

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

MICROSOFT CORPORATION,
Petitioner,

v.

ENFISH, LLC,
Patent Owner.

Case IPR2013-00563
Patent 6,151,604

Before THOMAS L. GIANNETTI, BRYAN F. MOORE,
and BARBARA A. PARVIS, *Administrative Patent Judges*.

PARVIS, *Administrative Patent Judge*.

FINAL WRITTEN DECISION
35 U.S.C. § 318(a) and 37 C.F.R. § 42.73

I. INTRODUCTION

A. *Background*

Microsoft Corporation (“Microsoft”) filed a petition to institute an *inter partes* review of claims 16–30 and 46–60 of U.S. Patent No. 6,151,604 (“the ’604 patent”). Paper 1 (“Pet.”). We instituted trial for claims 16–26, 30, 46–56, and 60 of the ’604 patent on certain grounds of unpatentability alleged in the Petition. Paper 14 (“Decision to Institute” or “Inst. Dec.”).

After institution of trial, Patent Owner, Enfish, LLC (“Enfish”), filed a Patent Owner Response, along with a Declaration by Dr. H.V. Jagadish (“Jagadish Declaration”).

On September 16, 2014, Enfish filed an unopposed motion to correct typographical errors in both papers and filed therewith the corrected Patent Owner Response (“PO Resp.,” Paper 42) and the corrected Jagadish Declaration (Ex. 2007). We grant Enfish’s motion to correct both papers.¹ Microsoft filed a Reply (“Pet. Reply”). Paper 37.

A consolidated hearing for IPR2013-00559, IPR2013-00560, IPR2013-00561, IPR2013-00562, and IPR2013-00563 was held on December 3, 2014. The transcript of the consolidated hearing has been entered into the record. Paper 64 (“Tr.”).

We have jurisdiction under 35 U.S.C. § 6(c). This final written decision is issued pursuant to 35 U.S.C. § 318(a).

Microsoft has shown by a preponderance of the evidence that claims 46–50 and 54 of the ’604 patent are unpatentable. Microsoft has not shown

¹ For ease of reference, throughout we refer to Enfish’s corrected Patent Owner Response (Paper 42) and corrected Jagadish Declaration (Exhibit 2007), both filed on September 16, 2014.

that claims 51–53, 55, 56, and 60 of the '604 patent are unpatentable. For the reasons discussed below, for claims 16–26 and 30, we are unable to reach a determination on the alleged grounds of unpatentability over prior art. Accordingly, we terminate this proceeding with respect to claims 16–26 and 30 under 37 C.F.R. § 42.72.

B. Additional Proceedings

In addition to this petition, Microsoft has filed a petition challenging the patentability of claims 1–15 and 31–45 of the '604 patent. *See Microsoft Corp. v. Enfish, LLC*, IPR2013-00562. Microsoft indicates that claims 1, 2, 16, 17, 31, 32, 46, and 47 of the '604 patent have been asserted against Microsoft in *Enfish LLC v. Microsoft Corporation, et al.*, Case No. 12-cv-7360 MRP, in the Central District of California (“California case”). Pet. 2. Microsoft further contends that a final judgment against Enfish has been entered in the California case. Ex. 1168.

Microsoft also has filed three petitions challenging the patentability of claims 1–60 of U.S. Patent 6,163,775 (“the '775 patent”).² *See Microsoft Corp. v. Enfish, LLC*, IPR2013-00559; IPR2013-00560; IPR2013-00561.

C. The '604 Patent

The '604 patent (Ex. 1101) is titled, “Method and Apparatus for Improved Information Storage and Retrieval System,” and generally relates to a system and method for data storage, manipulation, and retrieval in a logical table of a database. This table is a logical structure, not a physical structure, stored in the memory. Ex. 1101, 6:31–33.

² The '604 patent and the '775 patent both issued from continuations of application No. 08/383,752, filed March 28, 1995, now U.S. Patent No. 5,729,730. Pet. 6.

Figure 3 is reproduced below:

	120	122	130	124	134	126	132	100
108	OBJECT ID	TYPE [# 101]		[#1012] LABEL	ADDRESS [#1013]	EMPLOYED BY [#1019]	TITLE [#1033]	AUTHOR [#1032]
110	#1100	#1020 [COMPANY]		DEXIS	117 EAST COLORADO		N/A	N/A
138	#1101	#1010 [PERSON]		SCOTT WLASCHIN		#1100 [DEXIS]	N/A	N/A
	#1118	#1030 [BOOK]						#1122
	#1122	#1050 [MEMO]						#1122
	#1127	#1060 [DOCUMENT]			C:\WORD\ PROJ.DOC		PROJECT PLAN	#1101
136	#1019	# 210 [FIELD]		EMPLOYED BY				
135	# 210	# 111 [TYPE]		COLUMN				
140	# 111	# 111 [TYPE]		TYPE				

Figure 3 of the '604 patent illustrates a structure of a logical table. Ex. 1101, 3:30–31. As depicted by Figure 3 of the '604 patent, above, table 100 is defined by rows 108, 110, 138, 136, 135, and 140, and columns 120, 122, 124, and 126. *Id.* at Fig. 3. The intersection of a row and a column defines a cell in the table. *Id.* at 6:40–41. Each column corresponds to an attribute spanning various records. *Id.* at 6:38. An attribute is a single class description, such as an employer, denoted in column 126 of Figure 3, for example, by the text “Employed By.” *Id.* at 7:16–18.

Each row corresponds to a record spanning various attributes. Ex. 1101, 6:37–38. For example, row 110 corresponds to a company as shown in cell 130. *Id.* at 6:54–55.

D. Illustrative claims

Of the challenged claims, the independent claims are 16, 17, 24, 46, 47, and 54. Each of claims 18–23 depend, directly or indirectly, from claim 17. Claims 25, 26, and 30 depend, directly or indirectly, from claim 24.

Claims 48–53 depend, directly or indirectly, from claim 47. Claims 55, 56, and 60 depend, directly or indirectly, from claim 54. Claims 16, 46, and 47 illustrate the claimed subject matter and are reproduced below:

16. A data storage and retrieval system for a computer memory, comprising:

means for configuring said memory according to a logical table, said logical table including:

a plurality of logical rows, each said logical row including an object identification number (OID) to identify each said logical row, each said logical row corresponding to a record of information; and

a plurality of logical columns intersecting said plurality of logical rows to define a plurality of logical cells, each said logical column including an OID to identify each said logical column, wherein said OID's are variable length.

