer program is a computer element which
28 defines structural and functional interrelationships between the computer program and the rest of

1 the computer which permit the computer program's functionality to be realized, and is thus
2 statutory.") (Peters Decl. Ex. 5). Beauregard claims allow inventors to claim as patentable
3 subject matter: computer programs embodied on a floppy disk, hard disk, CD, DVD, computer
4 memory, and similar storage media. Beauregard claims can be economically significant, in that
5 they can be directly infringed by software companies in ways that computer system claims,
6 requiring physical hardware, may not be.

7 The issue presented to the Court is this: some of the asserted patents (the '104, '520, '720,
8 and '702) describe only storage media as embodiments of computer-readable media, the others
9 (the '447 and '476) describe both storage media and transmission media. Google's claim
10 construction approach is to take a passage from the '447 and '476 patent specifications and apply
11 it to *all* of the patents-in-suit. One mistaken result, as explained below, is that Google is using
12 1997 evidence to construe a 1992 patent. Oracle submits that the proper way to construe a patent
13 claim is to apply the *Phillips* methodology to that claim, considering first *that* patent's intrinsic
14 evidence (its claims, specification, and prosecution history), and considering second the evidence
15 extrinsic to that patent (such as other patents and dictionaries), because the "best source for
16 understanding a technical term is the specification from which it arose, informed, as needed, by
17 the prosecution history" and external evidence, while useful, is "less significant than the intrinsic
18 record in determining the legally operative meaning of claim language." *Phillips*, 415 F.3d at
19 1315, 1317.

20 Under the proper methodology, the construction of "computer-readable medium,"
21 "computer-readable storage medium," and "computer usable medium" in the '104, '520, '720,
22 and '702 patents is straightforward. Each patent discloses only storage media as embodiments, so
23 the correct construction is "a storage device for use by a computer."

24 The construction of "computer-readable medium" in the '447 and '476 patents is more
25 complicated. Although these patents disclose both storage media and transmission media (such
26 as coaxial cable and fiber optics), the Court should decline to construe "computer-readable
27 medium" to cover transmission media and therefore avoid a potential invalidity issue.
28

1 In 2007, the Federal Circuit addressed whether a “signal” was patentable subject matter.
2 *In re Nuijten*, 500 F.3d 1346, 1353 (Fed. Cir. 2007). *Nuijten* concerned an appeal from a decision
3 of the Board of Patent Appeals and Interferences. The PTO had found that a claimed “storage
4 medium having stored thereon a signal with embedded supplemental data” was a “manufacture”
5 and patentable, but that a claim directed to the signal itself was not. *Id.* at 1351-53. The court
6 agreed, holding that the “claims on appeal cover transitory electrical and electromagnetic signals
7 propagating through some medium, such as wires, air, or a vacuum. Those types of signals are
8 not encompassed by any of the four enumerated statutory categories: process, machine,
9 manufacture, or composition of matter.” *Id.* at 1352.

10 We expect Google to argue that the “computer-readable medium” claims are invalid under
11 Section 101 for being drawn to ineligible subject matter. But there are no “signal” claims in this
12 case, and there is no question that a storage medium such as a computer memory is an article of
13 manufacture or a composition of matter, and therefore patentable subject matter. *See In re Lowry*,
14 32 F.3d 1579, 1582, 1584-85 (Fed. Cir. 1994). What the *Nuijten* court did not address is how a
15 district court should construe a “computer-readable medium” claim in an issued patent, when the
16 specification describes both storage and transmission media. As explained below, the correct
17 approach is to construe the phrase to mean only storage media, and preserve its validity.

18 We now apply the *Phillips* claim construction methodology to each patent in turn.

19 **A. '104 patent**

20 Claims 12 and 23 of the '104 patent claim a “**computer-readable medium** containing
21 instructions for controlling a data processing system to perform a method” The Court should
22 construe a computer-readable medium in the '104 patent to mean “a storage device for use by a
23 computer.” This construction is supported by the claim language, the specification, and the
24 prosecution history of the '104 patent.

25 When construing a patent claim term, it has “the meaning that the term would have to a
26 person of ordinary skill in the art in question at the time of the invention, i.e., as of the effective
27 filing date of the patent application.” *Phillips*, 415 F.3d at 1313. The '104 patent claims priority
28 to U.S. Patent Application No. 07/994,655, which was filed in 1992. As such, the term

1 “computer-readable medium” must be construed as if by a person of ordinary skill in the art in
2 1992. References from 1997 (and particularly references from patents dating to 1997, which may
3 reflect patent lawyer drafting preferences rather than the meaning attached to a term by skilled
4 artisans) are therefore irrelevant to the claim construction inquiry.

