UNITED STATES DISTRICT COURT

NORTHERN DISTRICT OF CALIFORNIA

TWIN PEAKS SOFTWARE INC., Plaintiff, v. IBM CORPORATION, Defendant.	Case No. 14-cv-03933-JST **ORDER CONSTRUING CLAIMS OF U.S. PATENT NO. 7,418,439** Re: ECF No. 42

In this patent infringement case, the parties have requested that the Court construe disputed terms in the claims of U.S. Patent No. 7,418,439. See ECF Nos. 42, 43, 47 (claim construction briefs). After considering the arguments and evidence presented by the parties, as well as the relevant portions of the record, the Court construes the disputed claim terms as set forth below.

I. BACKGROUND

Plaintiff Twin Peaks Software Inc. brings this action against Defendant IBM Corporation, alleging that IBM is infringing U.S. Patent No. 7,418,439 (the "'439 Patent") relating to software for computer networking. ECF No. 7. The patent describes a system and method for sharing and storing files on a local or network file system. Id. ¶ 8. The '439 Patent was filed on March 19, 2001 and issued August 26, 2008. ECF No. 7-1, Ex. A, '439 Patent. The claimed invention was developed by John P. Wong and assigned to Twin Peaks. Id. The patent discloses a mirror file system ("MFS"), "a virtual file system that links two or more file systems together and mirrors between them in real time." Id., Abstract.

Twin Peaks alleges that IBM infringes the '439 Patent by marketing and distributing IBM Scale Out Network Attached Storage (SONAS), IBM Storwize V7000, IBM GPFS, IBM pCache, IBM Active Cloud Engine, IBM Active File Management, and IBM Automated File Management. Id. ¶ 10. These products use a technology called both "Panache" and "Active File Management."

Id. Twin Peaks asks that the Court declare that IBM has "infringed one or more claims, specifically including claim 1" of the '439 Patent. Id. at 5.[1] Since filing the Complaint, Twin Peaks has withdrawn its claim for infringement of claim 2. See ECF No. 47 at 6 n.1.

The parties identified seven terms, construction of which is "likely to be most significant to resolving the parties' dispute." See ECF No. 55; see also Patent L.R. 4-3. The Court held a claim construction hearing in this matter on December 15, 2015.

II. JURISDICTION

Because this is a civil action arising under an Act of Congress relating to patents, this Court has jurisdiction over this action pursuant to 28 U.S.C. § 1338.

III. LEGAL STANDARD

A. General Principles

The construction of terms found in patent claims is a question of law to be determined by the court. Markman v. Westview Instruments, Inc., 52 F.3d 967, 979 (Fed. Cir. 1995) (en banc), aff'd, 517 U.S. 370 (1996). "[T]he interpretation to be given a term can only be determined and confirmed with a full understanding of what the inventors actually invented and intended to envelop with the claim." Phillips v. AWH Corp., 415 F.3d 1303, 1316 (Fed. Cir. 2005) (quoting Renishaw PLC v. Marposs Societa' per Azioni, 158 F.3d 1243, 1250 (Fed. Cir. 1998)). Consequently, courts construe claims in the manner that "most naturally aligns with the patent's description of the invention." Id.

The first step in claim construction is to examine the language of the claims themselves. "It is a 'bedrock principle' of patent law that 'the claims of a patent define the invention to which the patentee is entitled the right to exclude.'" Phillips, 415 F.3d at 1312 (quoting Innova/Pure Water, Inc. v. Safari Water Filtration Sys., Inc., 381 F.3d 1111, 1115 (Fed. Cir. 2004)). A disputed claim term should be construed in light of its "ordinary and customary meaning," which is "the meaning that the term would have to a person of ordinary skill in the art in question at the time of the invention, i.e., as of the effective filing date of the patent application." Phillips, 415

[1] The pages refer to the pagination in the Court's Electronic Case Filing system.

United States District Court
Northern District of California

F.3d at 1312–13. In some cases, the customary meaning of a disputed term to a person of ordinary skill in the art is readily apparent, and claim construction involves "little more than the application of the widely accepted meaning of commonly understood words." Id. at 1314. Claim construction may deviate from the ordinary and customary meaning of a disputed term only if (1) a patentee sets out a definition and acts as his own lexicographer, or (2) "the patentee disavows the full scope of a claim term either in the specification or during prosecution." Thorner v. Sony Computer Entm't Am. LLC, 669 F.3d 1362, 1365 (Fed. Cir. 2012).

Ordinary and customary meaning is not necessarily the same as a dictionary definition. "Properly viewed, the 'ordinary meaning' of a claim term is its meaning to the ordinary artisan after reading the entire patent. Yet heavy reliance on the dictionary divorced from the intrinsic evidence risks transforming the meaning of the claim term to the artisan into the meaning of the term in the abstract, out of its particular context, which is the specification." Phillips, 415 F.3d at 1321. Typically, the specification "is the single best guide to the meaning of a disputed term." Vitronics Corp. v. Conceptronic, Inc., 90 F.3d 1576, 1582 (Fed. Cir. 1996). It is therefore "entirely appropriate for a court, when conducting claim construction, to rely heavily on the written description for guidance as to the meaning of claims." Phillips, 415 F.3d at 1317. However, while the specification may describe a preferred embodiment, the claims are not necessarily limited only to that embodiment. Id. at 1323.

Finally, in construing claims, courts may consider extrinsic evidence, such as "expert and inventor testimony, dictionaries, and learned treatises." Markman, 52 F.3d at 980. Expert testimony may be useful to "provide background on the technology at issue, to explain how an invention works, to ensure that the court's understanding of the technical aspects of the patent is consistent with that of a person of skill in the art, or to establish that a particular term in the patent or the prior art has a particular meaning in the pertinent field." Phillips, 415 F.3d at 1318. However, extrinsic evidence is "less reliable than the patent and its prosecution history in determining how to read claim terms." Id. When the intrinsic evidence supports one construction and the extrinsic evidence supports another, the court should choose the construction suggested by the intrinsic evidence.