46. A method for storing and retrieving data in a computer memory, comprising the steps of:

configuring said memory according to a logical table, said logical table including:

a plurality of logical rows, each said logical row including an object identification number (OID) to identify each said logical row, each said logical row corresponding to a record of information; and

a plurality of logical columns intersecting said plurality of logical rows to define a plurality of logical cells, each said logical column including an OID to identify each said logical column, wherein said OID's are variable length.

47. A method for storing and retrieving data in a computer memory, comprising the steps of:

configuring said memory according to a logical table, said logical table including:

a plurality of logical rows, each said logical row including an object identification number (OID) to identify each said logical row, each said logical row corresponding to a record of information;

a plurality of logical columns intersecting said plurality of logical rows to define a plurality of logical cells, each said logical column including an OID to identify each said logical column; and

indexing data stored in said table.

E. The Prior Art References Supporting Alleged Unpatentability

Microsoft relies on the following references:

MICROSOFT® VISUAL BASIC™ PROGRAMMING SYSTEM FOR WINDOWS™ VERSION 3.0 (1993) (“Visual Basic,” Ex. 1104).

American National Standard for Information Systems – Database Language – SQL (ANSI INCITS 135-1992 (R1998) (“SQL-92,” Ex. 1105).

Jensen et al., U.S. Patent No. 5,615,362 (issued Mar. 25, 1997) (“Jensen,” Ex. 1106).

GERARD SALTON & MICHAEL J. MCGILL, INTRODUCTION TO MODERN INFORMATION RETRIEVAL (1983) (“Salton,” Ex. 1107).

Smith et al., U.S. Patent No. 5,181,162 (issued Jan. 19, 1993) (“Smith ’162,” Ex. 1108).

Sudarshan Chawathe et al., *The TSIMMIS Project: Integration of Heterogeneous Information Sources* (Dep’t Computer Sci. Stanford Univ.) (“Chawathe,” Ex. 1116).

F. The Pending Grounds of Unpatentability

The following chart summarizes Microsoft’s patentability challenges:

Reference[s]	Basis	Claims Challenged
Visual Basic	§ 102(b)	17, 18, 24, 47, 48, and 54
SQL-92 and Chawathe	§ 103	16 and 46
Visual Basic and Jensen	§ 103	19, 20, 49, and 50
Visual Basic and Salton	§ 103	23, 25, 26, 30, 53, 55, 56, and 60
Visual Basic, Jensen, and Salton	§ 103	21 and 51
Visual Basic and Smith ’162	§ 103	22 and 52

In support of the above-referenced grounds of unpatentability, Microsoft relies on the Declaration of Dr. Hosking (Ex. 1119, “Hosking Declaration”).

II. CLAIM CONSTRUCTION

A. Legal Standard

In an *inter partes* review, claim terms in an unexpired patent are interpreted according to their broadest reasonable construction in light of the specification of the patent in which they appear. 37 C.F.R. § 42.100(b); *see also In re Cuozzo Speed Techs., LLC.*, No. 14-01301, slip op. at 16, 19 (Fed. Cir. Feb. 4, 2015) (“Congress implicitly adopted the broadest reasonable interpretation standard in enacting the AIA,” and “the standard was properly adopted by PTO regulation.”). Under the broadest reasonable construction

standard, claims are to be given their broadest reasonable interpretation consistent with the specification, and the claim language should be read in light of the specification as it would be interpreted by one of ordinary skill in the art. *In re Am. Acad. of Sci. Tech. Ctr.*, 367 F.3d 1359, 1364 (Fed. Cir. 2004). Also, we must be careful not to read a particular embodiment appearing in the written description into the claim, if the claim language is broader than the embodiment. *See In re Van Geuns*, 988 F.2d 1181, 1184 (Fed. Cir. 1993) (“[L]imitations are not to be read into the claims from the specification.”).

B. Overview of the Parties’ Positions

Microsoft contends that this case involves a computer-implemented invention. Pet. 6. Microsoft also contends that the challenged independent claims invoke means-plus-function claiming, but the ’604 patent fails to disclose an algorithm, which is an issue under § 112, sixth paragraph, that cannot be addressed in this proceeding. Pet. 10. In the Decision to Institute, we invited Enfish to direct us to the specific portions of the specification that clearly link or associate a computer program or algorithm to the function corresponding to the claimed means. Inst. Dec. 11. For the reasons discussed below, we determine that Enfish in its response does not identify sufficient corresponding structure, as required under 35 U.S.C. § 112, sixth paragraph, for the “means for configuring said memory according to a logical table,” recited in claims 16, 17, and 24.

Regarding terms in the method claims, in the Decision to Institute, we provided constructions for “logical table” and “object identification number,” which are shown in the table below. Inst. Dec. 11–13. We also determined that no express construction is needed for the following terms:

“memory,” “logical row,” and “logical column.” Inst. Dec. 14.

Claim Term or Phrase	Construction in the Decision to Institute
“logical table”	“[W]e construe the term “table” to mean: ‘a structure of a database comprising rows and columns.’” Inst. Dec. 12. “We determine no express construction of ‘logical’ is needed for this decision.” <i>Id.</i>
“object identification number”	“[W]e construe ‘identification number’ in light of the specification to mean: ‘an array of bits that define.’ For the purpose of this decision, an express construction of ‘object’ is not necessary.” <i>Id.</i> at 13.

Enfish contends that our construction for “object identification number” is incomplete. PO Resp. 16. We evaluate Enfish’s contention below. Enfish also provides contentions regarding two other terms: “anchor,” recited in claims 21 and 51, and “external documents,” recited in claims 30 and 60. PO Resp. 23. Because we determine that Microsoft has not met its burden with respect to these claims for other reasons, no construction of these terms for the purposes of this Decision is necessary.

We discern no reason, based on the complete record now before us, to change our constructions in the Decision to Institute of “logical table” and “object identification number.”

