

TABLE OF CONTENTS

I. TECHNOLOGY OVERVIEW 1

II. PRINCIPLES OF CLAIM CONSTRUCTION..... 3

III. LEVEL OF ORDINARY SKILL IN THE ART..... 4

IV. DISPUTED CLAIM CONSTRUCTIONS..... 4

 A. Claims Terms Not Governed by 35 U.S.C. 112 ¶ 6 4

 1. “linked list to store and provide access to records” / “linked list of records”
 [claims 1, 3, 5, and 7]..... 4

 2. “automatically expiring / expired” [claims 1, 3, 5, and 7]..... 6

 3. “identifying at least some of the automatically expired ones of the records”
 [claims 3 and 7]..... 7

 4. “removing at least some of the automatically expired records from the linked list
 when the linked list is accessed” [claims 3 and 7]..... 8

 5. “dynamically determining maximum number of expired ones of the records to
 remove when the linked list is accessed” [claims 4 and 8]..... 10

 B. Claim Terms Governed by 35 U.S.C. 112 ¶ 6 13

 1. “a record search means utilizing a search key to access the linked list” [claim 1]13

 2. “a record search means utilizing a search key to access a linked list of records
 having the same hash address [claim 5]..... 13

 3. “the record search means including a means for identifying and removing at least
 some [of the] expired ones of the records from the linked list [of records] when
 the linked list is accessed” [claims 1 and 5] 17

 4. “means, utilizing the record search means, for accessing the linked list and, at the
 same time, removing at least some of the expired ones of the records in the linked
 list” [claim 1] 21

5.	“mea[n]s, utilizing the record search means, for inserting, retrieving, and deleting records from the system and, at the same time, removing at least some expired ones of the records in the accessed linked list of records’ [claim 5]	22
6.	“a hashing means to provide access to records stored in a memory of the system and using an external chaining technique to store the records with same hash address, at least some of the records automatically expiring” [claim 5]	24
7.	“means for dynamically determining maximum number for the record search means to remove in the accessed linked list of records [claims 2 and 6]	25
C.	Miscellaneous Construction Issues	27
1.	Ordering of the claims [claims 3 and 7].....	27
2.	“identifying . . . and removing . . . when the linked list is accessed” [claims 3 and 7]	27
V.	CONCLUSION	28

TABLE OF AUTHORITIES

	Page(s)
CASES	
<i>Altiris v. Symantec Corp.</i> , 318 F.3d 1363 (Fed. Cir. 2003).....	27
<i>Baran v. Med. Techs., Inc.</i> , No. 2010 1058, 2010 WL 3178377 (Fed. Cir. Aug. 12, 2010).....	13, 17
<i>Daiichi Sankyo Co. v. Apotex, Inc.</i> , 501 F.3d 1254 (Fed. Cir. 2007).....	4
<i>E-Pass Techs., Inc. v. 3COM Corp.</i> , 343 F.3d 1364 (Fed. Cir. 2003).....	passim
<i>Finisar Corp. v. DirectTV Group, Inc.</i> , 523 F.3d 1323 (Fed. Cir. 2008).....	26
<i>Globetrotter Software, Inc. v. Elan Computer Group, Inc.</i> , 362 F.3d 1367 (Fed. Cir. 2004).....	5
<i>Lucent Techs., Inc. v. Gateway, Inc.</i> , 525 F.3d 1200 (Fed. Cir. 2008).....	9, 12
<i>Peer Commc'ns Corp. v. Skype Techs. SA</i> , No. 6:06-CV-370, 2008 U.S. Dist. LEXIS 92683	8
<i>Phillips v. AWH Corp.</i> , 415 F.3d 1303 (Fed. Cir. 2005).....	passim
<i>SuperSpeed Software, Inc. v. Oracle Corp.</i> , 447 F. Supp. 2d 672 (S.D. Tex. 2006)	5
<i>Telcordia Techs., Inc. v. Cisco Systems, Inc.</i> , Nos. 2009-1175, 2009-1184, 2010 WL 2653251 (Fed. Cir. July 6, 2010).....	13, 17
<i>Versa Corp. v. Ag-Bag Int'l Ltd.</i> , 392 F.3d 1325 (Fed. Cir. 2004).....	5
<i>VirnetX Inc. v. Microsoft Corp.</i> , No. 6:07-CV-80, 2009 U.S. Dist. LEXIS 65667	8, 11
STATUTES	
35 U.S.C. 112 ¶ 6.....	passim

Bedrock Computer Technologies LLC (“Bedrock”) respectfully submits its Claim Construction Brief regarding U.S. Patent No. 5,893,120 (“the ’120 patent” attached as Ex. A.1¹). This brief first introduces the technology of the patent, and then explains why Bedrock’s proposed constructions are proper.