B. Means Plus Function Claiming

The Patent Act authorizes functional claiming: "[a]n element in a claim for a combination may be expressed as a means or step for performing a specified function without the recital of structure, material, or acts in support thereof, and such claim shall be construed to cover the corresponding structure, material, or acts described in the specification and equivalents thereof." 35 U.S.C. § 112, ¶ 6. This provision is "intended to permit use of means expressions without recitation of all the possible means that might be used in a claimed apparatus." O.I. Corp. v. Tekmar Co., Inc., 115 F.3d 1576, 1583 (Fed. Cir. 1997). But the trade-off for inventors is that the "statutory provision was meant to preclude the overbreadth inherent in open-ended functional claims . . . which effectively purport to cover any and all means so long as they perform the recited functions." Halliburton Energy Servs., Inc. v. M–I LLC, 514 F.3d 1244, 1256, n.7 (Fed. Cir. 2008). The "duty to link or associate structure to function is the quid pro quo for the convenience of employing § 112, ¶ 6." Default Proof Credit Card Sys., Inc. v. Home Depot U.S.A., Inc., 412 F.3d 1291, 1300–02 (Fed. Cir. 2005).

Furthermore, in cases where the claim limitation subject to § 112, ¶ 6 "must be implemented in a special purpose computer," the specification must disclose more than "a general purpose computer or microprocessor." Williamson v. Citrix Online, LLC, 792 F.3d 1339, 1352 (Fed. Cir. 2015). Instead, the specification must "disclose an algorithm for performing the claimed function." Id. The specification can express the algorithm "in any understandable terms including as a mathematical formula, in prose, or as a flow chart, or in any other manner that provides sufficient structure." Finisar Corp. v. DirecTV Grp., Inc., 523 F.3d 1323, 1340 (Fed. Cir. 2008). For a claim term to be definite, "a recited algorithm . . . need not be so particularized as to eliminate the need for any implementation choices by a skilled artisan; but it must be sufficiently defined to render the bounds of the claim . . . understandable by the implementer." Ibormeith IP, LLC v. Mercedes–Benz USA, LLC, 732 F.3d 1376, 1379 (Fed. Cir. 2013).

In order to satisfy the definiteness requirement of § 112, ¶ 2, the specification must disclose the corresponding structure "in such a manner that one skilled in the art will know and understand what structure corresponds to the means limitation." Atmel Corp. v. Info. Storage

Devices, Inc., 198 F.3d 1374, 1382 (Fed. Cir. 1999). "A determination of claim indefiniteness is a legal conclusion that is drawn from the court's performance of its duty as the construer of patent claims," and "like claim construction, [it] is a question of law." Id. at 1378. Therefore, it is appropriate for the Court to address indefiniteness issues at the claim construction stage. See Exxon Research and Eng'g Co. v. United States, 265 F.3d 1371, 1376 (Fed. Cir. 2001).

IV. LEVEL OF ORDINARY SKILL IN THE ART

Claim construction is performed from the perspective of a person of ordinary skill in the art. Phillips, 415 F.3d at 1312–13.

IBM argues that the relevant time period is from 1999–2001 and that "one of ordinary skill in the field of computer systems and networking would have had the following qualifications":

> a bachelor's degree in electrical engineering and/or computer science or a related field; and either a Master's degree in electrical engineering and/or computer science or a related field or two years of experience in a technical role in the computer systems and networks industry, or equivalent relevant experience.

ECF No. 43 at 8.[2]

V. ANALYSIS

The parties identified the following seven claim terms as the most significant at time to

[2] Twin Peaks does not offer a competing proposal in its briefs but its expert provides a definition in his supplemental declaration:

> In my opinion, at the time of the patent application that led to the '439 patent, a person of ordinary skill in the art would have had: (i) a bachelor's degree in in a "STEM" discipline (Science, Technology, Engineering, or Math) utilizing computing as a significant part of their STEM discipline. (ii) the person will have had at least three to four years of similar Academic Experience (Masters or PHD degree), or at least three to four years of industry experience, and (iii) the person would have significant software development experience of two or more years specifically in both computer operating systems and file systems, as either a practitioner or as a researcher.

ECF No. 47-3, Supplemental Decl. of David Bernstein ¶ 6.

The Court construes Twin Peaks' failure to argue the point in its brief as a concession that the differences between the parties' person of ordinary skill in the art definitions are not meaningful, and adopts IBM's proposed definition.

United States District Court
Northern District of California

resolve the case:[3]

A. "Virtual file system"

Disputed Claim Term	Twin Peaks' Proposed Construction	IBM's Proposed Construction
"virtual file system" cl. 1, 2, 4, 17	"file system software that can manage data in physical storage indirectly through communication with other file systems that directly manages the data in physical storage"	"a file system that is layered on top of one or more conventional file systems and does not occupy physical storage"

The patent first describes the mirror file system as a type of virtual file system. See '439 Patent, Abstract. Claim 1 of the '439 Patent requires "[a] virtual file system which provides mirroring and linking of two file systems . . . ." Id. at 18:34–35. Claim 1 asserts that "[a] virtual file system which provides mirroring and linking of two file systems " with the "means for mounting components of . . . two file systems on a single mount point constituting a single root directory" that is "configured to manage the linking and mirroring" of the mounted components. Id. at 18:35–43.

Twin Peaks points out that the specification does not define the term but discusses the virtual file system interface in the UNIX Operating System. See id. at 5:42–45. The interface for UNIX's Operating System is the Virtual File System interface (VFS). Id. at 5:45–48. This VFS interface contains operations for the file system, such as mounting, unmounting, and synchronizing a file system. Id. at 5:50–65. This description, along with the accompanying figures in the patent, support the construction that a virtual file system provides a common interface that enables various applications access to file systems. For example, in describing the UNIX Operating System, the specification explains that when an application intends "to perform a file system operation on a file system," virtual file system macros invoke the appropriate file system operation. Id. at 7:2–9.

Twin Peaks first argues that the term "virtual" denotes software rather than a physical device. ECF No. 42 at 9. IBM, however, counters that "virtual" does not merely refer to software.

[3] The terms are discussed in the Order in which they appear in the claims.

Rather, the term refers to a "logical abstraction" that is "distinct from a physical implementation." ECF No. 43 at 20. The Court agrees with IBM that although software may create the virtual file system, the virtual file system itself is not software as suggested by Twin Peaks' construction.