C. Analysis of the Parties’ Claim Construction Positions

1. Means for configuring said memory according to a logical table

Independent claims 16, 17, and 24 include the limitation “means for configuring said memory according to a logical table.” In the Decision to Institute, we agreed with Microsoft that under the broadest reasonable interpretation, the function for the means for configuring is “configuring

memory according to a logical table.” Inst. Dec. 11. Additionally, we considered the corresponding structure for the recited function as including a general purpose computer. *Id.* Enfish does not challenge persuasively either of these determinations; however, Enfish identifies portions of the specification that Enfish contends provide algorithmic support for the recited function. PO Resp. 24–26. In particular, Enfish contends that the ’604 patent discloses a four-step algorithm that is linked to the recited function of configuring memory according to a logical table. *Id.*

“[T]he corresponding structure for a § 112 ¶ 6 claim for a computer-implemented function is the algorithm disclosed in the specification.” *Aristocrat Techs. Austl. Party Ltd. vs. Int’l Game Tech.*, 521 F.3d 1328, 1333 (Fed. Cir. 2008) (quoting *Harris Corp. v. Ericsson Inc.*, 417 F.3d 1241, 1249 (Fed. Cir. 2005)). Additionally, specific portions of the specification must clearly link or associate a computer program or algorithm to the function corresponding to the claimed means. *See Medical Inst. & Diag. Corp. v. Elektra AB*, 344 F.3d 1205, 1211 (Fed. Cir. 2003). For the reasons set forth below, we conclude that the four steps and other ’604 patent specification portions identified by Enfish do not describe an algorithm for the recited function of “configuring memory according to a logical table.” We further conclude that Enfish has not clearly linked or associated the recited function to the four steps or the portions of the ’604 patent specification that Enfish identifies.

For algorithmic support, Enfish identifies disparate excerpts of the ’604 patent specification, which do not link or associate clearly a computer program or algorithm to the function corresponding to the claimed means for configuring said memory according to a logical table. For example, the first

step, “[c]reate, in a computer memory, a logical table” (PO Resp. 24) appears to be similar to the recited function of configuring memory according to a logical table, but the first step is not found in the ’604 patent specification. Additionally, none of the portions of the ’604 patent specification that Enfish cites for this first step provide an algorithm or computer program for performing the recited function of “configuring memory according to a logical table” or clearly link or associate any algorithm or program to this recited function. PO Resp. 24 (citing Ex. 1101, Abstract, 2:53–57, 6:30–38, Figs. 1, 3, 9). These portions describe that an already-formed table has rows and columns, without describing how memory is configured to create a logical table having rows and columns. Ex. 1101, 2:53–57. One portion states that memories “need not store” the table contiguously, but fails to describe an algorithm or computer program for configuring memory such that a logical table is not stored contiguously. *Id.* at 6:30–38.

Enfish’s three remaining steps and the other ’604 patent specification portions identified by Enfish fail to remedy these deficiencies. The second step is: “[a]ssign each row and column an object identification number (OID) that, when stored as data, can act as a pointer to the associated row or column.” PO Resp. 25. One of the portions of the ’604 patent specification cited by Enfish for the second step indicates “the system must generate a unique OID when columns and rows are formed.” Ex. 1101, 8:6–7. The remainder of the identified portions relate to assigning an OID or indicate that an OID “may be used” as a pointer, without describing an algorithm for forming columns and rows of a table or showing how assigning an OID relates to steps for forming columns and rows. Ex. 1101, Abstract, 2:54–55,

6:42–49, 6:60–61, 8:10–51, Figs. 3, 4. Additionally, the identified portions discuss assigning a numeric value to an OID in the form of a bit array, but fail to describe how to configure memory such that an OID may be used as a pointer. *Id.*

The third step is a feature of claim 1 (PO Resp. 12). That claim is not challenged in this proceeding. Furthermore, the portions of the '604 patent specification cited by Patent Owner for the third step (PO Resp. 26 (citing Ex. 1101, Abstract, 2:60–63, 6:39–40, 7:16–22, Fig. 3)) indicate that a table has a row that corresponds to columns, such as a header row, which is a generic feature of an already-formed table. These '604 patent specification portions identified by Enfish (*id.*), however, do not describe how to form a table with this feature or link or associate this feature to the recited function.

The fourth step, i.e., storing and accessing data in cells (PO Resp. 26), is performed in an already-configured table and, therefore, occurs after the recited function of configuring a table has occurred. Nonetheless, the '604 patent specification portions identified by Enfish (*id.* (citing Ex. 1101, Abstract, 2:57–60, 6:40–41, 6:53–54, 7:1–2, 7:14–15, 11:47–59, Fig. 10)) suffer from the same deficiencies noted above.

We conclude that the four steps and '604 patent specification portions identified by Enfish do not describe an algorithm for configuring memory in accordance with the claimed table. The portions of the '604 patent specification identified by Enfish describe an already-formed table having generic characteristics, such as columns, rows, identifiers, and a header row. *See, e.g.*, Ex. 1101, Fig. 3. The description, however, does not disclose any algorithm or computer program for forming this table. Additionally, the description identified by Enfish is not linked or associated clearly with the

recited function of “configuring memory according to a logical table.”

Enfish also relies on Dr. Jagadish’s Declaration to show that the ’604 patent specification provides an algorithm and clearly links the algorithm to the recited function. PO Resp. 24–26 (citing Ex. 2007 ¶¶ 68–74). The Jagadish Declaration, however, relies on similar portions of the ’604 patent specification to those cited in Patent Owner’s Response. For the reasons given, the Jagadish Declaration does not support Enfish’s assertion. Dr. Jagadish further asserts, “[i]t is also my opinion that one of ordinary skill in the art would understand how to implement those algorithm steps using techniques and resources that were available at the time the ’775 Patent was filed.” Ex. 2007 ¶ 69. Enfish, however, cannot rely on the knowledge of one skilled in the art to address the deficiencies noted above. *See Function Media, LLC v. Google Inc.*, 708 F.3d 1310, 1319 (Fed. Cir. 2013) (“Having failed to provide any disclosure of the structure . . . FM cannot rely on the knowledge of one skilled in the art to fill in the gaps.”)

Enfish, furthermore, identifies the same algorithmic support for each of independent claims 16, 17, and 24 (PO Resp. 22–24). We are not persuaded by this argument because each of these claims recites a different table. Enfish contends that these differences are addressed because certain features in the algorithm are identified as optional. Tr. 43:1–43:23. This argument is not convincing because Enfish does not specify the claim or claims supported by each of these optional features. Furthermore, the ’604 patent specification portions identified by Enfish do not disclose a computer program or algorithm for configuring memory according to a logical table, regardless of whether so-called optional features are included.

