

NOTE: This disposition is nonprecedential.

**United States Court of Appeals
for the Federal Circuit**

MICROSOFT CORPORATION,
Appellant

v.

ENFISH, LLC,
Cross-Appellant

2015-1734, 2015-1736, 2015-1737, 2015-1738, 2015-1739,
2015-1740, 2015-1741, 2015-1742, 2015-1816, 2015-1817,
2015-1818, 2015-1819

Appeals from the United States Patent and Trade-
mark Office, Patent Trial and Appeal Board in Nos.
IPR2013-00559 through -00563.

Decided: November 30, 2016

CHAD S. CAMPBELL, Perkins Coie LLP, Phoenix, AZ,
argued for appellant. Also represented by THEODORE H.
WIMSATT; DAN L. BAGATELL, Hanover, NH; AMY
ELIZABETH SIMPSON, San Diego, CA.

ORION ARMON, Cooley LLP, Broomfield, CO, argued
for cross-appellant. Also represented by JAMES P.

BROGAN, BRITTON F. DAVIS, JANNA FISCHER, PETER SAUER;
FRANK V. PIETRANTONIO, JENNIFER VOLK, Reston, VA.

Before LOURIE, O'MALLEY, and TARANTO, *Circuit
Judges.*

TARANTO, *Circuit Judge.*

Enfish, LLC owns U.S. Patent Nos. 6,151,604 and 6,163,775, which describe and claim methods and apparatuses for storing and retrieving data using assertedly improved database techniques. In September 2013, Microsoft Corp. filed five petitions for inter partes review of the '604 and '775 patents. After instituting review based on those petitions, the Patent Trial and Appeal Board concluded, in five final written decisions, that some, but not all, of the challenged claims were unpatentable. Microsoft and Enfish cross-appeal from selected adverse portions of the Board's decisions. We affirm.

I

We have previously discussed the '604 and '775 patents in *Enfish, LLC v. Microsoft Corp.*, 822 F.3d 1327, 1329–34 (Fed. Cir. 2016). The patents issued from continuations of U.S. Patent Application No. 08/383,752 and share a specification. The patents also share much claim language. Accordingly, we cite only the relevant passages in the '604 patent, except where the '604 and '775 patents meaningfully differ, and we will generally use the singular “claim” to refer to corresponding claims that are materially the same.

The patents describe and claim techniques for storing data in a table consisting of rows and columns. '604 patent, col. 6, lines 36–41. Each row and column has an object identification number (OID) that identifies the row or column. *Id.*, col. 6, lines 42–49. To facilitate searching and synchronization of substantively related portions of a

table (such as “employed by” and “employees”), certain columns may be defined by rows in the database. *Id.*, Fig. 3; *id.*, col. 7, lines 10–23.

In September 2013, Microsoft filed five petitions for inter partes review of the ’604 and ’775 patents. In March 2014, the Board instituted review of claims 1–26, 30–56, and 60 of both patents. IPR2013-559, -560, -561, -562, -563. In March 2015, the Board found unpatentability, based on anticipation or obviousness, of claims 31, 41, 46–50, and 54 of both patents; claim 42 of the ’604 patent; and claim 45 of the ’775 patent. The Board rejected Microsoft’s unpatentability challenges to claims 32–40, 43–44, 51–53, 55–56, and 60 of both patents; claim 45 of the ’604 patent; and claim 42 of the ’775 patent. As to claims 1–26 and 30 of both patents—which are not at issue before us—the Board terminated proceedings after concluding that those claims were not amenable to construction.

As detailed *infra*, Enfish appeals certain Board determinations of unpatentability, and Microsoft appeals certain determinations of patentability. Both appeals are authorized by 35 U.S.C. §§ 141(c) and 319. We have jurisdiction under 28 U.S.C. § 1295(a)(4)(A).

II

Enfish argues as cross-appellant that the Board erred in its unpatentability rulings as to claims 31, 42, 48–50, and 54 of the ’604 and ’775 patents; claim 42 of the ’604 patent; and claims 41 and 46 of the ’775 patent. Enfish rests its appeal on a claim-construction contention. It contends that the Board erred in construing “object identification number (OID)” as “an array of bits that define an object.” Enfish contends that an OID must “identify,” rather than “define,” an object and, in addition, must be “unique,” “system-generated,” and “immutable.” We affirm the Board’s construction. We therefore affirm the Board’s decisions as to the claims Enfish appeals.