5 In the specification, all the examples of computer-readable media are storage media:
6 memories or storage devices to store code for use by a computer. ’104, 3:52-56, Fig. 2 (“As
7 shown in FIG. 2, the exemplary computer system 20 comprises a central processing unit (CPU)
8 22, a memory 24, and an I/O module 26. Additionally, the exemplary computer system 20 also
9 comprises a number of input/output devices 30 and a number of storage devices 28.”). Nowhere
10 does the specification suggest that these computer-readable media are “coaxial cables, copper
11 wire, fiber optics, acoustic or light waves, radio-waves and infra-red data communications” or
12 other transmission media as suggested by Google’s proposed construction.

13 The prosecution history confirms that a “computer-readable medium” in the ’104 patent is
14 a storage device for use by a computer. A claim directed to computer-readable media was first
15 introduced in the file history in 1996 in the ’204 reissue application, the parent reissue application
16 to the ’104 reissue application. In listing his reasons for filing the reissue application, the
17 applicant stated that a “further error was that the claims may be challenged by an infringer as not
18 reading literally on computer program code devices embodying the invention. Applicant is
19 informed and believes that subsequent to the Commissioner of Patents’ change of position in *In re*
20 *Beauregard*, claims more specifically directed to computer program code devices are now
21 permissible. I believe I have the right to further specifically claim computer program code
22 devices which embody my invention as supported by the original disclosure.” ’104 file history,
23 11/21/1996 Reissue Application Declaration of James Gosling at 3 (Peters Decl. Ex. 6).

24 The applicant particularly pointed out and distinctly claimed the subject matter of a
25 Beauregard claim: a computer program embodied on a tangible medium. His stated intent was to
26 capture the statutory, patentable subject matter identified by a Beauregard claim. Oracle’s
27 proposed construction—a storage device for use by a computer—is supported by the intrinsic
28 evidence. Google’s proposed construction is not.

1 Google’s proposed construction for the ’104 patent does not make sense in light of the
2 claim language, specification, or prosecution history. None of the intrinsic evidence or
3 contemporaneous extrinsic evidence identifies “coaxial cables, copper wire, fiber optics, acoustic
4 or light waves, radio-waves and infra-red data communications” as ’104 patent embodiments.
5 Therefore, it would be improper to construe the claims to cover such things.

6 **B. ’720 patent**

7 Claim 19 of the ’720 patent claims a “**computer-readable storage medium** holding code
8 for performing the method according to claim 10.” The Court should construe computer-readable
9 storage medium in the ’720 patent to mean “a storage device for use by a computer.” This
10 construction is supported by the plain meaning of the claim language and the specification.

11 The invention of claim 19 is a storage medium that holds code. Memories and disks are
12 storage media that hold code. “Coaxial cables, copper wire, fiber optics, acoustic or light waves,
13 radio-waves and infra-red data communications” are not storage media, and they do not hold
14 code. In the specification, all the examples of computer-readable storage media holding code are
15 memories or storage devices for use by a computer:

16 Each client 13 is operatively coupled to a storage device 15 and maintains a set of
17 classes 16 and class libraries 17

18 a server 18 is operatively coupled to a storage device 19 in which globally
shareable class libraries 20 are maintained.

19 Upon initialization, the master JVM process 33 reads an executable process image
20 from the storage device 35

21 ’720, 4:32-34, 4:40-42, 5:48-49. Figures 1 and 2 illustrate the storage devices holding code. The
22 specification does disclose computing devices connected by hardwired and wireless networks, but
23 does not describe the networks as being storage media holding code. Indeed, transmission media
24 cannot be storage media. Storage media store. Transmission media transmit. Transmission
25 media do not hold code. Google’s proposed construction, which includes transmission media, is
26 contrary to the claim language. It would be improper to import such limitations as “coaxial
27 cables, copper wire, fiber optics, acoustic or light waves, radio-waves and infra-red data
28

1 communications” into the construction of computer-readable storage medium for the ’720 patent.
2 The ’720 claim language and specification only support “a storage device for use by a computer.”

3 C. ’520 patent

4 Claim 18 and its dependent claims 19-23 of the ’520 patent claim a “**computer-readable**
5 **medium** containing instructions for controlling a data processing system to perform a
6 method” Like computer-readable storage medium in the ’720 patent, this phrase means “a
7 storage device for use by a computer.” This construction is supported by the plain meaning of the
8 claim language and the specification.

9 The analysis is the same as for the ’720 patent. The medium in the ’720 patent *holds*
10 code. The medium in the ’520 patent *contains* instructions. Memories and disks are computer-
11 readable media that contain instructions. Transmission media such as coaxial cables, copper
12 wire, and fiber optics, do not “contain” instructions. Considered in its full context, the ’520 claim
13 language points to computer-readable medium being “a storage device for use by a computer.”