Because we conclude that the four steps and ’604 patent specification

portions identified by Enfish do not describe an algorithm for configuring memory in accordance with the claimed table, we terminate this proceeding with respect to the claims that recite this means-plus-function limitation. As explained in *BlackBerry Corporation v. Mobile Media Ideas, LLC*, Case IPR2013-00036 (PTAB Mar. 7, 2014) (Paper 65), the specification must disclose enough of a specific algorithm to provide the necessary structure under § 112, sixth paragraph. In the circumstance when the specification of the challenged patent lacks sufficient disclosure of structure under 35 U.S.C. § 112, sixth paragraph, the scope of the claims cannot be determined without speculation and, consequently, the differences between the claimed invention and the prior art cannot be ascertained. *Id.* For the reasons given, we determine that independent claims 16, 17, and 24 are not amenable to construction and, thus, we terminate this proceeding with respect to these claims under 37 C.F.R. § 42.72. Because claims 18–23 depend, directly or indirectly, from claim 17, and claims 25, 26, and 30 depend, directly or indirectly, from claim 24 we also terminate this proceeding with respect to these claims under 37 C.F.R. § 42.72.

2. Other Means-Plus-Function Terms

Because we determine that this proceeding should be terminated as to claims 16–26 and 30 for the reasons discussed above, we need not construe other means-plus-function terms appearing in those claims for the purposes of this Decision.

3. Object Identification Number (OID)

Independent claims 46, 47, and 54 recite “object identification number (OID)” and “OID.” In the Decision to Institute, we construed “identification

number” in light of the specification to mean: “an array of bits that define.” Inst. Dec. 13 (citing Ex. 1101, 3:32–34, 8:35–37). We also decided that an express construction of “object” is not necessary.

Enfish “agrees in principle” with our construction, but contends that the construction is incomplete. PO Resp. 16. Enfish contends that “OID” should be defined further as “a unique, immutable, and system-generated value that identifies an object.” *Id.* at 15. Microsoft contends that Enfish seeks to read into the claims extraneous limitations that are unsupported by either intrinsic or extrinsic evidence. Pet. Reply 2. Microsoft additionally contends that Enfish’s proposed construction does not provide an appropriate context for the additional limitations. *See, e.g.*, Pet. Reply 3.

At the heart of Enfish’s contention is an assertion that an OID will not function properly if it is not unique, immutable, and system generated. PO Resp. 17–22. For example, Enfish, relying on Dr. Jagadish, contends that an OID must be a unique value because “if an OID were not unique the database would be non-functional because objects could not be reliably retrieved.” PO Resp. 17 (citing Ex. 2007 ¶ 51). Dr. Jagadish cites to a portion of the specification stating that the OID is used for “exact retrieval” and states that without uniqueness, retrieval using an OID will result in more than one object. Ex. 2007 ¶ 51 (citing Ex. 1101, 1:60–64).

We determine that Enfish’s proposed construction of OID introduces requirements beyond what is needed for an OID to identify data and retrieve data from memory. Enfish, for example, contends that “an OID is unique among all rows and columns of the logical table.” PO Resp. 32. Additionally, each of the independent claims recites that an OID is included “to identify” each logical row and each logical column, but does not

specifically recite using an OID to retrieve data from memory.

Microsoft advances additional contentions regarding why Enfish's proposed construction should not be adopted. For example, Microsoft contends that Enfish's "unique" OID requirement would conflict with the patent, which describes a row in a table with an OID that has the same value as the OID of a column in the same table. Pet. Reply 3 (citing Ex. 1101, Fig. 3). Microsoft also contends that "immutable" should not be imported into the construction of OID because the term is absent from the specification, claims, and file history, and the specification is devoid of disclosure relating to immutability of an OID. Pet. Reply 3–4. As further support, Dr. Hosking states that an OID may change in certain circumstances and still retrieve information reliably. Ex. 1143 ¶¶ 21, 22.

We determine that adopting Enfish's proposed construction without further context would create ambiguity. Additionally, limitations are not to be read into the claims from embodiments in the specification. *See In re Van Geuns*, 988 F.2d at 1184. Furthermore, we agree with Microsoft that Enfish's reliance on extrinsic evidence is misplaced. Enfish, for example, relies on excerpts of "text books" (PO Resp. 21) for its contention that an OID is "immutable," without showing sufficiently that the definition ascertained from these excerpts of extrinsic evidence is consistent with the definition that would be ascertained by reading the patent documents.

We, therefore, determine that the construction of "OID" in the Decision to Institute should not be changed.

III. ANALYSIS

A. Alleged Anticipation of Claims 47, 48, and 54 by Visual Basic

For the reasons given below, after consideration of the arguments in Enfish’s Patent Owner Response, and the evidence cited therein, we conclude that Microsoft has shown, by a preponderance of the evidence, that each of claims 47, 48, and 54 is unpatentable as anticipated by Visual Basic.

1. Visual Basic

Visual Basic describes a programming system for the Windows™ operating system that enables programmers to create databases. Ex. 1104, PF 1, PG 453.³ The programming system uses objects to represent tables of a database. *Id.* at PF 47. As described in Visual Basic, a table object is a logical representation of a physical table with records (rows) and fields (columns). *Id.* at LR 558.

Visual Basic also describes adding an index to a database, “[a]dding an index to your database can increase the speed with which you get access to information.” Ex. 1104, PF 37. Visual Basic provides an example of an index with key fields Name, Last Name, City, and Zip Code. *Id.* at 38.

2. Claims 47 and 54

We have reviewed Microsoft’s anticipation argument, supporting evidence, and the detailed claim analysis, which reads persuasively all elements of each of claims 47 and 54 onto the disclosure of Visual Basic.

³ Citations in this decision to Exhibit 1104 refer to the volumes Programmer’s Guide (PG), Language Reference (LR), and Professional Features (PF), respectively, as well as the page number within each volume.

Pet. 31–43 (citing Ex. 1104, LR 162, 185, 276, 277, 280, 532, 425, 558, PF 1, 3, 18, 37–40, 47, 76, 80, PG 471, 481, 482, 484; Ex. 1119 ¶¶ 68–93, 119). For example, we are persuaded that Microsoft has shown by a preponderance of the evidence that Visual Basic discloses configuring memory according to a logical table, as recited in claims 47 and 54. In the cited portions above, Visual Basic discloses storing data in a computer formatted as a table object, which is “a logical representation of a physical table.” Ex. 1104, LR 558. The table object comprises “records (rows) and fields (columns).” *Id.*

As an additional example, we are persuaded that Microsoft has shown by a preponderance of the evidence that Visual Basic discloses each row and each column including an OID, as recited in claims 47 and 54. In the cited portions above, Visual Basic describes assigning a primary key and that “[a] table’s primary key is the determining factor when testing to see if the record is unique within the table.” Ex. 1104, PF 39. According to Visual Basic “[Visual Basic] creates an index on the primary key of the table and uses it to find records and to create joins between tables.” *Id.* Regarding each column including an OID, according to Visual Basic, “[e]ach Field in the Fields collection of a TableDef has a unique value in the OrdinalPosition property.” *Id.* at LR 425.