A

We review the Board’s decisions under the standards set forth in the Administrative Procedure Act, 5 U.S.C. § 706. *Dickinson v. Zurko*, 527 U.S. 150, 154 (1999); *Pride Mobility Prods. Corp. v. Permobil, Inc.*, 818 F.3d 1307, 1313 (Fed. Cir. 2016). We must set aside the Board’s actions if they are “arbitrary, capricious, an abuse of discretion, or otherwise not in accordance with law” or “unsupported by substantial evidence.” 5 U.S.C. § 706(2). We review the Board’s legal conclusions without deference and its factual findings for substantial evidence. *In re Gartside*, 203 F.3d 1305, 1316 (Fed. Cir. 2000).

Claim construction is a question of law based on underlying questions of fact. *Teva Pharm. USA, Inc. v. Sandoz Inc.*, 135 S. Ct. 831, 840–42 (2015). Because the ’604 and ’775 patents expired while this appeal was pending, the governing construction is not the broadest reasonable interpretation, given that there is no possibility of claim amendment. *See In re CSB-System Int’l, Inc.*, 832 F.3d 1335, 1340–42 (Fed. Cir. 2016). We adhere to the normal judicial standards of claim construction. *See Phillips v. AWH Corp.*, 415 F.3d 1303, 1312–15 (Fed. Cir. 2005) (en banc). We note, however, that we see no basis on which the broadest-reasonable-interpretation approach would lead to a different result. *Cf. CSB-System Int’l*, 832 F.3d at 1341 (“In many cases, the claim construction will be the same under [both] standards.”).

B

We reject Enfish’s challenges to the Board’s construction of “OID.” As an initial matter, Enfish has forfeited any argument that the Board should have required an OID to “identify” an object—instead of, or in addition to, “defining” the object. Before the Board, Enfish argued that “OID” should be construed as “a unique, immutable, and system-generated value that identifies an object.” J.A. 970. To support that construction, Enfish presented

evidence that an “OID” was “unique,” “system-generated,” and “immutable.” J.A. 970–73. Enfish did not argue that “defining” and “identifying” were meaningfully different. Enfish’s “identifying” argument is not merely a new or additional argument supporting the construction it urged before the Board, but an argument for a construction different from any Enfish clearly and timely urged before the tribunal we review. Such arguments are generally impermissible on appeal as having been forfeited, and we see no basis for disregarding the forfeiture in the present setting. *See, e.g., Digital-Vending Servs. Int’l, LLC v. Univ. of Phoenix, Inc.*, 672 F.3d 1270, 1273 (Fed. Cir. 2012); *Broadcom Corp. v. Qualcomm Inc.*, 543 F.3d 683, 694 (Fed. Cir. 2008).

As to Enfish’s contention that an “OID” must be “unique” in the sense that each object in the database has a different OID, we agree with the Board’s rejection of that contention. Because the term “unique” does not appear in the claims, Enfish relies entirely on the specification to support its proposed construction. But the specification does not warrant Enfish’s construction.

Contrary to Enfish’s position, the specification affirmatively illustrates that objects may share the same OID in some cases. In particular, Figure 3 depicts a table in which row 136 and column 126 have the same OID (namely, 1019). Enfish argues that Figure 3 does not contradict its proposed construction because row 136 and column 126 are actually the same object. The specification is to the contrary. It states a lesser relationship than identity. In particular, it states that “[e]ach column has an associated column definition,” which is not the same as the column, but instead “determines the properties of the column, such as the domain of the column, the name of the column, whether the column is required and other properties that may relate to the column.” ’604 patent, col. 7, lines 10–14. It is the associated “column definition,” not the column, that “is stored as a record [row] in

the table 100 of FIG. 3. For example, the ‘Employed By’ column 126 has a corresponding row 136.” *Id.*, col. 7, lines 16–18. The row is not the same object as the column; indeed, Figure 3 shows different content of the row and the corresponding column.