Twin Peaks also proposes that the virtual file system manages data "indirectly" through file systems that manage the data "directly" but identifies nothing in the specification that distinguishes between indirect and direct management.

IBM contends that a "virtual file system" is not defined to occupy physical storage. See ECF No. 43 at 20–21. IBM argues that conventional file systems have physical storage, for example, the UNIX file system ("UFS") has local physical storage and the network file system ("NFS") has remote physical storage. IBM argues that the virtual file system, by contrast, does not have physical storage. Id. at 21. The specification and prosecution history supports IBM's construction. The specification states that "[t]he mirror file system does not have physical storage." '439 Patent at 9:40. This point is reiterated in the preferred embodiment. Id. 12:61–64 ("there is no physical storage for any file or directory within the mirror file system"). The prosecution history supports the construction that the virtual file system does not occupy physical storage. ECF No. 43-3, Declaration of Brice Lynch (Lynch Decl.) Ex. 3 at 76 of 80 ("the mirror file system of the present invention is a virtual, stackable, file system that does not occupy physical storage").

Figure 2, "a block diagram of a file system incorporating the present invention," describes that "the mirror file system is loaded on top of a Unix File System (UFS) and a Network File System (NFS)." Id. at 9:49–51. The figure depicts the virtual file system layered on top of local and remote file systems. See id., Fig. 2.

The parties dispute whether the claim term encompasses "conventional" file systems. Twin Peaks argues that the patent does not specify the kind of file systems that might have been managed beneath a virtual file system. See ECF No. 47 at 8. Twin Peaks, however, does note that the specification employs the term "regular file systems" or local or network file systems. Id. (citing to '439 Patent at 3:36 and the Abstract). IBM highlights that the written description describes Figure 1 as a "conventional file system framework," and includes a layout of several file systems. Id. at Fig. 1; 4:40–41. IBM further notes that this proposed construction is also supported by the inventor in his description of the invention. ECF No. 43-3, Lynch Decl. Ex. 3 at 76 of 80 ("[The mirror file system] is layered on top of two or more conventional file systems, e.g. local or network file systems.").

At the claim construction hearing, the Court proposed construing "virtual file system" as "a file system without physical storage that is layered above one or more local or network file systems." The parties stipulated to this construction, and accordingly, the Court adopts it.

B. "Means for mounting components of each of said two file systems on a single mount point constituting a single root directory for the components of both of said two file systems such that each mounted component of one of said two file systems is linked together with and becomes a mirroring pair with a corresponding mounted component in the other one of said two file systems"

Disputed Claim Term	Twin Peaks' Proposed Construction	IBM's Proposed Construction
"means for mounting components of each of said two file systems on a single mount point constituting a single root directory for the components of both of said two file systems such that each mounted component of one of said two file systems is linked together with and becomes a mirroring pair with a corresponding mounted component in the other one of said two file systems" cl. 1	"associating data with a directory so that the data can be accessed, wherein the directory is the highest-level one in a hierarchical directory structure to which components of two file systems are associated such that each component of the file system is associated with the corresponding components of the other file system"	**This term is governed by 35 U.S.C. § 112(6).** Indefinite Function: mounting components of each of said two file systems on a single mount point constituting a single root directory for the components of both of said two file systems such that each mounted component of one of said two file systems is linked together with and becomes a mirroring pair with a corresponding mounted component in the other one of said two file systems Structure: None.

The Court must first decide whether this claim is a "means-plus-function" claim as defined in 35 U.S.C. § 112, ¶ 6. Means-plus-function claims are construed to cover only "the corresponding structure . . . described in the specification and equivalents thereof." 35 U.S.C. §

112, ¶ 6. The Court concludes that it is.

First, there is a rebuttable presumption that § 112, ¶ 6, applies "[i]f the word 'means' appears in a claim element in association with a function." Callicrate v. Wadsworth Mfg., Inc., 427 F.3d 1361, 1368 (Fed. Cir. 2005); see also Williamson, 792 F.3d at 1348. Claim 1 uses the word "means" and also specifies the corresponding function. Second, while this rebuttable presumption can be overcome if "the claim recites sufficient structure for performing the described functions in their entirety," TecSec, Inc. v. Int'l Bus. Machines Corp., 731 F.3d 1336, 1347 (Fed. Cir. 2013) (quoting TriMed, Inc. v. Stryker Corp., 514 F.3d 1256, 1259 (Fed. Cir. 2008)), claim 1 does not do that. "Sufficient structure exists when the claim language specifies the exact structure that performs the function in question without need to resort to other portions of the specification or extrinsic evidence for an adequate understanding of the structure." TriMed, 514 F.3d at 1259–60. Claim 1 does not recite a structure that performs the recited function; instead, Twin Peaks cites to other portions of the specification to describe the structure. See ECF No. 42 at 11. Finally, the prosecuting attorney admitted that the claim term is written in means-plus-function language. ECF 43-9, Lynch Decl., Ex. 7, LaBarre Aug. 28, 2015 Tr. at 90:16–21. For these reasons, the Court will apply § 112, ¶ 6 in analyzing the claim.

Having determined that Claim 1 is a means-plus-function claim, the Court next "attempt[s] to construe the disputed claim term by identifying the 'corresponding structure, material, or acts described in the specification' to which the claim term will be limited." Robert Bosch, LLC v. Snap-On Inc., 769 F.3d 1094, 1097–98 (Fed. Cir. 2014) (quoting Welker Bearing Co. v. PHD, Inc., 550 F.3d 1090, 1097 (Fed.Cir.2008)). If the Court cannot identify any "corresponding structure, material, or acts described in the specification," the claim term is indefinite. Id.

For computer-implemented means-plus-function limitations that cannot be performed by a general purpose computer without special programming, the specification must disclose structure in the form of an algorithm. Aristocrat Techs. Austl. Pty Ltd. v. Int'l Game Tech., 521 F.3d 1328, 1332 (Fed. Cir. 2008). When assessing a disclosed algorithm's adequacy, the question is "whether the [] algorithm, from the viewpoint of a person of ordinary skill, is sufficient to define the structure and make the bounds of the claim understandable." Medical Instr. & Diagnostics Corp.

v. Elekta AB, 344 F.3d 1205, 1211–12 (Fed. Cir. 2013). The algorithm need not take a particular form; "it suffices if the specification recites in prose the algorithm to be implemented by the programmer." Typhoon Touch Techs., Inc. v. Dell, Inc., 659 F.3d 1376, 1386 (Fed. Cir. 2011).