Furthermore, we are persuaded that Microsoft has shown by a preponderance of the evidence that Visual Basic discloses indexing data stored in the table, as recited in claims 47 and 54, and a cell including a pointer to an index record, as recited in claim 54. In the cited portions above, Visual Basic discloses an index object used for creating indexes to “speed up the process of finding records.” Ex. 1104, LR 276–277; *see also*

id. at PF 37, 38, PG 482. Visual Basic also discloses a cell pointing to an index object. *Id.* at LR 280, PG 484.

Enfish contends that Microsoft has not met its burden of showing that Visual Basic discloses logical column OIDs and logical row OIDs. PO Resp. 30–34. Enfish’s contentions are based on its proposed construction of OID, which we decline to adopt for the reasons given above.

For the foregoing reasons, Microsoft has shown, by a preponderance of the evidence, that claims 47 and 54 of the ’604 patent are anticipated by Visual Basic.

3. *Claim 48*

Claim 48 depends directly from independent claim 47. Microsoft has shown by a preponderance of the evidence that Visual Basic describes the additional elements recited in claim 48 of searching the table for a key word and inserting into the table a record corresponding to the key word. Pet. 42–43 (citing Ex. 1104, PF 1, 37, PG 487, Fig. 20.6; Ex. 1119 ¶ 91). Accordingly, Microsoft has demonstrated by a preponderance of the evidence that claim 48 of the ’604 patent is unpatentable, under 35 U.S.C. § 102(b), as anticipated by Visual Basic.

B. Obviousness of Claim 46 over SQL-92 and Chawathe

For the reasons given below, after consideration of Enfish’s Patent Owner Response, and the evidence cited therein, we conclude that Microsoft has shown, by a preponderance of the evidence, that claim 46 is unpatentable as obvious over the combination of SQL-92 and Chawathe.

1. *SQL-92*

SQL-92 describes a standard specifying the syntax and semantics of

the SQL database language. Ex. 1105, 1. In particular, SQL-92 describes a table object, which is a multi-set of rows. *Id.* at 12, 29; *see also id.* at 82 (“A <table name> identifies a table.”). Additionally, the SQL-92 table object comprises columns. *Id.* at 12; *see also id.* at 82 (“A <column name> identifies a column.”).

2. Chawathe

Chawathe describes a project for integrating heterogeneous information sources referred to as The Stanford-IBM Manager of Multiple Information Sources (TSIMMIS) project. Ex. 1116, 1. Chawathe states, for the TSIMMIS project, an object exchange model (OEM) is adopted. *Id.* Additionally, a SQL-like query language, OEM-QL, is adopted for requesting OEM objects. *Id.* at 2.

3. Claim 46

We have reviewed Microsoft’s obviousness argument, supporting evidence, and the detailed claim analysis, which reads persuasively all elements of claim 46 onto the teachings of SQL-92 and Chawathe, taken together. Pet. 43–50 (citing Ex. 1105, 1, 12, 28–30, 82, 156, 193, 220, 312–14, 316, 322, 477, 497; Ex. 1116, 1, 2, 5, 6; Ex. 1119 ¶¶ 125, 127–29, 131, 133–35, 138–40, 143–44, 146–51). For instance, we are persuaded that Microsoft has shown by a preponderance of the evidence that the combination of SQL-92 and Chawathe teaches configuring memory according to a logical table, as recited in claim 46. SQL-92 teaches “the descriptors of enabling objects (e.g., tables) are said to *include* the descriptors of enabled objects (e.g., columns or table constraints).” Ex. 1105, 12. Additionally, SQL-92 teaches “[a] table is a multiset of rows.” *Id.*

at 29.

Additionally, we determine that Microsoft has shown by a preponderance of the evidence that the combination of SQL-92 and Chawathe teaches each row and each column including an OID, as recited in claim 46. For instance, SQL-92 teaches assigning a primary key that uniquely identifies a row in a table. Ex. 1105, 32–33, 497. SQL-92 also teaches that each column has an ordinal position. *Id.* at 29. Furthermore, Chawathe teaches a unifying object model, in which each object has a unique variable-length identifier referred to as “Object-ID.” Ex. 1116, 5.

Enfish contends that Microsoft’s motivation-to-combine arguments are insufficient. PO Resp. 42. We, however, are persuaded that Microsoft has set forth a sufficient articulated reason with a rational underpinning to support obviousness. *See KSR Int’l Co. v. Teleflex, Inc.*, 550 U.S. 398, 418 (2007) (citing *In re Kahn*, 441 F.3d 977, 988 (Fed. Cir. 2006)). For instance, Microsoft relies on Dr. Hosking’s Declaration (Ex. 1119 ¶¶ 358–59), which states it would have been obvious to one of ordinary skill in the art to combine the techniques of SQL-92 and Chawathe to use an SQL-like language to query objects because Chawathe suggests combination with SQL-92 to enhance SQL-92 language capabilities. Ex. 1119 ¶ 359 (citing Ex. 1116, 2, 6). Chawathe states “OEM-QL is an SQL-like language extended to deal with labels and object nesting” Ex. 1116, 2; *see also id.* at 6 (“OEM-QL adapts existing SQL-like languages for object-oriented models . . . to OEM.”). We credit Dr. Hosking’s statement as it is consistent with the teachings of Chawathe (*see e.g.*, Ex. 1116, 2, 6).

For the foregoing reasons, Microsoft has shown, by a preponderance of the evidence, that claim 46 of the ’604 patent is unpatentable as obvious

over the combination of SQL-92 and Chawathe.

C. Obviousness of Claims 49 and 50 in view of Visual Basic and Jensen

For the reasons given below, after consideration of the Petition, Enfish's Patent Owner Response, and the evidence cited therein, Microsoft has shown, by a preponderance of the evidence, that each of claims 49 and 50 is unpatentable as obvious over the combination of Visual Basic and Jensen.