The specification passages on which Enfish relies do not contradict what Figure 3 shows or otherwise establish that each object in the database must have an OID not used by any other object. Those passages describe various “OIDs” as “unique,” but do not define uniqueness to require what Enfish urges—not just lack of repetition of OIDs among rows (often called records), and lack of repetition of OIDs among columns, but lack of repetition of OIDs between rows and columns. The passages need not be read, and in light of Figure 3 should not be read, as requiring any more than that the system be capable of retrieving objects using OIDs. *Id.*, col. 1, lines 65–67, through col. 2, line 1 (“Key features of the object oriented [database] model” include that “each item has a unique system-generated object identification number that can be used for exact retrieval.”); *id.*, col. 6, lines 42–45 (“Each row is assigned a unique object identification number (OID) stored in column 120 and each column also is assigned a unique OID, indicated in brackets and stored in row 108.”); *id.*, col. 8, lines 7–9 (“[T]he system must generate a unique OID when columns and rows are formed.”). We conclude that the Board properly rejected Enfish’s “unique” construction.

For similar reasons, we conclude that the Board properly rejected Enfish’s contention that an “OID” must be “system-generated” rather than created by, say, a user. There is no arguable basis in the claim language for such a requirement. Enfish relies for its contention on several of the same passages just discussed regarding its “unique” contention. But those passages describe only certain prior-art databases, *id.*, col. 1, lines 65–67, and certain embodiments, *id.*, col. 8, lines 7–9. Such specification

material does not meet the standard for establishing a requirement that lacks any basis in the claim language.

Finally, we agree with the Board that an “OID” does not need to be “immutable.” Neither “immutable” nor a synonym appears in the claims or specification (or any prosecution history called to our attention). Before the Board, Enfish relied on expert testimony and other extrinsic evidence to support its argument that an “OID” must be immutable. Microsoft introduced contrary testimony and evidence to rebut Enfish’s position. Given the absence of intrinsic evidence, and conflicting extrinsic record, the Board was entitled to reject Enfish’s proposed construction.

III

For its part, Microsoft argues in the principal appeal that the Board erred in upholding claims 32, 36–37, 55–56, and 60 of both patents; claim 43 of the ’604 patent; and claim 42 of the ’775 patent. Specifically, Microsoft contends that the Board should have found those claims unpatentable for anticipation or obviousness. Anticipation is a question of fact. *See Bischoff v. Wethered*, 76 U.S. (9 Wall.) 812, 814–15 (1869); *In re Gleave*, 560 F.3d 1331, 1334–35 (Fed. Cir. 2009). Obviousness is a question of law based on underlying questions of fact, including the scope and content of the prior art, the differences between the prior art and the claimed invention, the level of ordinary skill in the field of the invention, and any relevant objective indicia of nonobviousness. *See Graham v. John Deere Co.*, 383 U.S. 1, 17–18 (1966); *Randall Mfg. v. Rea*, 733 F.3d 1355, 1362 (Fed. Cir. 2013). We reject Microsoft’s contentions.

A

Claim 31 of the ’604 and ’775 patents recites a “method for storing and retrieving data in a computer memory,” in which the “memory” is configured “according to a

logical table” consisting of “logical rows” and “logical columns.” ’604 patent, col. 23, lines 15–31. Each “logical row” and “logical column” includes an OID to “identify” that row or column. *Id.* At least one of the “logical rows” has an OID equal to the OID of a “corresponding” logical column, and at least one of the “logical rows” “includes column information defining each of said logical columns.” *Id.* Claim 32, dependent on claim 31, adds that the “logical column information defines one of said logical columns to contain information for enabling determination of OIDs from text entry.” *Id.*, col. 23, lines 32–35.

European Patent Application 03 36580 (Chang) discloses a relational database, which consists of multiple tables. Chang discloses the use of two tables, SYS.TABLES and SYS.COLUMNS, to store data about tables and columns in the database, respectively. Chang, col. 5, line 54, through col. 7, line 19; *id.*, col. 10, line 15, through col. 11, line 10; *id.*, Figs. 2–3, 6–7. Each row in SYS.TABLES includes a “packed description,” which includes column information. *Id.*, col. 10, line 15, through col. 11, line 10; *id.*, Figs. 6–7. Chang also discloses a table, SYS.INDEXES, which is used to index columns for searching. *Id.*, col. 7, line 20, through col. 8, line 3; *id.*, Fig. 4.