I. TECHNOLOGY OVERVIEW

The ’120 patent is directed to “on-the-fly removal of expired data” in an information storage and retrieval system. *See* ’120::1:1-5.² Though the patent addresses a general problem related to providing constantly-available, high-performance storage/retrieval operations while, at the same time, handling expiring data, *see* ’120::2:22-26, the ’120 patent’s solution and claims are directed to a specific type of information storage and retrieval system—namely, one that (i) uses a hashing technique and (ii) uses external chaining. *See id.*

“Hashing” is the translation, via a mapping function (commonly known as hash or hashing function), of a record key value to a hash table array address. *See* ’120::1:34-46; *see also* ’120::4:53-62.³ Hashing is commonly used because it provides a very quick and efficient way to answer the question: Where in the table is the record with a particular key value? One problem, however, is that two different key values can often map to the same hash table address. This is known as a collision. *See id.*

Information storage and retrieval systems that use a hashing technique must handle collisions so that collided records are not lost and are still retrievable using the hash function. “External chaining” is one way to handle such collisions. *See* ’120:1:58-59; *see also* ’120::5:16-17. As used in the ’120 patent’s claims, the parties have agreed that “external chaining” is “a

¹ References herein to “Ex. A. _” are the numbered exhibits accompanying the Declaration of J. Austin Curry.

² Citations to “’120::xx:y-z” refer to the ’120 patent, col. xx, ll.y-z.

³ “Typical hashing functions include truncation, folding, transposition, and modulo arithmetic.” ’120:1:49-50; *see also* ’120::5:5-7.

technique for resolving hash collisions *using a linked list(s).*” See Dkt. No. 251 at 2 (Joint Claim Construction Statement) (emphasis added).

In a system that uses external chaining, a hash table entry does not contain the record itself; instead, the hash table entry contains a pointer to a linked list of records that have collided at that location. See ’120::1:59-6; *see also* ’120::5:16-20. Thus, when the information system is asked to store a new record, the record is inserted into a linked list that chains off of the hash table entry corresponding to the new record. But while accumulating collided records on linked lists solves the problem of hash collisions, the accumulation of records can, itself, become a problem.

Specifically, the ’120 patent teaches that some data records, after a limited period of time, become obsolete, and their presence in the storage system is no longer needed or desired. See ’120::2:7-10. If these expired records are not removed from the information system, they will seriously degrade the performance of the information system. See ’120::5:41-44. The ’120 patent describes two ways in which expired data can burden the performance of an information storage and retrieval system: (i) “the presence of expired records lengthens search times since they cause the external chains to be longer than they otherwise would be” and (ii) “expired records occupy dynamically allocated memory storage that could be returned to the system memory pool for useful allocation.” See ’120:5:44-49. As stated in the specification of the ’120 patent, the objective of the invention, then, is to “provide the speed of access of hashing techniques for large, heavily used information storage systems having expiring data and, at the same time, prevent the performance degradation resulting from the accumulation of many expired records.” See ’120::2:22-26.

Removal of unneeded records in an information system is commonly referred to as garbage collection, which often required that the system be taken off-line while the garbage was being collected. See ’120::2:64-67. In contrast, the garbage collection routines disclosed and claimed in the ’120 patent execute on-the-fly while other types of access to the linked lists take place. See ’120::2:54-63. As the patent emphasizes, “[t]his incremental garbage collection

technique has the decided advantage of automatically eliminating unneeded records without requiring that the information storage system be taken off-line for such garbage collection. This is particularly important for information storage systems requiring rapid access and continuous availability to the user population." *See* '120::2:64-3:3.

II. PRINCIPLES OF CLAIM CONSTRUCTION

Bedrock proposes constructions of the '120 patent in accordance with long-established principles of claim construction—giving a claim term its ordinary meaning that one of skill in the art, at the time of the invention and in light of the patent's specification and prosecution history, would have given it, except in two unusual circumstances: (1) where the intrinsic record provides a special definition for the term; or (2) where the patentee disclaims a portion of the term's ordinary meaning. *See, e.g., Phillips v. AWH Corp.*, 415 F.3d 1303, 1316–17 (Fed. Cir. 2005). “[A]lthough the specification often describes very specific embodiments of the invention, [the Federal Circuit has] repeatedly warned against confining the claims to those embodiments.” *Phillips*, 415 F.3d at 1323, citing *Nazomi Communications, Inc. v. ARM Holdings, PLC*, 403 F.3d 1364, 1369 (Fed. Cir. 2005). Limitations from the specification should not be read into the claims unless the patentee “acted as his own lexicographer and imbued the claim terms with a particular meaning or disavowed or disclaimed scope of coverage, by using words or expressions of manifest exclusion or restriction.” *E-Pass Techs., Inc. v. 3COM Corp.*, 343 F.3d 1364, 1369 (Fed. Cir. 2003) (citations omitted).

The '120 patent's specification is straight-forward, concise, and well-written. Despite a lack of support for their positions, Defendants seek to interject several limitations into the claims beyond the ordinary meaning of the terms as interpreted by one of ordinary skill in the art reading the patent's disclosure in light of the specification.

Because the Court is familiar with the law of claim construction, Bedrock will discuss specific claim construction principles only where applicable to each dispute.