1. Jensen

Jensen describes managing information retrieved from a structured database, such as a relational database, by constructing object instances in which each object instance has its own unique object identifier that provides a mapping between the object instance and at least one row in the structured database. Ex. 1106, Abstract. Jensen further describes a pointer that is an object instance attribute containing an address, such as a physical address, of another object instance. *Id.* at 7:33–35. Jensen also describes bidirectional pointers that are set up between a department instance and related employee instances. *Id.* at 10:9–11.

2. Claims 49 and 50

Claims 49 and 50 depend, directly or indirectly, from claim 48. Microsoft has shown by a preponderance of the evidence that Visual Basic describes the additional elements recited in claims 49 and 50. Pet. 50–52 (citing Ex. 1104, PF 69, PG 487, 489, Fig. 20.6; Ex. 1106, 7:29–35, 8:24–29, 8:50–10:17, 10:45–48, 10:62–11:26, Fig. 2, 3; Ex. 1119 ¶¶ 360–369, 377–377).

Enfish contends that Microsoft's motivation-to-combine arguments

are insufficient. PO Resp. 43. Dr. Jagadish states “[d]ue to the dissimilarity of environments between [Visual Basic] and Jensen (a relational database employing unidirectional pointers in an indexing structure vs. linking related objects in an object cache), extensive redesign of the system in [Visual Basic] would be required.” Ex. 2007 ¶ 228. In contrast to Dr. Jagadish’s statement, however, Jensen teaches an object-oriented application for managing information retrieved from a structured database, such as a relational database. Ex. 1106, Abstract.

Enfish further contends that Microsoft does not explain why one of ordinary skill in the art would have been motivated to apply the pointers of Jensen to the indexing structure in Visual Basic. PO Resp. 43 (citing Ex. 2007 ¶ 228). We, however, are persuaded that Microsoft has shown a sufficiently articulated reason with rational underpinning to support obviousness. Microsoft states that one of ordinary skill in the art would have been motivated to combine Jensen with Visual Basic considering Jensen’s benefits of referencing objects through two-way pointers as an improvement in flexibility (Pet. 51 (citing Ex. 1106, 7:29–35)) as compared to Visual Basic’s one-way pointers (Pet. 51 (citing Ex. 1104, PG 487)). Also, Dr. Hosking states that a person of ordinary skill in the art would have been motivated to combine Jensen’s teaching of a pointer with Visual Basic’s disclosure of index objects. Ex. 1119 ¶¶ 366, 369; *see also id.* ¶ 376 (“It was *well known* in the art at the time that Index Objects locate data by maintaining object identification numbers in the internal pointers as disclosed in Jensen” (emphasis added)). The predictable use of familiar prior art elements according to their established functions renders the recited invention obvious. *See KSR*, 550 U.S. at 417.

For the foregoing reasons, we determine that Microsoft has shown, by a preponderance of the evidence, that claims 49 and 50 of the '604 patent are unpatentable as obvious over the combination of Visual Basic and Jensen.

D. Obviousness of Claims 53, 55, 56, and 60 in view of Visual Basic and Salton and Obviousness of Claim 51 in view of Visual Basic, Jensen, and Salton

For the reasons given below, we conclude that Microsoft has not shown that claims 51, 53, 55, 56, and 60 are unpatentable as obvious.

1. Salton

Salton describes information retrieval. Ex. 1107, 7. In particular, Salton describes speeding up a search for information by developing an index. *Id.* at 16. An index of Salton includes values for each key for records in a file. *Id.* at 17.

2. Claims 53, 55, 56, and 60

Enfish contends that Microsoft's motivation-to-combine contentions are insufficient. PO Resp. 48–51. Dr. Jagadish states, "it is my opinion that a POSITA would not have been motivated to import, into [Visual Basic], the teachings in Salton regarding indexing architectures." Ex. 2007 ¶ 239. Dr. Jagadish provides various reasons including a lack of teachings in Visual Basic, difficulties described in Salton, as well as difficulties that would have been known to one of ordinary skill in the art. Ex. 2007 ¶¶ 239–242.

Microsoft discusses reasons for combining Visual Basic and Salton in three places in the Petition. First, Microsoft states that one of ordinary skill in the art would have been motivated to combine Salton with Visual Basic to employ the indexing and text analysis techniques taught by Salton in the

object database systems of Visual Basic. Pet. 53 (citing Ex. 1119 ¶ 424). We are not persuaded that this is a sufficient rationale for combining Visual Basic and Salton, because Microsoft's citation does not provide adequate support. Microsoft cites only to Dr. Hosking, who states that Visual Basic and Salton are in the same field and provides a conclusion that one of ordinary skill in the art would have been motivated to combine Visual Basic and Salton. Ex. 1119 ¶ 424. Additionally, in its contentions regarding other challenged claims, Microsoft relies on Visual Basic for describing indexing. *See* Pet. 40 (Ex. 1104, PF 37, 38). In light of these other contentions, Microsoft does not explain sufficiently why one of ordinary skill in the art would have been motivated to employ Salton's indexing techniques with Visual Basic.

Second, Microsoft states, “[o]ne of ordinary skill in the art would combine Salton with [Visual Basic] to search for key words and create [an] index for those key words with pointers to the logical cells in [Visual Basic], because both Salton and [Visual Basic] discuss information search and retrieval using computer databases.” Pet. 56 (citing Ex. 1119 ¶ 457); *see also* Pet. 56 (“One of ordinary skill in the art would know to combine Salton and [Visual Basic] because both describe storing data in databases, allowing users to query the data, and using indexes to facilitate those queries”). Microsoft's contentions are insufficient because they are based on solely the references teaching the same techniques. Dr. Hosking states, “[o]ne having ordinary skill in the art would know to combine what's described in Salton with what's described in [Visual Basic] because they both deal with solving the problem of locating key words in a database.” Ex. 1119 ¶ 442; *see also id.* at ¶ 457 (referencing claim element [25C]). Again, Dr. Hosking's

conclusion is based solely on the references being in the same field. He fails to provide sufficient rational underpinning to explain why a person of ordinary skill would have combined the references.

Third, Microsoft states “[o]ne [of] ordinary skill in the art would know to add the weighting techniques described in Salton to [Visual Basic] to weigh key words and retrieve[] cells.” Pet. 58 (citing Ex. 1119 ¶¶ 469–477). In the cited portions of the Hosking Declaration, Dr. Hosking states that “[o]ne having skill in the art would know to combine [Visual Basic] with weigh[t]ing techniques of Salton because both have to do with providing access to data in a database through a search.” Ex. 1119 ¶ 473. Although Dr. Hosking indicates that weighting increases the value of search results (*id.* ¶¶ 469–477), he does not indicate that increasing value using weighting is a reason for combining the technology of Salton with Visual Basic to arrive at the claims, which do not recite weighting.