14 The specification confirms this construction. The ’520 patent states:

15 Although an exemplary embodiment of the present invention is described as being
16 stored in **memory 206**, one skilled in the art will appreciate that it may also be
17 stored on **other computer-readable media, such as secondary storage devices**
18 **like hard disks, floppy disks, or CD-Rom**; a carrier wave received from the
Internet 204; or other forms of RAM or ROM. Additionally, one skilled in the art
will appreciate that computer 202 may contain additional or different components.

19 ’520, 4:48-56. The specification does not disclose coaxial cables, copper wire, fiber optics, or
20 other transmission media as examples of “computer-readable media” or embodiments of the
21 patent, and the claim should not be construed to cover them.

22 D. ’702 patent

23 Claims 7-12 of the ’702 patent are directed to a “computer program product comprising: a
24 **computer usable medium** having computer readable program code embodied therein for pre-
25 processing class files, said computer program product comprising” ’720, 51:43-46. In the
26 context of the ’702 claims and specification, this phrase means “a storage device for use by a
27 computer.”
28

1 Figure 1 of the '702 patent shows the software application (bytecode class files 107)
2 stored on server 100: “Bytecode class files 107 are stored (e.g., in temporary or permanent
3 storage) on server 100, and are available for download over network 101.” '702, 3:32-35. The
4 client computer that runs the software application is illustrated in Figure 2. Computer 200 is
5 shown and described as having a main memory 215 that may include DRAM ('702, 6:27-28),
6 mass storage 212 that “may include both fixed and removable media, such as magnetic, optical or
7 magnetic optical storage systems or any other available mass storage technology” ('702, 6:11-14),
8 and a communications interface 220 that may use a variety of forms of electrical,
9 electromagnetic, or optical signals to transport application code. '702, 6:37-7:14. The client
10 computer may receive application code through the communication interface, and the “received
11 code may be executed by CPU 213 as it is received, and/or stored in mass storage 212, or other
12 non-volatile storage for later execution.” '702, 7:10-12.

13 In the '702 patent, the “computer program product comprising a computer usable medium
14 having computer readable program code embodied therein” is found in the storage devices on the
15 server 100 and the client computer 200. The product is stored on the server, and can be
16 downloaded upon request to the client, where it may be run on by the processor or stored on a
17 local file system. '702, 3:47-59. Although transmission media can be used to transport
18 information and permit downloading, copper wires, coaxial cables, fiber optics, and the like are
19 not part of a “computer program product” and do not have “program code embodied therein.”
20 Accordingly, the phrase “computer usable medium” in the '702 patent should not be construed to
21 cover such.

22 E. '447 and '476 patents

23 Claims 10-18 of the '447 patent and claims 10-13 and 15-18 of the '476 patent claim a
24 “**computer-readable medium** carrying one or more sequences of one or more instructions”
25 Claim 14 of the '476 patent claims a “**computer-readable medium** bearing instructions for
26 providing security”

27 Like the other patents, the '447 and '476 specifications disclose non-volatile and volatile
28 media such as optical or magnetic disks, storage devices, main memory, floppy disks, hard disks,

1 magnetic tape, CD-ROMs, RAM, and other physical media. '447, 5:18-25; '476, 5:18-25.

2 **Unlike** the other patents, the '447 and '476 patents **also** disclose that a “computer-readable
3 medium” may be a transmission medium:

4 The term “computer-readable medium” as used herein refers to any medium that
5 participates in providing instructions to processor 104 for execution. Such a
6 medium may take many forms, including but not limited to, non-volatile media,
7 volatile media, and transmission media. Non-volatile media includes, for example,
8 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.

9 '447, 5:4-16; '476, 5:4-16. Google will argue that this disclosure means that the computer-
10 readable medium claims are invalid under Section 101 for embracing non-patentable subject
11 matter.

12 But that is not the right way to analyze this issue. When construing a patent claim term, it
13 is important to give the claim term “the meaning that the term would have to a person of ordinary
14 skill in the art in question at the time of the invention, *i.e.*, as of the effective filing date of the
15 patent application.” *Phillips*, 415 F.3d at 1313. In December 1997, when the specifications were
16 written, and in 2000 and 2001 when the claims were issued by the PTO after examination, no one
17 thought the claims were not statutory subject matter under Section 101. To the contrary: 1994’s
18 *Lowry* and 1995’s *Beauregard* ruled the day, and the PTO’s considered position was that “a
19 claimed computer-readable medium encoded with a computer program defines structural and
20 functional interrelationships between the computer program and the medium which permit the
21 computer program’s functionality to be realized, and is thus statutory.” EXAMINATION
22 GUIDELINES FOR COMPUTER-RELATED INVENTIONS 9 (1996) (Peters Decl. Ex. 7).⁶ It was not until
23 more than ten years later that the *Nuijten* court held that a “signal per se” did not belong to any of
24 the four enumerated statutory categories: process, machine, manufacture, or composition of
25 matter. Yet the Federal Circuit did not hold that a “computer-readable medium” is not statutory
26 matter, nor would it be expected to.