Here, the parties agree that if the Court determines that the term is a means-plus-function, the means used in claim 1 performs one function:

> mounting components of each of said two file systems on a single mount point constituting a single root directory for the components of both of said two file systems such that each mounted component of one of said two file systems is linked together with and becomes a mirroring pair with a corresponding mounted component in the other one of said two file systems.

ECF No. 34 at 5; '439 Patent at 18:37–46.

Twin Peaks argues that the specification fully discloses the method of mounting file systems using the invention's mount protocol. ECF No. 42 at 13. According to Twin Peaks' expert, David Bernstein, one skilled in the art could understand and implement the claim using software techniques and tools available at the time of the invention. ECF 42-7, Bernstein Decl. ¶ 22.

Twin Peaks identifies the relevant structure as software containing algorithms disclosed at 3:11–28, 9:49–10:26, 10:32–12:14, 12:24–14:18, 15:55–16:67, and Figures 3-10. The text at 3:11–28, however, discusses the outcome after a system administrator implements the MFS mounting protocol. The text at 9:49–10:26 describes Figure 2, which portrays several file systems and the mirror file system. This section and corresponding figure explain that the interface for the virtual file system and the interface for individual file or directory are kept "between the system call and MRS, and between the MFS and the underlying UFS and NFS." Id. at 9:55–56. This section then describes the outcomes from this set-up. The text at 10:32–12:14 discusses the MFS Mount Protocol in detail, but only recites functional, rather than structural language. See id. at 10:46–48 ("The MFS mount protocol allows either an entire system of [sic] part of a file system . . . to be mounted."); id. at 10:49–51 ("When the MFS mounts a directory with a file system or a part of a file system, the previous contents of the mounted directory are not hidden."); id. at 10:52–54 ("The MFS inherits all of the contents of the mounted directory into its mfs_vfs

United States District Court
Northern District of California

virtual file system data structure. The inherited content is a copy of a mirroring pair."); and *id.* at 10:59–62 ("The application still sees the previous contents of the mounted directory through its previous path name. The application also sees the contents of the newly mounted file system through its previous path name."). The section goes to explain that the MFS holds information for itself, the file system inherited from the mounted directory, and the newly mounted file system. The preferred embodiment then includes the data structure of the MFS, which includes the MFS and the two vfs structures of the file systems linked by the MFS:

```
Struct mfs_vfs {
        Struct vfs      *mfs;   /* mirror file system vfs */
        Struct vfs      *X_vfs;/* first file systems vfs */
        int     X_vfs_state;    /* state of X_vfs file system */
        Struct vfs      *Y_vfs;/* second file system vfs */
        int     Y_vfs_state;    /* state of Y_vfs file system */
        Other fields
        .............
};
```

Id. at 11:6–14.

IBM argues that claim 1 is invalid for indefiniteness because the '439 Patent fails to recite algorithms that perform the function. ECF No. 43 at 14. IBM notes that the specification provides the standard UNIX file system mount command, but insists that this is not an algorithm and that is not sufficient. *See id.* at 15; *see also* '439 Patent at 5:55 (identifying the Unix file system mount command "vfs_mount( )"). IBM contends that the MFS Mount Protocol disclosed in the specification "merely provides a description of its outcome" and not the means for achieving it. *Id.* at 14. IBM argues the '439 Patent does not disclose an algorithm for mounting two file systems on a single mount point that (1) does not hide the contents of the first-mounted file systems, (2) creates the MFS super data structure, (3) inherits the components of two or more underlying file systems, and (4) that is configurable during the operation.

The MFS Mounting Protocol differs from the Unix file system mount command in that it does not hide the contents of the previously mounted file system. '439 Patent at 10:36–51. Yet the MFS mount protocol does not disclose how to perform the command such that two file systems can be mounted on a single directory making content from both file systems are available.

For example, although the section on the MFS mount protocol discloses code for data structures,'439 Patent at 11:6–14 and 12:46–56, the specification provides no guidance about the algorithms that ultimately fall within the scope of the claim. See ECF No. 43-13, Declaration of Nicholas Bambos (Bambos Decl.) ¶ 152.

The Court concludes that the specification of the '439 Parent does not disclose sufficient structure for the mounting function associated with the means for mounting limitation. Although one skilled in the art could "arrive at a 'means for mounting,'" id. ¶ 152, the specification "fails to disclose the steps necessary to actually perform that suggested algorithm," Ibormeith IP, 732 F.3d at 1379. The specific algorithm for the structure of the means for mounting term need not be explicitly disclosed in the specification, but here, the specification does not contain a sufficiently precise description of the particular structure able to implement the function. See Blackboard, Inc. v. Desire2Learn, Inc., 574 F.3d 1371, 1385 (Fed. Cir. 2009) (distinguishing enablement, where an "ordinarily skilled artisan might be able to design a program," from definiteness).

The specification fails to describe the means by which the MFS Mount Protocol mounts components from two file systems on a single mount point so that the pointed components become a mirrored pair. Twin Peaks "attempt[s] to capture any possible means for achieving that end. [But] section 112, paragraph 6, is intended to prevent such pure functional claiming." Id. (citing Aristocrat Techs. Australia Pty Ltd. v. Int'l Game Tech., 521 F.3d 1328, 1333 (Fed. Cir. 2008)). The Court finds that the '439 Patent discloses insufficient structure to perform the function of "mounting components of each of said two file systems on a single mount point constituting a single root directory for the components of both of said two file systems such that each mounted component of one of said two file systems is linked together with and becomes a mirroring pair with a corresponding mounted component in the other one of said two file systems." Accordingly, the Court finds that claim 1 is indefinite.