As discussed above, Dr. Jagadish disagrees with Dr. Hosking. Ex. 2007 ¶¶ 239–242. In its Reply, Microsoft does not address Enfish’s contentions or the testimony provided by Dr. Jagadish. Pet. Reply 11–12.

3. Claim 51

With respect to claim 51, Microsoft, relying on Dr. Hosking, further states that “[o]ne of ordinary skill in the art would have been motivated to combine Salton with [Visual Basic] and Jensen to employ the indexing and text analysis techniques taught by Salton in the object database systems of [Visual Basic] and Jensen to improve data searching and retrieval efficiency.” Pet. 54 (citing Ex. 1119 ¶ 383). Dr. Hosking, however, states that the references relate to data managing and then states his conclusion that one of ordinary skill in the art would have been motivated to combine them.

Ex. 1119 ¶ 383.

4. Conclusion

We determine that Dr. Hosking's statements are not a sufficient rationale for combining these references because he bases his conclusions on simple statements that the references are in the same field. Microsoft does not provide sufficient additional reasons, such as expert testimony that the combinations of prior art references are of familiar elements according to known methods that yield no more than predictable results. We, therefore, conclude that Microsoft has not demonstrated that the teachings would have been combined by a person of ordinary skill.

For the foregoing reasons, we determine that Microsoft has not set forth a sufficient articulated reason with a rational underpinning to support a showing of obviousness by a preponderance of the evidence. We, therefore, conclude that Microsoft has not shown that claims 51, 53, 55, 56, and 60 of the '604 patent are unpatentable as obvious.

E. Obviousness of Claim 52 in view of Visual Basic and Smith '162

For the reasons given below, Microsoft has not shown that claim 52 is obvious.

1. *Smith '162*

Smith '162 describes an object-oriented document management and production system. Ex. 1108, Abstract. Stored objects are organized, accessed, and manipulated through a database management system. *Id.* For example, documents and folders can be represented as objects. *Id.* at 3:29–35. The hierarchically superior folder object pointers specify documents. *Id.* at 8:19–20.

2. *Claim 52*

Enfish contends that Microsoft's motivation-to-combine arguments are insufficient. PO Resp. 46. We are not persuaded that Microsoft has set forth a sufficient articulated reason with a rational underpinning to support a determination of obviousness. Microsoft states that one of ordinary skill in the art would have been motivated to combine the folder and document objects of Smith '162 with the objects disclosed in Visual Basic for the benefits of providing varying data architectures in the database. Pet. 54–55 (citing Ex. 1119 ¶ 393). Microsoft, however, relies on Dr. Hosking, who states that one of ordinary skill in the art would have been motivated to combine Visual Basic and Smith '162 because they “address the same technical issues and disclose closely related subject matters.” Ex. 1119 ¶ 393. Although Dr. Hosking mentions that Smith '162 relates to managing documents and folders, he does not testify persuasively regarding the benefits of providing varying data architectures in connection with combining Visual Basic and Smith '162. Microsoft does not offer persuasive additional contentions or evidence supporting the asserted combination of Visual Basic and Smith '162.

For the foregoing reasons, Microsoft has not shown that claim 52 of the '604 patent is unpatentable as obvious.

F. Secondary Considerations

We note that factual inquiries for an obviousness determination include secondary considerations based on evaluation and crediting of objective evidence of nonobviousness. *Graham v. John Deere Co.*, 383 U.S. 1, 17 (1966). Notwithstanding what the teachings of the prior art would have suggested to one with ordinary skill in the art at the time of the

invention, the totality of the evidence submitted, including objective evidence of nonobviousness, may lead to a conclusion that the claimed invention would not have been obvious to one with ordinary skill in the art. *In re Piasecki*, 745 F.2d 1468, 1471–72 (Fed. Cir. 1984). However, to accord substantial weight to objective evidence requires the finding of a nexus between the evidence and the merits of the claimed invention. *In re GPAC Inc.*, 57 F.3d 1573, 1580 (Fed. Cir. 1995); *see also In re Huang*, 100 F.3d 135, 140 (Fed. Cir. 1996) (“success is relevant in the obviousness context only if there is proof that the sales were a direct result of the unique characteristics of the claimed invention.”).

Enfish contends that the claimed invention received industry accolades, including praise from Microsoft, satisfied a long-felt need, resulted in success where others had failed, as well as commercial success. PO Resp. 53–57. Enfish points to features of claim 24, which Enfish contends are the features of the claimed invention that resulted in the objective indicia of success to which Enfish refers. *Id.* at 53–54. Claim 24 of the ’604 patent, however, is challenged on the basis that it is anticipated by Visual Basic. Secondary considerations do not weigh into determinations regarding anticipation. *Cohesive Techs., Inc. v. Waters Corp.*, 543 F.3d 1351, 1364 (Fed. Cir. 2008).

Independent claim 46, which is challenged on the basis of obviousness, recites some of the same features as claim 24. We, however, determine that SQL-92 teaches all of the features of claim 46, including OIDs, except SQL-92 does not teach explicitly that certain OIDs have variable lengths. Microsoft contends that SQL-92 should be combined with Chawathe because Chawathe teaches each object having a unique variable-

length identifier referred to as “Object-ID.” Pet. 50 (citing Ex. 1116, 5). Enfish does not claim to have invented a variable length for an OID. Thus, Enfish has not shown a nexus between any of the accolades or successes it says occurred and the use of the claimed variable length OIDs.

Because we conclude that Microsoft has not shown that claims 51–53, 55, 56, and 60 of the ’604 patent are unpatentable for other reasons, the only other claims that we need to evaluate are dependent claims 49 and 50 of the ’604 patent. Microsoft asserts that claims 49 and 50 are obvious over the combination of Visual Basic with a reference that teaches referencing objects through two-way pointers. Pet. 51 (citing Ex. 1106, 7:29–35). Enfish has not shown a nexus between any of the accolades or successes it says occurred and the use of two-way pointers.

Enfish additionally submits evidence that Enfish states shows that Microsoft failed at developing a suitable search engine. PO Resp. 58–59. We are not persuaded by this argument. The statements submitted by Enfish are not tied sufficiently to any claim at issue in this proceeding. Instead, the statements broadly refer to search engines. None of the statements reference a variable length for an OID or two-way pointers.