The Board rejected Microsoft’s contention that claim 32 is anticipated by Chang. The Board concluded that Microsoft did “not explain how OID determination by text searching as recited in claim 32 would be conducted on such stored information and definitions in those particular tables, or even that SYS.TABLES or SYS.COLUMNS are indexed to provide a search function.” J.A. 27; 112.

Microsoft appeals that ruling on two grounds. It argues that the Board erred by construing “text entry” to require “text searching” and, alternatively, that SYS.TABLES’s “packed descriptions” contain index information that enables the determination of OIDs, even

if “text entry” requires “text searching.” Neither argument has merit.

Contrary to Microsoft’s contention, the Board did not construe “text entry” to require “text searching.” It used “search” language because Microsoft itself used that language to describe the two asserted Chang disclosures of this limitation (one using indexes, the other column information). First, Microsoft argued that SYS.TABLES’s packed description collects keyword indexes (for each column), which “may be searched,” such “searching being “meant to return a particular row or rows” from which OIDs may be determined. J.A. 234, 370. Second, Microsoft argued that Chang discloses “primary key columns,” which contain information uniquely identifying each column. J.A. 234, 371. Microsoft asserted that SYS.COLUMNS and SYS.TABLES’s packed description store information “defining the column from which to obtain an OID in the form of a primary key.” *Id.* Microsoft urged that Chang thus discloses both “index and column defining information from which text entry searches can be used to obtain row OIDs.” *Id.* The Board rejected those arguments as unsupported by the evidence. The Board’s statement that Microsoft did “not explain” how Chang disclosed “text searching” reflected Microsoft’s formulation, not the Board’s adoption of a contested construction that “text entry” required “text searching.”

Additionally, substantial evidence supports the Board’s finding that Chang does not disclose that SYS.TABLES is indexed to enable the determination of OIDs from text entry. Figure 4 of Chang, cited in Microsoft’s Petition, refers to SYS.INDEXES, not SYS.TABLES or SYS.COLUMNS. Although Figure 6 indicates that SYS.TABLES’s packed descriptions contain index information, Chang does not state that those packed descriptions are actually used in index searching. To bridge the gap, Microsoft relied on assertions by Dr. Hosking. In response, Enfish argued that SYS.INDEXES,

not SYS.TABLES and SYS.COLUMNS, practiced indexing and that there was “absolutely no indication in Chang that either of SYS.TABLES or the SYS.COLUMNS table in Chang are indexed” or “indexed to determine OIDs.” J.A. 1000. The Board was entitled to credit Enfish’s response to Dr. Hosking’s analysis.

On appeal, Microsoft argues that Chang describes validation, which requires the system to validate user operations, *e.g.*, retrieving information, against the column definitions contained in SYS.TABLES and SYS.COLUMNS before executing those operations. But Microsoft did not meaningfully present that argument to the Board. We will not disturb the Board’s ruling on such a ground. *See Wallace v. Dep’t of Air Force*, 879 F.2d 829, 832 (Fed. Cir. 1989) (An “issue must be raised with sufficient specificity and clarity that the tribunal is aware that it must decide the issue, and in sufficient time for the agency to do so,” in order for the issue to be preserved for appeal.).

B

Claim 36 of the ’604 and ’775 patents depends on claim 31 and adds the requirements that (1) “at least one of said plurality of logical rows includes information defining the type of a different logical row” and (2) “at least one of said plurality of logical rows includes a logical cell that contains a pointer to said logical row including logical row type information.” ’604 patent, col. 23, lines 47–53. In rejecting Microsoft’s obviousness challenge to claim 36, a challenge based on Chang and U.S. Patent No. 5,201,046 (Goldberg), the Board found that Chang relies on two different tables, SYS.TABLES and SYS.COLUMNS, not a single table, to satisfy the “different logical row” requirement. Microsoft does not dispute that finding about Chang. But it contends that the Board erred in construing “different logical row” in claim 36 to

require that the defined and defining rows be in the same table. We disagree.

Both the claims and the specification support the Board's single-table construction. Earlier this year, we held that claim 31, which configures memory "according to a logical table," involved a single logical table. *Enfish*, 822 F.3d at 1340–43. Neither claim 36 nor any other relevant claim suggests the use of a second table. Microsoft argues that the '604 patent specification does not require a different logical row to be in the same table. But as we explained in *Enfish*, the specification "makes clear" that the invention claims a "single" table. *Id.* at 1340–43. Microsoft does not present any arguments that undermine that conclusion.