C. "Mirroring"/"mirroring pair"

Disputed Claim Term	Twin Peaks' Proposed Construction	IBM's Proposed Construction
"Mirroring" / "mirroring pair"	"file system software providing replication or	"maintaining identical copies by synchronizing them with

United States District Court
Northern District of California

cl. 1, 2, 4	duplication of the same file(s), and directory(ies)."	one another instantaneously (or immediately)" "a pair of identical components that is synchronized instantaneously (or immediately)"

Claim 1 describes an operation where "each mounted component of one of said two file systems is linked together with and becomes a mirroring pair with a corresponding mounted component in the other one of said two file systems." '439 Patent at 18:39–43 (emphasis added). Claim 2, which depends on claim 1, recites "wherein said super application interface data structure of said virtual file system is configured to serve as a fundamental interface frame structure to link said mounted file systems together as a mirroring pair." And claim 4 recites "[a] method for mirroring files and directories between file systems on a computer system or on two computer systems connected to each other via a network" in the preamble.

The parties first disagree about the temporal aspect of the claim term. Twin Peaks argues that, by itself, the term "mirroring" does not include the limitation of "real time delivery of updates." ECF No 42 at 14. Conversely, IBM contends that Twin Peaks' construction "reads out the term's temporal aspect." ECF No. 43 at 8. IBM points out that the specification refers to the term "real time" in reference to mirroring eight times in the Summary of Invention alone. Id. at 9; see, e.g., '439 Patent at 3:36–38 ("the mirror file system links any two regular file systems together and provides data management to make sure that the two file systems contain identical data and are synchronized with each other in real time") (emphasis added). IBM also argues that a construction that does not include real-time synchronization eliminates a major benefit of the invention. ECF No. 43 at 10 (citing to MyMail, Ltd. v. Am. Online, Inc, 476 F.3d 1372, 1376 (Fed. Cir. 2007).

The Court agrees with IBM that "mirroring" in the context of this patent includes a temporal characteristic that implies immediacy. See, e.g., '439 at 3:40–44 ("All other vnode operations . . . follow the same procedure as described in mfs_open( ) to perform the same identical operations with the same parameters on both copies of the mirroring pair. This is how the mirror file system achieves the real-time mirroring effect between the mirroring pair."); id. at

3:26–28 (“With this mirroring mechanism of the MFS, the files/directories in one file system are mirrored to their mirroring counterparts of another file system in real time.”). The Court will construe the term “mirroring” in accordance with the scope of the invention. See Netcraft Corp. v. eBay, Inc., 549 F.3d 1394, 1398 (Fed.Cir.2008) (holding that language describing “present invention,” when read in light of the specification and prosecution history, described the entire invention and therefore had to be reflected in claim constructions).

Critically, the specification unequivocally states the present invention mirrors between two or more file systems in real time. And although it is not phrased as a concession, Twin Peaks acknowledges that the temporal aspect of mirroring is an important aspect of the invention. See ECF No. 59 (hearing transcript) at 20–22; ECF No. 47 (reply brief) at 12 (mirroring “happens much more quickly than conventional daily or monthly backups”). This suggests that immediacy is an important feature of the invention. For example, in Verizon Services Corp. v. Vonage Holdings Corp., the Federal Circuit addressed the claim term “localized wireless gateway system.” 503 F.3d 1295, 1308 (Fed. Cir. 2007). In construing the term, the court observed that while describing “the present invention,” “the specification then states that ‘[t]he gateway compresses and decompresses voice frequency communication signals and sends and receives the compressed signals in packet form via the network.’” Id. (citation omitted). The court determined that the description limited the scope of the invention as a whole and concluded that the term “localized wireless gateway system” “must be limited to one performing compression and packetization functions at the gateway.” Id.

The parties next disagree about the acts required by the claim term. Twin Peaks interprets “mirroring” as synonymous with “reflecting” and that the term refers to duplication. See ECF No. 42 at 14; ’439 Patent at 3:42–44 (“[T]he mirror file system allows critical information to be reflected simultaneously on multiple servers at different sub-networks”). IBM argues that Twin Peaks’ construction could include a single act of copying and this does not comport with “mirroring” or yield a “mirroring pair.” ECF No. 43 at 10–11. The Court agrees that the claim term should not be construed in a manner that could encompass only a single instance of replication.

The Court construes "mirroring" as "maintaining identical copies by synchronizing them with one another immediately" and "mirroring pair" as "two components that maintain identical copies by synchronizing with each other immediately."

D. "A super application interface data structure containing an application interface data structure of said virtual file system, and said application interface data structures of each of said two file systems"

Disputed Claim Term	Twin Peaks' Proposed Construction	IBM's Proposed Construction
"a super application interface data structure containing an application interface data structure of said virtual file system, and said application interface data structures of each of said two file systems" cl. 1	"computer code for the management and access of a virtual file system having two or more files systems containing pointers or equivalent linking methods between the two or more file systems"	Indefinite <u>Alternatively</u>: "a super data structure for interfacing with user level applications that has as part of its structure a data structure of the virtual file system for interfacing with user applications through file system operation system calls, and said application interface data structures of each of said two file systems"

This disputed term appears in claim 1, which asserts a method for mirroring and linking two file systems. '439 Patent at 18:34–35. The claim describes that a virtual file system includes "a super application interface data structure" which contains three "application interface data structure[s]": one from the virtual file system and one from each of the two file systems. <u>See</u> <u>id.</u> at 18:49–53.

"Application interface data structure" and "super application interface data structure" were coined for the '439 Patent. <u>See</u> ECF No. 43-2, Lynch Decl., Ex. 1, Wong Sept. 30, 2015 at 85:6–15. "A patentee is free to act as his or her own lexicographer . . . . However, if the patentee chooses to act as his or her own lexicographer, the special definition must be clearly stated within the patent specification or file history." <u>Vitronics Corp.</u>, 90 F.3d at 1582. A court will not find a patented claim indefinite unless the claim interpreted in light of the specification and the prosecution history fails to "inform those skilled in the art about the scope of the invention with reasonable certainty." <u>Nautilus, Inc. v. Biosig Instruments, Inc.</u>, 134 S. Ct. 2120, 2129 (2014).

IBM argues that the terms do not have an ordinary meaning and the specification does not provide a clear statement of definitions for the coined terms in the specification or file history. ECF No. 43 at 23. Twin Peaks responds that the meaning can be inferred from the specification and terms such as "data structure," "application interface," and "super" have accepted meanings in the invention. ECF No. 47 at 13.