Enfish further submits evidence that Enfish alleges shows commercial success. PO Resp. 59–60. In addition to the shortcomings discussed above, the proffered evidence is further deficient for the following reasons. Enfish contends that the evidence supports that its product was “well received,” 75,000 users downloaded Enfish’s software, and the functionality of the software resulted in a collaborative effort. See PO Resp. 59–60 (citing Ex. 2024; Ex. 2025, 2; Ex. 2030). Enfish’s evidence, however, does not indicate that users paid for or actually used the downloaded software. The evidence

also does not indicate how the number of downloads illustrates commercial acceptance, for example, as compared to downloads of other software at the time. Additionally, a planned collaborative effort does not indicate the results of the collaboration. Enfish's evidence does not establish commercial acceptance or financial success. *See In re Fielder*, 471 F.2d 640, 644 (CCPA 1973). Thus, Enfish's evidence is not sufficient to show commercial success.

We, therefore, determine that features of variable length OIDs and pointers would have been predictable variations within the technical grasp of a person of ordinary skill in the art, and are not persuaded by the objective evidence that claims 46, 49, and 50 of the '604 patent are not obvious.

G. Motion to Correct Patent Owner Response

After institution of trial, Enfish timely filed a Patent Owner Response (Paper 26), along with the Jagadish Declaration (Ex. 2007), which Enfish later corrected (Paper 27). On September 16, 2014, Enfish filed an unopposed motion to file a second corrected Patent Owner Response and a corrected Jagadish Declaration, which Enfish contends correct only typographical errors and erroneous citations. Paper 40. We grant Enfish's September 16, 2014 request.

H. Joint Stipulation

On November 14, 2014, the parties filed a joint stipulation requesting that we expunge confidential versions of exhibits 2049–2058 and 2060–2065. Paper 57. The parties contend that Microsoft withdraws its motion to seal (Paper 30) provided that we expunge the confidential versions. Paper 55. Microsoft agrees that the sealed version of Exhibit 2059 may be

unsealed. *Id.* We hereby grant the motion and expunge only confidential versions of exhibits 2049–2058 and 2060–2065.

I. Motion to Exclude

On November 3, 2014, Microsoft filed a motion to exclude Exhibit 2071, the Declaration of Dr. Sharad Mehrotra (“Mehrotra Declaration”), and two paragraphs of the Declaration of Louise Wannier (“Wannier Declaration,” Exhibit 2077 ¶¶ 32, 33). Paper 50.

Regarding the Mehrotra Declaration, we agree with Microsoft’s assertion that Dr. Mehrotra provides only conclusory opinions and, therefore, we give his Declaration no weight and do not rely on it in this Decision. 37 C.F.R. § 42.65(a). Because Microsoft has not argued persuasively any other reason to exclude the Mehrotra Declaration, we dismiss Microsoft’s request to exclude it as moot.

Regarding the Wannier Declaration, we disagree with Microsoft that “Patent Owner has no basis to file the Wannier Declaration as supplemental evidence because Microsoft has not moved to exclude the Armon Declaration.” Paper 50, 4–5. Patent Owner is entitled to submit supplemental evidence in response to Microsoft’s objection. 37 C.F.R. § 42.64(b)(2). Microsoft further contends that the Wannier Declaration inserts untimely, conclusory, and improper technical opinions. Paper 50, 5. Patent Owner contends that paragraphs 32 and 33 do not exceed the scope because they are submitted to support admissibility. Paper 60, 3–4. We agree with Microsoft that the Wannier Declaration provides conclusory technical opinions, and, therefore, give the technical opinions in paragraphs 32 and 33 of her Declaration no weight. Because Microsoft has not argued persuasively any other reason to exclude paragraphs 32 and 33 of

the Wannier Declaration, we dismiss Microsoft's request to exclude it as moot.

IV. CONCLUSION

We conclude that Microsoft has demonstrated by a preponderance of the evidence that (1) claims 47, 48, and 54 of the '604 patent are anticipated by Visual Basic, (2) claim 46 of the '604 patent is obvious over the combination of SQL-92 and Chawathe, and (3) claims 49 and 50 of the '604 patent are obvious over the combination of Visual Basic and Jensen.

We further conclude that Microsoft has not shown that claims 51–53, 55, 56, and 60 of the '604 patent are unpatentable as obvious. In addition, we terminate this proceeding with respect to claims 16–26 and 30 under 37 C.F.R. § 42.72. Claims 27–29 and 57–59 are not at issue in this trial.⁴

This is a final written decision of the Board under 35 U.S.C. § 318(a). Parties to the proceeding seeking judicial review of this decision must comply with the notice and service requirements of 37 C.F.R. § 90.2.

⁴ In the Decision to Institute, we declined to institute an *inter partes* review of claims 27–29 and 57–59 because we were not persuaded that Petitioner had shown that there was a reasonable likelihood of prevailing on its challenges to these claims. Inst. Dec. 2, 25–27.

V. ORDER

For the reasons given, it is

ORDERED that claims 46–50 and 54 of U.S. Patent No. 6,151,604 are determined by a preponderance of the evidence to be unpatentable;

FURTHER ORDERED that this proceeding is TERMINATED, under 37 C.F.R. § 42.72, with respect to claims 16–26 and 30;

FURTHER ORDERED Enfish’s motion to file a second corrected Patent Owner Response and a corrected Jagadish Declaration (Paper 40) is GRANTED;

FURTHER ORDERED that Microsoft’s motion to exclude (Paper 50) is DISMISSED;

FURTHER ORDERED that confidential versions of Exhibits 2049–2058 and 2060–2065 are expunged;

FURTHER ORDERED Microsoft’s motion to seal is DISMISSED; and

FURTHER ORDERED that because this is a final written decision, parties to the proceeding seeking judicial review of the decision must comply with the notice and service requirements of 37 C.F.R. § 90.2.

Case IPR2013-00563

Patent 6,151,604

For PETITIONER:

Chun M. Ng

Amy E. Simpson

Chad Campbell

Theodore H. Wimsatt

PERKINS COIE LLP

CNg@perkinscoie.com

ASimpson@perkinscoie.com

CCampbell@perkinscoie.com

TWimsatt@perkinscoie.com

For PATENT OWNER:

Frank Pietrantonio

Orion Armon

COOLEY LLP

fpietrantonio@cooley.com

oarmon@cooley.com

zpatdcdocketing@cooley.com