III. LEVEL OF ORDINARY SKILL IN THE ART

Claims are to be construed from the viewpoint of a person of ordinary skill in the art. *Phillips*, 415 F.3d at 1313. The level of ordinary skill in the art is a function of many factors, including ““(1) the educational level of the inventor; (2) type of problems encountered in the art; (3) prior art solutions to those problems; (4) rapidity with which innovations are made; (5) sophistication of the technology; and (6) educational level of active workers in the field.”” *See Daiichi Sankyo Co. v. Apotex, Inc.*, 501 F.3d 1254, 1256 (Fed. Cir. 2007) (quoting *Envtl. Designs, Ltd. v. Union Oil Co.*, 713 F.2d 693, 696 (Fed. Cir. 1983)).

Considering these factors in the context of the technology of the ’120 patent, one of ordinary skill in the art of computer programming would have a Bachelor degree in a Computing Science, where the degree program requires extensive training and practice in computer programming. *See* Decl. of Dr. Mark Jones at ¶¶ 6-7.

IV. DISPUTED CLAIM CONSTRUCTIONS

A. Claims Terms Not Governed by 35 U.S.C. 112 ¶ 6

1. “linked list to store and provide access to records” / “linked list of records” [claims 1, 3, 5, and 7]

Bedrock’s Proposed Construction	Defendants’ Proposed Construction
a list in which each record contains a pointer to the next record or information indicating that there is no next record	two or more records in which each record contains a pointer to the next record in the list or information indicating that there is no next record

The specification of the ’120 patent discusses linked lists:

Such linked lists are formed by storing the records individually in dynamically allocated storage and maintaining with each record a pointer to the location of the next record in the chain of collided records.

’120::5:20-25. The specification also indicates that, when there is no next record, there will be information indicating as much. *See* ’120 at col. 11 (Search Table Procedure containing

psuedocode statement “while p ≠ nil”). In this way, Bedrock’s proposed construction is consistent with the intrinsic record and should be adopted by the Court.

Although the Defendants’ proposed construction includes most of the language proposed by Bedrock, the Defendants seek to include an extraneous limitation in the definition, namely the requirement that there be “two or more records” in a linked list. But nothing in the common meaning of the term “list” implies a requirement for multiple members. On the contrary, “list” is a common word that simply means “list.” *See SuperSpeed Software, Inc. v. Oracle Corp.*, 447 F. Supp. 2d 672, 682-83 (S.D. Tex. 2006).

Presumably, Defendants will emphasize the “to records”/“of records” language to support their plural members “list” construction. However, as the Federal Circuit has recognized, the plural form of a claim term can describe a universe ranging from one to some higher number, rather than requiring more than one item. *See Versa Corp. v. Ag-Bag Int’l Ltd.*, 392 F.3d 1325, 1330 (Fed. Cir. 2004). In *Versa*, the Federal Circuit looked to the “context in which the patentee used the plural” in determining whether the dispute term should be construed to require a plurality. *See id.* Here, the ’120 patent specifically contemplates the situation of a linked list containing only a single record:

The invoking procedure expects the remove procedure, on completion, to have advanced the passed pointer that originally pointed to the now-removed element so that it points to the successor element in that linked list, or NIL *if the removed element was the final element.*

’120::7:34-38 (emphasis added). Accordingly, the full context of “records” in “linked list of records” in the ’120 patent dictates that a linked list could have just one record. Also, the “linked list” referenced in the quote above ceased being a linked list when it had just one, final element. The Defendants’ proposed construction of “two or more” would thus read out this preferred embodiment, which offends a canon of claim construction. *See Globetrotter Software, Inc. v.*

Elan Computer Group, Inc., 362 F.3d 1367, 1381 (Fed. Cir. 2004) (“A claim interpretation that excludes a preferred embodiment from the scope of the claim is rarely, if ever, correct.”) (inner quotation omitted). The Defendants’ proposed construction for this term should therefore be rejected.

2. “automatically expiring / expired” [claims 1, 3, 5, and 7]

Bedrock’s Proposed Construction	Defendants’ Proposed Construction
after a limited period of time or after the occurrence of some event, becoming obsolete and therefore no longer needed or desired in the storage system / obsolete and therefore no longer needed or desired in the storage system	becoming obsolete and no longer needed or desired in the storage system because of some external condition

In each of the independent claims of the patent, the on-the-fly garbage collection takes aim at automatically expiring records. The specification of the ’120 patent discusses why automatically expiring records are problematic for information systems:

Some forms of information are such that individual data items, after a limited period of time, **become obsolete, and their presence in the storage system is no longer needed or desired.** Scheduling activities, for example, involve data that become obsolete once the scheduled event has occurred. An automatically-expiring data item, once it expires, needlessly occupies computer memory storage that could otherwise be put to use storing an unexpired item.