The Court first turns to the specification to determine the meaning of "super application interface data structure" and "application interface data structure." See MyMail, Ltd., 476 F.3d at 1376 (Fed. Cir. 2007) (looking to the specification to determine the claim term where both parties agreed that the term was coined and had no meaning apart from the patent).

IBM argues that the only example of a "super application interface data structure" is disclosed at 11:5–14 (showing the that the super mfs_vfs structure includes the structure from the virtual file system as well as the structure of the two mounted file systems). As discussed below, the specification – and in particular, the description of the super mfs_vfs data structure – provides sufficient information to one skilled in the art about the scope of the invention.

IBM and Twin Peaks both agree that the data structure is "super" because the data structure inherits all of the contents of the vfs data structure and the two file systems beneath it. See ECF No. 43 at 20; ECF No. 47 at 13–14. The specification also supports this interpretation, because the mfs_vfs structure holds the vfs data structure for the MFS itself and the vfs structures for the two file systems. See '439 Patent at 11:7–13.

Next, IBM and Twin Peaks dispute whether the inclusion of "computer code" in the construction is too generic. IBM contends while the embodiment may contain underlying computer code, Twin Peaks' proposed construction reads specific data structures out of the term. See '439 Patent at 11:6–14 (identifying the following data structures: struct mfs_vfs, struct vfs *mfs, struct vfs *X_vfs, struct vfs * Y_vfs). At the claim construction hearing, however, Twin Peaks agreed to adopt "data structure" in lieu of "computer code."

Turning to "application interface data structure," Twin Peaks' proposed construction does not describe the relationship between the virtual file system, the file systems, and application programs. In its reply brief, however, Twin Peaks expands its argument to explain that the data structure receives file system requests from applications. See ECF No. 47 at 13–14. This is disclosed by the specification, which explains that an application program send system calls to perform file system operations like open, read, and write. ’439 Patent at 7:1–25.

The parties also disagree whether the construction includes “pointers or equivalent linking methods” or “file system operation system calls.” The specification describes that generally, operating systems have two types of interfaces: “one is the interface for the file system itself” (the VFS interface) and “the other is the interfaces for individual files or directories within the file system” (the vnode interfaces). See ’439 Patent at 5:41–46. The VFS interface includes the “vsops structure,” which contains file system operations invoked by macros definitions. The “vnode interface” includes the “vnodeops structure,” which contains file/directory operations invoked by macros definitions. The specification also discusses the preferred embodiment of the mirror file system and explains that each file or directory of the MFS has a “super vnode structure called the mnode. This mnode contains a vnode structure and two vnode pointers.” Id. at 12:38–40. The embodiment then goes on to show the super vnode structure:

```
/*
  * The mnode is the “vnode” mirror files. It contains
  * all the information necessary to handle two real vnodes it links
  */
typedefstruct mnode {
        struct vnode    m_vnode;        /* vnode for mirror file system */
        struct mnode    *m_next;        /* link for hash chain */
        struct vnode    *m_Xvp;         /* pointer to X vnode */
        struct vnode    *m_Yvp;         /* pointer to Y vnode */
        int        state;   /* state of the mnode */
} mnode_t;
```

Id. at 12:46–56.

The specification reveals that the mnode contains its own vnode and two pointers that point to each vnode of the two file systems. ’439 Patent at 12:39–43. The vnode interface, in turn, has “30-40 interfaces/operations for a file/directory.” Id. at 7:10–25.

Taken together, the Court adopts a modified version of IBM’s alternative construction. The Court construes the term to mean “a data structure for interfacing with applications that has as part of its structure a data structure of the virtual file system, which includes an interface for file system operation calls and pointers, and the data structures of the two file systems, which contain

United States District Court
Northern District of California

system operation calls for each file system."

E. "A mechanism for managing said component within said virtual file system and the corresponding linked components within said two file systems"

Disputed Claim Term	Twin Peaks' Proposed Construction	IBM's Proposed Construction
"a mechanism for managing said component within said virtual file system and the corresponding linked components within said two file systems" cl. 4	"computer code for managing the components with the virtual file system and within the components and corresponding components within each of the two file systems"	**This term is governed by 35 U.S.C. § 112(6).** Indefinite Function: managing said component within said virtual file system and the corresponding linked components within said two file systems Structure: None

The parties dispute whether the "mechanism for managing" is a means-plus-function term. At the claim construction hearing, however, Twin Peaks appeared to concede that this is a means-plus-function term. The term does not use the word "means" and is therefore presumed to fall outside of § 112, ¶ 6. Watts v. XL Sys., Inc., 232 F.3d 877, 880 (Fed. Cir. 2000).[4] Nonetheless, the term "mechanism" may be used in the claim in the same manner as "means." Williamson, 792 F.3d at 1350 ("Generic terms such as 'mechanism' . . . and words that reflect nothing more than verbal constructs may be used in a claim in a manner that is tantamount to using the word 'means' because they 'typically do not connote sufficiently definite structure' and therefore may invoke § 112, para. 6."). Also, if a challenger demonstrates that the claim term fails to "recite sufficiently definite structure," the presumption can be overcome and § 112, ¶ 6 may apply. Id.

Claim 4 asserts "a method for mirroring files and directories between file systems . . . comprising the steps of" mounting and then:

> configuring said virtual file system so that each component of said virtual file system has a super application interface data structure containing an application interface data structure of said component in said virtual file system, an application interface data structure of a

[4] The Federal Circuit has recently clarified that this presumption is not "strong." Williamson, 792 F.3d at 1349.

> linked component in said one of said two file systems, and an application interface data structure of said corresponding linked component in said other one of said two file systems, said application interface data structure of said component in said virtual file system providing a mechanism for managing said component within said virtual file system and the corresponding linked components within said two file systems[.]

’439 Patent at 19:1–25 (emphasis added). The claim term describes that each component of the virtual file system’s application interface data structure provides a “mechanism for managing” the component as well as the linked components of the two mounted file systems.

Twin Peaks and IBM agree that the means used in claim 4 performs the function of “managing said component within said virtual file system and the corresponding linked components within said two file systems.” The parties disagree whether the structure is sufficiently disclosed.