27 _____
28 ⁶ Available at <http://www.uspto.gov/web/offices/pac/dapp/pdf/ciig.pdf>.

1 There is nothing inherent in the phrase “computer-readable medium” that compels a court
2 to conclude that it embraces unpatentable subject matter. There is ample support for the
3 proposition that a computer-readable medium is an article of manufacture or a composition of
4 matter, not only in the intrinsic evidence, but also in the extrinsic evidence. Looking to several
5 contemporary technical dictionaries,⁷ Oracle notes that definitions of “medium” and “media” in
6 the computer context encompass storage devices: MICROSOFT PRESS COMPUTER DICTIONARY 301
7 (3d ed. 1997) (media defined as “The *physical material*, such as paper, disk, and tape, used for
8 storing computer-based information.”); THE ILLUSTRATED DICTIONARY OF ELECTRONICS 430 (7th
9 ed. 1997) (medium defined as “In a computer system, that *storage* device onto or into which data
10 is recorded for input into memory (e.g. magnetic disk, magnetic tape, optical disk, etc.).”); THE
11 IEEE STANDARD DICTIONARY OF ELECTRICAL AND ELECTRONICS TERMS 643 (6th ed. 1996)
12 (medium defined as “(1) (computers) The *material*, or configuration thereof, on which data are
13 recorded; for example, paper tape, cards, magnetic tape.”). (Peters Decl. Ex. 8). Each of these
14 definitions falls well within one of the statutory categories and describes a medium having a non-
15 transitory character: a physical material, a storage device, or a material.

16 Given the choice to construe an issued patent claim to cover statutory or nonstatutory
17 subject matter, it would be unreasonable choose to construe the claim to cover nonstatutory
18 subject matter and risk invalidating the claim. Such a construction would not comport with the
19 intent of the inventor or the PTO’s issuance of the patent. The principle that “claims should be so
20 construed, if possible, as to sustain their validity” applies here. *Rhine v. Casio, Inc.*, 183 F.3d
21 1342, 1345 (Fed Cir. 1999); *see also Astra Aktiebolag v. Andrx Pharms., Inc.*, 222 F. Supp. 2d
22 423, 458 (S.D.N.Y. 2002) (“Whenever a claim is susceptible to one construction that would
23 render it valid and another construction that would render it invalid, the claim will be construed to
24 sustain its validity.”). The disclosure of transmission media in the ’447 and ’476 specifications
25 does not bar application of the principle. Even when the specification contains an express

26 _____
27 ⁷ “Dictionaries or comparable sources are often useful to assist in understanding the commonly
28 understood meaning of words and have been used both by our court and the Supreme Court in
claim interpretation.” *Phillips*, 415 F.3d at 1322.

1 definition of a claim term, a court may construe it more narrowly. *See, e.g., Trading Techs. Int'l,*
2 *Inc. v. eSpeed, Inc.*, 595 F.3d 1340, 1353-55 (Fed. Cir. 2010) (holding that, despite an express
3 definition in the specification, the district court was correct to make “two important changes to
4 this express definition in construing the word”); *Ecolab, Inc. v. FMC Corp.*, 569 F.3d 1335, 1344-
5 45 (Fed. Cir. 2009) (holding that, although the patent specification provided an express definition
6 of “sanitize” (it “denote[s] a bacterial population reduction to a level that is safe for human
7 handling and consumption”), the term should be construed to “mean that the treated meat has
8 become safe for human handling and post-cooking consumption.”).

9 The Court should construe “computer-readable medium” in a way that sustains its
10 validity. As the Supreme Court held in *Warner-Jenkinson* and reaffirmed in *Festo*: “To change
11 so substantially the rules of the game now could very well subvert the various balances the PTO
12 sought to strike when issuing the numerous patents which have not yet expired and which would
13 be affected by our decision.” *Festo Corp. v. Shoketsu Kinzoku Kogyo Kabushiki Co.*, 535 U.S.
14 722, 739 (2002) (quoting *Warner-Jenkinson Co. v. Hilton Davis Chem. Co.*, 520 U.S. 17, 32 n.6
15 (1997)). As the Court considers how to construe PTO-examined and -issued Beauregard claims,
16 each entitled to a presumption of validity, it should do so in a manner that does not “risk
17 destroying the legitimate expectations of inventors in their property” or “unfairly discount the
18 expectations of a patentee who had no notice at the time of patent prosecution.” *Id.*

19
20
21 Dated: March 17, 2011

MICHAEL A. JACOBS
MARC DAVID PETERS
MORRISON & FOERSTER LLP

22
23 By: /s/ Michael A. Jacobs

24 *Attorneys for Plaintiff*
25 ORACLE AMERICA, INC.