’120::2:7-11 (emphasis added). Bedrock’s proposed construction incorporates the specification’s explanation of what it means to “expire” (boldfaced above) and what it means to expire “automatically” (underlined above). Bedrock’s proposed construction for “automatically expiring” amalgamates these explanations in the following way: “after a limited period of time or after the occurrence of some event, **becoming obsolete and therefore no longer needed or desired** in the storage system.” The specification further supports this construction when it states that “records can become obsolete merely by the passage of time or by the occurrence of some event.” See ’120::5:38-41

The Defendants’ proposed construction suffers from a variety of problems. First, the Defendants’ proposed construction improperly engrafts a *test* for expiration that the ’120 patent uses in its preferred embodiment, *see* ’120::6:5-9, as the definition of expiration itself. Because there was no lexicography, no disavowal, and no disclaimer, this should be rejected. *See E-Pass Techs.*, 343 F.3d at 1369. Second, the Defendants’ proposed construction places two requirements on a record for it to be considered “expiring”—becoming obsolete *and* no longer needed or desired. The discussion of automatic expiration in the specification of the ’120 patent first equates “expired” with “obsolete” and then clarifies what it means to be expired or obsolete from the perspective of an information storage system. *See* ’120::2:7-11. The ’120 patent does not teach that a record must be obsolete *and* no longer needed or desired for it to be considered expired. Third, Defendants have put forth the same construction for “expired” and “automatically expiring.” At a minimum, this reads awkwardly when substituted into the claims. For example, “removing at least some of the expired ones of the records” becomes “removing at least some of the *becoming obsolete and no longer needed or desired in the storage system because of some external condition* ones of the records.” For these reasons, the Defendants’ proposed construction for this term should be rejected.

3. “identifying at least some of the automatically expired ones of the records” [claims 3 and 7]

Bedrock’s Proposed Construction	Defendants’ Proposed Construction
No construction necessary; however, should the Court construe this term: “identifying at least some of the automatically expired ones of the records when the linked list is accessed for a purpose other than garbage collection, using the same linked list traversal performed for the purpose other than garbage collection”	determining whether a record is expired by comparing some portion of the contents of the record to some external condition ⁴

⁴ Defendants indicated in their P.R. 4-3 chart that this limitation further applies to claims 1 and 5.

The goal of claim construction process is to explain the meaning of the claims to the jury. If a claim term “has an ordinary meaning that a jury would understand without construction” then no construction is necessary. *See VirnetX Inc. v. Microsoft Corp.*, No. 6:07-CV-80, 2009 U.S. Dist. LEXIS 65667, at **19-20 (E.D. Tex. July 30, 2009) (Davis, J.); *see also Peer Commc’ns Corp. v. Skype Techs. SA*, No. 6:06-CV-370, 2008 U.S. Dist. LEXIS 92683, at **9–10 (E.D. Tex. May 29, 2008) (Love, J.) (declining to construe term that the jury can understand). Here, once the term “automatically expired” is construed, the jury will understand the meaning of the phrase “identifying at least some of the automatically expired ones of the records.” To the extent the Court determines that construction of this term is necessary, Bedrock respectfully requests the Court adopt its construction, which merely clarifies that the referenced identification occurs “when the linked list is accessed for a purpose other than garbage collection, using the same linked list traversal performed for the purpose other than garbage collection.”

The Defendants’ proposed construction attempts to limit the “identifying” to one specific way in which an expired record could be identified, i.e., “by comparing some portion of the contents of the record to some external condition.” As support for their proposed construction, the Defendants’ rely on a discussion of a preferred embodiment. *See* ’120::6:5-20. Here, the patentee, Dr. Richard Nemes, was not acting as his own lexicographer. Nor did Dr. Nemes, in his discussion of the preferred embodiment, disavow or disclaim the ordinary scope of “identifying” to that embodiment. The Defendants’ proposed construction for this term, therefore, should be rejected. *See E-Pass Techs.*, 343 F.3d at 1369.

4. “removing at least some of the automatically expired records from the linked list when the linked list is accessed” [claims 3 and 7]

Bedrock’s Proposed Construction	Defendants’ Proposed Construction
No construction necessary; however, should the Court construe this term: “removing at least some	while traversing the linked list, both adjusting the pointers in the linked list to

of the automatically expired records from the linked list when the linked list is accessed for a purpose other than garbage collection, using the same linked list traversal performed for the purpose other than garbage collection”	bypass the previously identified expired records and de-allocating the memory occupied by those records
---	---

For the same reasons stated above, no construction is necessary for this claim term. Once the terms “automatically expired” and “linked list” are construed, the jury will understand the meaning of the phrase “removing at least some of the automatically expired records from the linked list when the linked list is accessed.”

The Defendants’ proposed construction attempts to confine “removing” to “while traversing the linked list,” by “adjusting the pointers in the linked list to bypass the previously identified expired records and de-allocating the memory occupied by those records.” Again, as support for their proposed construction, the Defendants’ rely on a discussion of a preferred embodiment. *See, e.g.,* ’120 at Figs. 3 and 4. These limitations from the specification should not be read into the claims because Dr. Nemes was not “act[ing] as his own lexicographer and imbu[ing] the claim terms with a particular meaning or disavow[ing] or disclaim[ing] scope of coverage, by using words or expressions of manifest exclusion or restriction.” *See E-Pass Techs.*, 343 F.3d at 1369.

Further, the prosecution history of the ’120 patent does not support the Defendants’ construction. A “clear and unmistakable disavowal of scope during prosecution may affect the construction of a claim term.” *Lucent Techs., Inc. v. Gateway, Inc.*, 525 F.3d 1200, 1211 (Fed. Cir. 2008) (citations omitted). Dr. Nemes made no disavowal that supports the Defendants’ proposed construction. Relevantly, Dr. Nemes distinguished the ’120 patent from the prior art in part by reiterating that the prior art does not remove records “when the linked list is accessed.” *See Ex. A.2 at BTEX0000367.* Plainly, the phrase “when the linked list is accessed” is not the

same as “while traversing the linked list.”⁵ As such, the Defendants’ proposed construction should be rejected.