The prosecution history also does not support Microsoft's construction. Microsoft argues that during prosecution of the application that issued as U.S. Patent No. 5,729,730, the parent of the '604 and '775 patents, Enfish acquiesced in the examiner's rejection of claim 6—worded like claim 36 at issue here—over a prior-art reference (U.S. Patent No. 5,421,012 (Khoyi)) that discloses a multi-table invention. The record does not support Microsoft's argument. The cited claim 6 issued without amendment after its base claim, claim 1, was amended to address a different reference. '730 patent, col. 20, lines 62–67.

C

Claim 37 of the '604 and '775 patents depends on claim 31 and adds the requirement that "at least one of said logical columns defines logical cells that include a plurality of pointers to other logical columns within the same record, said pointers indicating those logical columns within the same record that contain defined values." '604 patent, col. 23, lines 54–58. Claim 42 of the '775 patent recites a similar method, using similar language. '775 patent, col. 24, lines 58–62. The Board

rejected Microsoft's obviousness challenge to those claims 37 and 42, a challenged based on Chang and U.S. Patent No. 5,463,724 (Anderson). In so ruling, the Board found that Anderson does not disclose "pointers." Microsoft's challenge to the Board's ruling on claims 37 and 42 rests on its challenge to that finding. We reject that challenge.

Before the Board, Microsoft argued that Anderson, which describes a spreadsheet program, discloses "pointers" in the form of references, in a formula in one cell of a spreadsheet, to other cells in the spreadsheet. In response, Enfish's expert, Dr. Jagadish, quoting several C++ and Java references as evidence of the usage of "pointer," declared that Anderson does not disclose a "pointer" because the ordinary and customary meaning of that phrase was "a variable that stores the address where another object resides" and that the spreadsheet examples of Anderson would not have been understood as "pointers." J.A. 7882–83. The Board credited Dr. Jagadish's declaration in finding that Anderson's spreadsheet-cell references were not pointers. Microsoft has not identified contrary evidence that precludes the Board's determination, viewed either as an extrinsic-evidence-based claim construction or as an application of the ordinary meaning to the Anderson reference.

D

Like claim 31, claim 42 of the '604 patent recites a "method for storing and retrieving data in a computer memory," which involves "configuring" that memory into a "logical table" with "logical rows" and "logical columns." '604 patent, col. 24, lines 22–40. Unlike claim 31, claim 42 also involves "searching" the "table" for a "pointer" that defines a "logical column." *Id.* Claim 43 depends on claim 42 and adds the requirement that "at least one of said logical columns defines logical cells that include a plurality of pointers to other logical columns within the same record, said pointers indicating those logical columns

within the same record that contain defined values.” *Id.*, col. 24, lines 41–45. The Board rejected Microsoft’s obviousness challenge to claim 43, a challenge based on Chang and U.S. Patent No. 5,226,158 (Horn). In so ruling, it concluded that Microsoft’s arguments relied on “at least two different tables in Chang, SYS.TABLES and SYS.COLUMNS,” rather than a “single table.” J.A. 31. On appeal, Microsoft argues that it relied on SYS.TABLES only. We find no reversible error by the Board in this ruling.

The Board reasonably interpreted Microsoft’s argument as relying on both SYS.TABLES and SYS.COLUMNS. That is one natural reading of the relevant paragraph of Microsoft’s Petition on claim 43, even if it is not the only possible reading. J.A. 241–42. Microsoft’s Reply did not clarify its position. To the extent that Microsoft intended to argue that SYS.TABLES alone disclosed claim 43’s additional limitations, it should have done so with greater clarity. See *Wallace*, 879 F.2d at 832.