Twin Peaks argues that the term identifies the structure as the application interface data structure. ECF No. 42 at 18. Alternatively, if the term does not identify sufficient structure, Twin Peaks contends that the “software containing algorithms at 12:24–14:18, 15:55–16:67, and Figs. 3-10 in the ’439 Patent that manage operations of the virtual file system and the corresponding linked components of the two file systems” provides the structure. Id. Twin Peaks maintains that the specification discloses an algorithm for managing by providing an example of the “open” operation. See ’439 Patent at 13:22–51 (providing code for mfs_open( ) in an mirror file system). The specification goes on to explain that “[a]ll other vnode operations like mfs_read( ), mfs_write( ), mfs_setattr( ), mfs_close( ), etc., follow the same procedure as described in mfs_open( ).” Id. at 13:53–54. Twin Peaks argues that the most common and relevant operations to the invention, including write, read, and open, are disclosed. See ECF No. 47 at 15.

Because “application interface data structure” has no generally understood meaning in the art, IBM contends that the corresponding structure for this term must be disclosed in the form of an algorithm. ECF No. 43 at 18–19. IBM asserts that the disclosure of one mnode operation, the open operation (“mfs_open( )”), is insufficient. The specification states that there are 30 to 40 other vnode operations but only enumerates 11 of the operations. ’439 Patent at 7:15–25.[5]

[5] The specification identifies operations that checks access to a file, closes a file, creates a file, gets

Further, the specification explains only one mnode operation and then states that all the operations "follow the same procedure as described in mfs_open( )." ECF No. 43 at 19–20 (citing to '439 Patent at 13:52–57). IBM additionally points out that the specification reveals that the operations are carried out differently. For example, the "open," "create," and "write" operations go to both mounted copies of the mirroring pair but the "read" operation need only go to one. Id. at 21. IBM argues that because the read operation does not "follow the same procedure" as the open operation disclosed and the specification does not disclose algorithms as to the remaining managing operations. At the claim construction hearing, IBM also stressed that the "write" operation is carried out differently because it requires "a locking mechanism" to ensure that no other operations are allowed during data writing. See '439 Patent at 14:19–36. Yet the specification does not disclose any algorithm for controlling access to the components during the write operation. Because of these omissions, IBM urges the Court to find claim 4 indefinite for failing to recite sufficient structure.

The Court agrees with IBM and finds that the specification fails to disclose sufficient structure. The specification first discloses the code for one operation in the managing function—the open operation (mfs_open). However, the specification does not identify which file operations are necessary to accomplish the managing function. Instead, the specification recites an algorithm for the open operation, and then states that all other operations, including the read, write, set attributes, and close operations follow the same procedure.[6] The specification then differentiates the read operation and the write operation from the open operation but does not disclose the arguments, if any, required for the other operations.

Twin Peaks' expert, Mr. Bernstein, states that the operations are "very similar" in look. See ECF No. 47-3, Bernstein Supplemental Decl. ¶ 34. He posits that one skilled in the art need only (1) "replace the name of one operation for the other," (2) include "the appropriate arguments

the attributes for a file, creates a link for a file, looks up a path name for a file, makes a directory, opens a file, reads the data from a file, sets the attributes for a file, and writes the data to a file. '439 Patent at 7:15–25.

[6] The specification also identifies 11 vnode operations of the "30 to 40 interfaces/operations." See '439 Patent at 7:10–25.

for each operation (which are set out in the specification)," and (3) "change the read operation so that only one file system is called (also as explained in the specification)." Id. Bernstein concludes that "one skilled in the art would easily have been able to code these data structures and operations without using any extraordinary insight or engaging in any experimentation." Id.

Mr. Bernstein also provides "actual source code one skilled in the art would produce" for the mfs_close, mfs_read, and mfs_write operations to rebut IBM's arguments. See id. ¶¶ 31–33. For example, using the disclosures from the specification, Bernstein produced source code for the write operation. The specification describes the following about the write operation: first, the write operation will go to both copies and second, to make sure the two copies remain identical,

> the write operation on both copies should be atomic; in other words, during the data writing to both copies, no other operations (read and/or write) should be allowed on the two copies. To achieve this, a locking mechanism is needed. The MFSs' vop_write( ) operation acquires the locks by calling the vop_rwlock( ) operation of the first vnode, then acquires the lock for second vnode. Both locks of vnode have to be secured before the writing can proceed. If only one lock is granted, and the other one is held by another process, the MFS releases the first lock it is holding to avoid a deadlock in the case that another process that held the second lock also is trying to hold the first lock. After releasing the lock of the first vnode, the vop_write( ) operation uses a backoff algorithm to wait for a period of time before trying to acquire the locks on both vnodes again.

'439 Patent at 14:5–6, 14:20–36. As IBM noted at the claim construction hearing, however, the code implementation provided by Mr. Bernstein did not actually include the code necessary to accomplish atomicity. See Bernstein Supplemental Decl. ¶ 33. Bernstein's sample code demonstrates that the disclosure of the open operation is a partial but inadequate guide for other operations. Thus, the Court disagrees with Twin Peaks' contention that sufficient structure is disclosed. The specification fails to delineate the operations encompassed by the managing function and also fails to set out the appropriate arguments for each of these operations. See Media Rights Techs., Inc. v. Capital One Fin. Corp., 800 F.3d 1366, 1374 (Fed. Cir. 2015) ("Where there are multiple claimed functions, as there are in this case, the patentee must disclose adequate corresponding structure to perform all of the claimed functions.").

Typhoon Touch Techs., Inc. v. Dell, Inc., 659 F.3d 1376 (Fed. Cir. 2011), provides a useful contrast. There, the Federal Circuit addressed the claim term "means for cross-referencing

United States District Court
Northern District of California

said responses with one of said libraries of said possible responses." Id. at 1383. The court found the disclosed algorithm sufficient because the specification stated that "cross-referencing entails the steps of data entry, then storage of data in memory, then the search in a library of responses, then the determination if a match exists, and then reporting action if a match is found." Id. at 1386. The court concluded that the steps "are carried out by known computer-implement operations" and "readily implemented by persons of skill in computer programming." Id. The Federal Circuit reversed the district court's judgment of invalidity. Id.