5. “dynamically determining maximum number of expired ones of the records to remove when the linked list is accessed” [claims 4 and 8]

Bedrock’s Proposed Construction	Defendants’ Proposed Construction
determining, during program execution, maximum number of expired ones of the records to remove when the linked list is accessed	immediately before the linked list is traversed, determining a single number that serves as an upper limit on the number of records to remove as the linked list is traversed

Bedrock’s proposed construction merely clarifies that “dynamically” means “during program execution.” “Dynamic” is a term of art that has special meaning in the field of computer science, but it does not differ significantly from a lay understanding of the term, namely, the opposite of static. Bedrock recognizes that extrinsic evidence is “less significant than the intrinsic record in determining the legally operative meaning of claim language” *see Phillips*, 415 F.3d at 1317 (inner quotation removed); still, extrinsic evidence can be useful in understanding what one of ordinary skill in the art would have understood a term to mean at the time of the invention. The *New IEEE Standard Dictionary of Electrical and Electronic Terms* (5th ed. 1990) defines “dynamic” as “[p]ertaining to an event or process that occurs during computer program execution; for example, dynamic analysis, dynamic binding. Contrast with: static.” *See* Ex. A.3 (emphasis removed). In sum, one of ordinary skill in the art at the time of the invention would have understood that “dynamically” means “during program execution.”

The Defendants’ proposed construction attempts to confine “dynamically” to “immediately before the linked list is traversed.” As support for their proposed construction, the

⁵ Should the Court find disavowal of scope related to this disputed term, Bedrock respectfully requests the Court adopt its alternative construction, which essentially restates any such disavowed scope by clarifying that removal “when the linked list is accessed” is specifically a removal that occurs “when the linked list is accessed for a purpose other than garbage collection, using the same linked list traversal performed for the purpose other than garbage collection.”

Defendants' rely on a discussion of a preferred embodiment, *see* '120::6:56-7:15, and extrinsic evidence. But because Dr. Nemes was not acting as his own lexicographer, and further because Dr. Nemes did not disavow or disclaim the ordinary scope of "dynamically," the Defendants' attempt to limit this term to the preferred embodiment should be rejected. *See Phillips*, 415 F.3d at 1323; *see also E-Pass Techs.*, 343 F.3d at 1369. Moreover, the Defendants' extrinsic evidence does not support their proposed construction. The closest extrinsic evidence that the Defendants cite to support their construction is Microsoft Press's *Computer Dictionary* (3rd ed. 1997) (attached as Ex. A.4), which defines "dynamic" as: "[o]ccuring immediately and concurrently. The term is used in describing both hardware and software; in both cases it describes some action or event that occurs when and as needed." *See* Ex. A.4. This definition, however, does not support a construction that requires the "determination" in the disputed term to be "immediately before the linked list is traversed." To the contrary, the two definitions in Microsoft's *Computer Dictionary* immediately following the definition of "dynamic" provide examples of dynamic actions: "dynamic address translation" happens "when a program is run," and "dynamic allocation" happens "during program execution." *See* Ex. A.4. In this way, Defendants' extrinsic evidence actually supports Bedrock's proposed construction.

Further, the Defendants' proposed construction attempts to limit "dynamically determining maximum number of expired ones of the records" to ". . . determining a single number that serves as an upper limit on the number of records to remove. . . ." Once the Court construes the "dynamically" and "expired" aspects of this disputed claim term, the jury will understand the remainder of the term without construction. As such, no construction is necessary. *See VirnetX*, 2009 U.S. Dist. LEXIS 65667, at **19-20.

As support for their position, the Defendants rely on the prosecution history of the '120 patent, specifically, remarks made in reference to prior-art U.S. Patent No. 5,287,499 (the "'499 patent" attached as Exhibit A.5). As background, Dr. Nemes is also the inventor of, the '499 patent, which is directed to choosing between collision resolution techniques based on the number of records colliding to a particular hash table address. *See* '499::2:50-68. If the number of records colliding is below a certain upper threshold, a storage system practicing the '499 patent uses a collision technique called "linear probing." *See id.* If the number of collisions rises above that threshold, however, the records are collected from the linear probing technique and reorganized using the external chaining technique. *See id.* When the USPTO rejected the claims of the '120 patent because of the '499 patent, Dr. Nemes responded that the '499 patent's "threshold", which is actually a pair of upper and lower thresholds, acts as "two-way signals" in determining whether to use linear probing or external chaining in organizing the records. *See* Ex. A.2 at BTEX0000367. Dr. Nemes further explained that, in the '120 patent, the dynamically determined "maximum number" acts as "a single quantity that serves as an upper limit on the number of records removed." *See id.* In making this distinction, Dr. Nemes neither redefined the term "maximum number" nor provided any clear and unequivocal disavowal of claim scope. Rather, the remarks are merely part of an overall argument that the '499 patent's "threshold" has nothing to do with the "maximum number of expired ones of the records to remove when the linked list is accessed" of the '120 patent. As such, the Defendants' proposed construction should be rejected.⁶ *See Lucent*, 525 F.3d at 1211-1212 ("[S]tatements by the applicants must be

⁶ Defendants again attempt to limit the "removal" in this step to "as the linked list is traversed." For the same reasons discussed in section IV.A.4 of this brief, that aspect of the Defendants' proposed construction should be rejected.

read in the context of its overall argument distinguishing the claimed method from the method disclosed in [prior art].”)