E

Claims 55–56 and 60 of the ’604 and ’775 patents depend, directly or indirectly, on claim 54, which recites a method for “storing and retrieving data in a computer memory” comprising two steps: (1) “configuring” that “memory according to a logical table,” consisting of “logical rows” and “logical columns,” and (2) “indexing” the “data stored” in that “table.” ’604 patent, col. 26, lines 1–15. Claims 55–56 and 60 add requirements to the indexing step, including “searching” the “table” for a “key word,” “creating an index record” of that “key word,” “locating” that “index record” in response to user queries, and “indexing external documents.” *Id.*, col. 26, lines 16–28, 42–44. The Board rejected Microsoft’s obviousness challenge to claims 55–56 and 60, a challenge based on Microsoft Corp., *Visual Basic Programming Systems for*

Windows Version 3.0 (1993) (Visual Basic); and Gerald Salton & Michael J. McGill, *Introduction to Modern Information Retrieval* (1983) (Salton). The Board found that Microsoft did not present a “sufficient articulated reason with rational underpinning” to support its contention that a person of ordinary skill would have been motivated to combine those references. J.A. 79, 186. We agree.

Whether it would have been obvious for a person of ordinary skill in the art to combine two or more references is a flexible inquiry. A factfinder “need not seek out precise teachings directed to the specific subject matter of the challenged claim” and “can take account of the inferences and creative steps that a person of ordinary skill in the art would employ.” *KSR Int’l Co. v. Teleflex. Inc.*, 550 U.S. 398, 418 (2007). Nevertheless, obviousness cannot be established “by mere conclusory statements; instead, there must be some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness.” *Id.* (quoting *In re Kahn*, 441 F.3d 977, 988 (Fed. Cir. 2006)). And “references to ‘common sense’ . . . cannot be used as a wholesale substitute for reasoned analysis and evidentiary support.” *Arendi S.A.R.L. v. Apple, Inc.*, 832 F.3d 1355, 1362 (Fed. Cir. 2016).

Before the Board, Microsoft argued that Visual Basic disclosed “configuring,” “indexing,” “searching,” and “querying,” while Salton disclosed other requirements present in claims 55–56 and 60. Visual Basic describes a general-purpose programming language that enables programmers to create relational databases. In particular, Visual Basic discloses a Table object that represents a logical table consisting of rows and columns. A Table can be indexed using an Index object. According to Visual Basic, indexes can “increase the speed” with which the programmer or user can “access information.” J.A. 7564. Salton describes information retrieval, including the use of indexes. Like Visual Basic, Salton discloses that index-

es “may be used to speed up a search for information.” J.A. 7653. As described, an index includes values for each key in each record.

Although Salton and Visual Basic arguably disclose every limitation in claims 55–56 and 60, the Board correctly concluded that Microsoft did not articulate a sufficient motivation to combine. With respect to claims 55 and 56, Microsoft gave no reason for the motivation of a person of ordinary skill to combine Visual Basic and Salton except that the references were directed to the same art or same techniques, *viz.*, that they “discuss[ed] information search and retrieval using computer databases” or “describe[d] storing data in databases, allowing users to query the data, and using indexes to facilitate those queries.” J.A. 319. With respect to claim 60, Microsoft asserted that a person of ordinary skill would have been motivated to combine the references and did not elaborate. The Board did not err in finding those assertions inadequate to show persuasively that a relevant skilled artisan would have been motivated to combine the references.

Nor did the Board err in finding insufficient the declarations of Microsoft’s relevant expert, Dr. Hosking. He asserted that a person of ordinary skill would have been motivated to combine Visual Basic’s and Salton’s indexing systems because they “address[ed] the same technical issues and disclose[d] closely related subject matters.” J.A. 1651–52, 2782. He also stated that Visual Basic and Salton “both dea[lt] with solving the problem of locating key words in a database.” J.A. 1657, 2787. The Board could find those assertions, without more, insufficient to show persuasively that a relevant artisan would have recognized some deficiency in one reference or had some other reason to motivate looking to the other and combining them.

On appeal, Microsoft argues that Visual Basic's high-level disclosure of indexing would have motivated a person of ordinary skill in the art to turn to Salton for low-level implementation details. But Microsoft did not present that explanation to the Board.

CONCLUSION

For the foregoing reasons, we affirm the Board's determinations of unpatentability of claims 31, 41, 46–50, and 54 of the '604 and '775 patents; claim 42 of the '604 patent; and claim 45 of the '775 patent. We also affirm the Board's decisions that Microsoft did not prove the unpatentability of claims 32–40, 43–44, 51–53, 55–56, and 60 of both patents; claim 45 of the '604 patent; and claim 42 of the '775 patent.

No costs.

AFFIRMED