Here, unlike Typhoon Touch, where the specification described all the steps the "cross-referencing" function entailed, the specification at issue fails to describe all the operations the "managing" function entails. Although the specification provides an example of the computer-implement operation (open) and also provides additional instruction on two of the operations (read and write), the specification does not identify all of the operations necessary to manage a component of the virtual file system. Nor does the specification provide the arguments each operation may require such that one of ordinary skill in the art could accomplish the other operations. The specification does not disclose sufficient defining structure, through code and prose, "to render the bounds of the claim understandable to one of ordinary skill in the art." AllVoice Computing PLC v. Nuance Commc'ns, Inc., 504 F.3d 1236, 1245 (Fed. Cir. 2007). And the "corresponding structure" does not include "all structure that actually performs the recited function." Cardiac Pacemakers, Inc. v. St. Jude Med., Inc., 296 F.3d 1106, 1119 (Fed. Cir. 2002).

The Court finds that the '439 Patent discloses insufficient structure to perform the function "managing said component within said virtual file system and the corresponding linked components within said two file systems." Accordingly, claim 4 is indefinite.

/ / /

/ / /

/ / /

/ / /

/ / /

/ / /

F. “A super application interface data structure containing an application interface data structure of said component in said virtual file system, an application interface data structure of a linked component in said one of said two file systems, and an application interface data structure of said corresponding linked component in said other one of said two file systems”

Disputed Claim Term	Twin Peaks' Proposed Construction	IBM's Proposed Construction
“a super application interface data structure containing an application interface data structure of said component in said virtual file system, an application interface data structure of a linked component in said one of said two file systems, and an application interface data structure of said corresponding linked component in said other one of said two file systems” cl. 4	See super application interface data structure, application interface data structure, above; plain and ordinary meaning	Indefinite Alternatively: “a super data structure for interfacing with user applications through file operation system calls that has as part of its structure a data structure of said component in said virtual file system for interfacing with user application through file operation system calls, a data structure of a linked component in said one of said two file systems for interfacing with user applications through file operation system calls, and a data structure of said corresponding linked component in said other one of said two file systems for interfacing with user applications through file operation system calls”

Twin Peaks argues that the dispute for this term is the same as those presented with respect to the disputed term “a super application interface data structure containing an application interface data structure of said virtual file system, and said application interface data structures of each of said two file systems.” Twin Peaks accordingly incorporates by reference the arguments for that term and applies it to this term. See ECF No. 42 at 20.

IBM asserts that because the coined phrase is never used in the specification and is not defined, claim 4 is invalid for indefiniteness for the same reason as claim 1. See ECF No. 43 at 26. Alternatively, IBM contends that Twin Peaks ignores the intrinsic record and broadens the scope of the claims. Id. As IBM points out, there is a distinction between claim 4, which claims a method for mirroring files and directories between file systems, and claim 1, which claims a method for mirroring two file systems. Id.

Twin Peaks' proposed construction is not consistent with the claim language that the each component of the virtual file system has a super application interface data structure. As made clear by the claim, claim 4 asserts a method for files and directories, rather than file systems. The Court will modify the construction for the analogous term in claim 1 to include this difference. The Court construes the term to mean "a data structure for interfacing with applications that has as part of its structure the data structure of the virtual file system component, which include an interface for file system operation calls and pointers, and the data structures of the two corresponding linked components of the file systems, which include file/directory operation calls for each component."

G. "In real time"

Disputed Claim Term	Twin Peaks' Proposed Construction	IBM's Proposed Construction
"in real time" cl. 4, 17	"as soon as file system modifications are made, as opposed to a scheduled periodic basis"	"instaneously (or immediately)"

"In real time" appears in claims 4 and 17 as it relates to when the virtual file system causes updates to a file system upon receiving a request to perform a write operation in another file system. See '439 Patent at cl. 4, 17.

Twin Peaks argues that IBM "cherry picks" its construction from a passage that discusses the embodiment of the invention in the specification. See ECF No. 42 at 20. The section describes the features and benefits of the mirror file system:

> There are several benefits associated with the use of the mirror file system. A network server with the mirror file system on a sub-network can mirror its file system to another file system located on another network server, or on a different sub-network, in real time. Thus, the mirror file system allows critical information to be reflected simultaneously on multiple servers at different sub-networks, which synchronize with one another instantaneously so that neither time nor information is lost during updates.

'439 Patent at 3:38–54 (emphasis added). Twin Peaks states that its proposed construction is a

United States District Court
Northern District of California

"common sense meaning" of the claim term in the context of the "patent's environment . . . and its objectives." ECF No. 42 at 21.

A person of ordinary skill in the art would understand that "real time," in the specification, does not literally mean "instantaneously" such that no time is lost. IBM's expert, Dr. Bambos, concedes that computer systems with normal operations have physical limitations that "naturally impose unintended delays." ECF No. 43-13, Bambos Decl. ¶ 105 ("Just how instantaneous the mirroring is depends on the system's physical limitations, but those limitations would naturally impose unintended delays on a millisecond timescale in computer systems and networks.").

IBM cites to extrinsic evidence, such as dictionaries from the relevant time period. ECF No. 43 at 12. Random House Webster's Computer and Internet Dictionary defines "real time" as an "immediate response by a computer system." ECF No. Lynch Ex. 5 (Philip Margolis, Computer & Internet Dictionary at 470 (3d ed. 1999).

According to the intrinsic and extrinsic record, the Court construes "in real time" to mean "immediately as file system modifications occur."

H. Agreed Upon Terms in the Parties' Joint Claim Construction Statement

Finally, in the parties' first amended joint claim construction statement, the parties agreed to the construction of the following terms. See ECF No. 55. The Court will therefore adopt the parties' constructions.

Claim Term	Agreed Upon Construction
"the mounted components of each file system are replicated in the other file system"	Plain and ordinary meaning
"the write operation performed on said one component stored on one of said two file systems to be replicated in the corresponding component"	Plain and ordinary meaning

/ / /

/ / /

/ / /

/ / /

/ / /

United States District Court
Northern District of California

CONCLUSION

The Court, for the foregoing reasons, construes the disputed claim terms as set forth above.

The Court will conduct a further Case Management Conference on May 25, 2016 at 2:00 p.m. The parties are ordered to submit a Joint Case Management Statement by May 18, 2016.

IT IS SO ORDERED.

Dated: April 11, 2016

JON S. TIGAR
United States District Judge