B. Claim Terms Governed by 35 U.S.C. 112 ¶ 6

“In construing a means-plus-function claim, the district court must first determine the claimed function and then identify the corresponding structure in the written description of the patent that performs that function.” *See Baran v. Med. Techs., Inc.*, No. 2010 1058, 2010 WL 3178377, at *5 (Fed. Cir. Aug. 12, 2010) (citing *Applied Med. Res. Corp. v. U.S. Surgical Corp.*, 448 F.3d 1324, 1332 (Fed.Cir.2006)). In defining the claimed function, if the preamble of the means-plus-function claim is “not an idle description but a vital function to be performed” by the element, then the claimed function should be construed to include that preamble. *See id.* at *6. In identifying the corresponding structure, the absence of internal circuitry or code within a corresponding structure does not automatically render the claim indefinite; rather, “the specification need only disclose adequate defining structure to render the bounds of the claim understandable to an ordinary artisan.” *See Telcordia Techs., Inc. v. Cisco Systems, Inc.*, Nos. 2009-1175, 2009-1184, 2010 WL 2653251, at *10 (Fed. Cir. July 6, 2010).

1. “a record search means utilizing a search key to access the linked list” [claim 1]

Bedrock’s Proposed Recited Function	Defendants’ Proposed Recited Function
The <u>recited function</u> is record searching utilizing a search key to access the linked list.	(none disclosed)

2. “a record search means utilizing a search key to access a linked list of records having the same hash address [claim 5]

Bedrock’s Proposed Construction	Defendants’ Proposed Recited Function
The <u>recited function</u> is record searching utilizing a search key to access a linked list of records having the same hash address.	(none disclosed)
Bedrock’s Proposed Corresponding Structures (for both terms above)	Defendants’ Proposed Construction (for both terms above)
The <u>corresponding structure</u> is: (1) Portions of	Indefinite

the application software, user access software or operating system software, as described at col. 4, lines 30-48 and illustrated in FIG. 2, of a computer system that includes at least a CPU 10 and RAM 11, see FIG. 1 and col. 3 lines 52-56; and (2) Executable software instructions as illustrated in Boxes 31-36 and Boxes 39-41 of FIG. 3, or as portions of the pseudo-code of Search Table Procedure (cols. 11 and 12) or Alternate Version of Search Table Procedure (cols. 11, 12, 13, and 14), and described in col. 5, line 57-col. 6 line 4 and col. 6 lines 15-20, or the equivalents thereof.	
---	--

These claim terms are taken together because Bedrock proposes the same corresponding structures across both disputed terms, and the Defendants propose that both terms are indefinite. Bedrock's identified recited functions mirror, almost verbatim, the language found in the claim for these terms. The structure that performs these functions is the portions of the application software, user access software or operating system software, described at '120::4:30-48 and illustrated in FIG. 2, of a computer system that includes at least a CPU 10 and RAM 11 of FIG. 1 as described at '120::3:52-56, where the portion of the software includes executable software instructions (i) as disclosed in flow chart form:

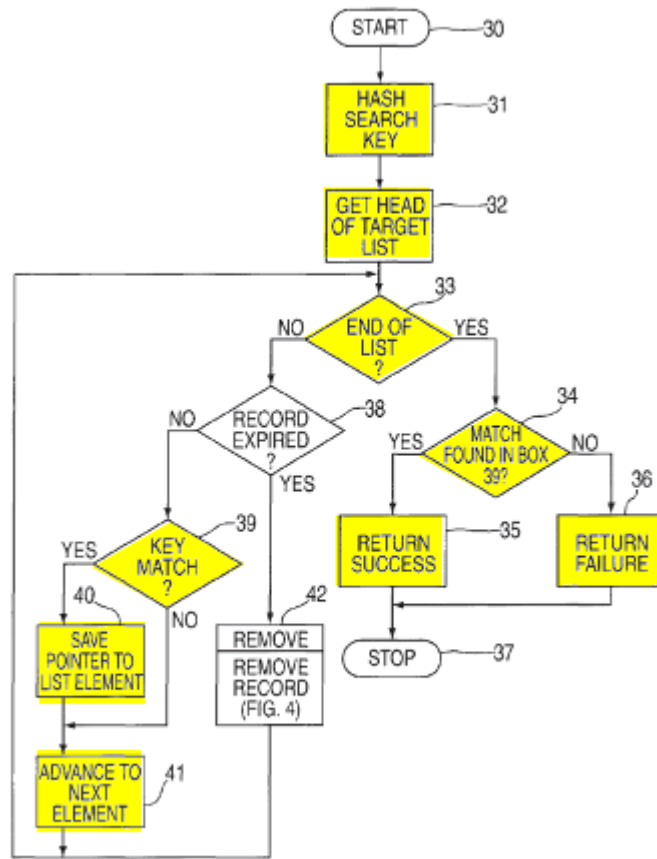


FIG. 3

(ii) as disclosed as pseudocode in Search Table Procedure:


```

end
end;
return (false)
end

/* else begin */
/* Record not found. */
/* search_table */

```

or the equivalents thereof. *See* 35 U.S.C. 112 ¶ 6. The specification provides a clear link between these terms and the corresponding structures above. *See* '120:: 5:57- 6:4 and 6:15-20.

The Defendant’s contention that these terms are indefinite is without merit. The '120 patent is presumed to be valid, and the Defendants bear the burden of proving that an ordinary artisan would not understand the bounds of the claim from the disclosed structure. *See Telcordia*, 2010 WL 2653251, at *10 (citing *Aero Prods. Int'l, Inc. v. Intex Rec. Corp.*, 466 F.3d 1000, 1015 (Fed. Cir. 2006)). This is a burden that Defendants cannot carry here. An ordinary artisan would understand the bounds of the disputed terms in light of the disclosed structures cited above. *See also* Decl. of Dr. Mark Jones at ¶¶ 10-13 and 31-32.

3. “the record search means including a means for identifying and removing at least some [of the] expired ones of the records from the linked list [of records] when the linked list is accessed” [claims 1 and 5]

Bedrock’s Proposed Construction	Defendants’ Proposed Construction
The <u>recited function</u> is record searching including identifying and removing at least some of the expired ones of the records from the linked list when the linked list is accessed.	<u>Function.</u> ⁷ identifying and removing at least some [of the] expired ones of the records from the linked list [of records] when the linked list is accessed.
The <u>corresponding structure</u> is: (1) Portions of the application software, user access software or operating system software, as described at col. 4, lines 30-48 and illustrated in FIG. 2, of a	Both identification and removal of an automatically expired record occurs during the same traversal of the linked list.

⁷ Before briefing, counsel for Bedrock sought clarification from the Defendants as to whether they intended that *all* sentences and phrases after “Function.” and before “Means.” comprise the recited function. The Defendants responded that all such sentences and phrases “relate to both the scope of the function and the overall claim limitations resulting from the specified structure. The defendants are proposing that the Court adopt the limitations in order to simplify the relevant issues for the jury.” *See* Ex. A.6 (email chain between A. Curry and counsel for the Defendants). The Court should not adopt this extraneous commentary in addition to determining the claimed function and then identifying the corresponding structure in the written description of the patent that performs that function. *See Baran*, 2010 WL 3178377, at *5.

<p>computer system that includes at least a CPU 10 and RAM 11, see FIG. 1 and col. 3 lines 52-56; and (2) Executable software as described in Boxes 33-42 of FIG. 3, and/or as pseudo-code in the Search Table Procedure (cols. 11 and 12) or Alternate Version of Search Table Procedure (cols. 11, 12, 13, and 14) including the lines “while ... /*HEART OF THE TECHNIQUE...”, and/or as described in col. 5, line 63 - col. 6, line 34, or the equivalents thereof.</p>	<p>For claim 1, the phrase “when the linked list is accessed” refers to the time during which the “utilizing a search key to access the linked list” function in limitation is carried out in claim 1.</p> <p>For claim 5, the phrase “when the linked list is accessed” refers to the time during which the “utilizing a search key to access a linked list of records having the same hash address” function is carried out in claim 5.</p> <p>Removing requires, while traversing the linked list, both adjusting the pointers in the linked list to bypass the previously identified expired records and de-allocating the memory occupied by those records.</p> <p><u>Means:</u> Boxes 10 and 11 of Fig. 1, Boxes 38 and 42 of Fig. 3, Fig 4 psuedocode in the Search Procedure (cols, 11-14) and Remove Procedure (cols. 13-14) and corresponding portions of the specification.</p> <p>The inclusions of “the records search means,” however, renders these limitations indefinite as the “record search means” limitation is indefinite.</p>
---	--

The recited function of this claim term is “record searching including identifying and removing at least some of the expired ones of the records from the linked list when the linked list is accessed.” This is a very close restatement of the claim language. In contrast, the Defendants attempt to import additional limitations—some of which are not even recast as language describing functionality—in an improper attempt to confine this term. The structure that performs the recited function is the portions of the application software, user access software or operating system software, described at ’120::4:30-48 and illustrated in FIG. 2, of a computer system that includes at least a CPU 10 and RAM 11 of FIG. 1 as described at ’120::3:52-56,

where the portion of the software includes executable software instructions (i) as disclosed in flow chart form:

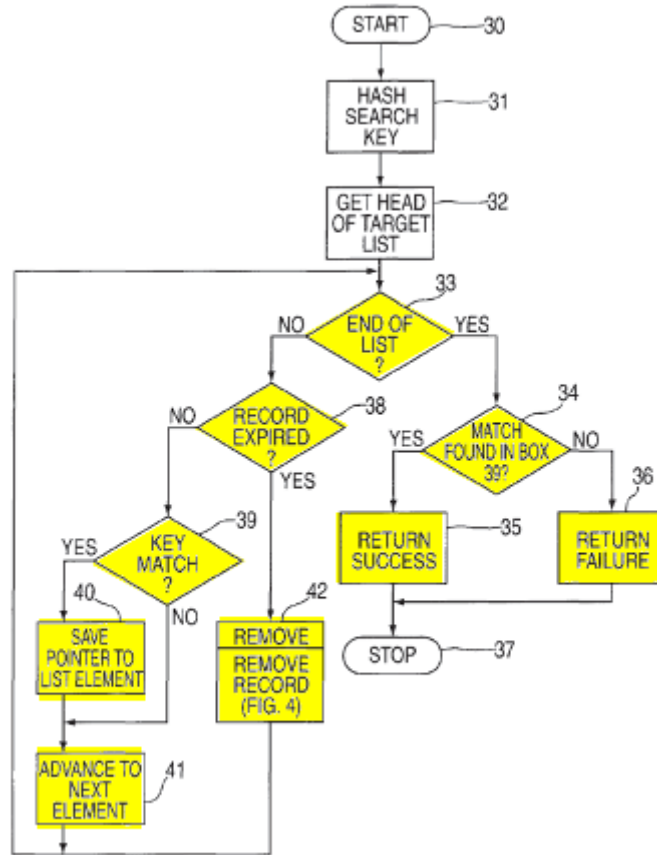


FIG. 3

(ii) as disclosed as pseudocode in Search Table Procedure:

Search Table Procedure

```
function search_table (record_key: record_key_type;
                      var position: list_element_pointer;
                      var previous_position: list_element_pointer;
                      var index: 0 . . . table_size - 1): boolean;
/* Search table for record_key and delete expired records in target list; if found, position is made to
   point to located record and previous_position to its predecessor, and TRUE is returned; otherwise
   FALSE is returned. index is set to table subscript that is mapped to by hash function in either
   case. */
var p: list_element_pointer; /* Used for traversing chain. */
    previous_p: list_element_pointer; /* Points to p's predecessor. */
begin
  index := hash (record_key); /* hash returns value in the range 0 . . . table_size - 1. */
  p := table[index]; /* Initialization before loop. */
  previous_p := nil; /* Ditto */
  position := nil; /* Ditto */
  previous_position := nil; /* Ditto */
  while p ≠ nil /* HEART OF THE TECHNIQUE: Traverse entire list, deleting */
                /* expired records as we search. */
  begin
    if p↑.record_contents is expired
    then remove (p, previous_p, index) /* ON-THE-FLY REMOVAL OF EXPIRED RECORD! */
    else begin
      if position = nil then if p↑.record_contents.key = record_key
      then begin position := p; previous_position := previous_p end; /* If this is record wanted, */
      /* save its position. */
      previous_p := p; /* Advance to */
      p := p↑.next /* next record. */
    end /* else begin */
  end; /* else begin */
  return (position ≠ nil) /* Return TRUE if record located, otherwise FALSE. */
end /* search_table */
```

(iii) as disclosed as psuedocode in the Alternate Version of Search Table Procedure:

Alternate Version of Search Table Procedure

```
function search_table (record_key: record_key_type;
                      var position: list_element_pointer;
                      var previous_position: list_element_pointer;
                      var index: 0 . . . table_size - 1): boolean;
/* SAME AS VERSION SHOWN ABOVE EXCEPT THAT THE SEARCH TERMINATES IF
   RECORD IS FOUND, INSTEAD OF ALWAYS TRAVERSING THE ENTIRE CHAIN. */
var p: list_element_pointer; /* Used for traversing chain. */
    previous_p: list_element_pointer; /* Points to p's predecessor. */
begin
  index := hash (record_key); /* hash returns value in the range 0 . . . table_size - 1. */
  p := table[index]; /* Initialization before loop. */
  previous_p := nil; /* Ditto */
  position := nil; /* Ditto */
  previous_position := nil; /* Ditto */
  while p ≠ nil /* HEART OF THE TECHNIQUE: Traverse list, deleting */
                /* expired records as we search. */
  begin
    if p↑.record_contents is expired
    then remove (p, previous_p, index) /* ON-THE-FLY REMOVAL OF EXPIRED RECORD! */
    else begin
      if p↑.record_contents.key = record_key /* If this is record wanted, */
      then begin /* save its position. */
        position := p;
        previous_position := previous_p;
        return (true) /* We found the record, so terminate search. */
      end;
      previous_p := p; /* Advance to */
      p := p↑.next /* next record. */
    end
  end
```



```

end
end;
return (false)
end

/* else begin */
/* Record not found. */
/* search_table */

```

or the equivalents thereof. *See* 35 U.S.C. 112 ¶ 6. The specification provides a clear link between these terms and the corresponding structures above. *See* '120::5:63-6:39.

Bedrock’s construction, which pairs the recited function with the corresponding structures disclosed in the specification, should be adopted, and the Defendants’ construction, which ignores disclosed structures for some functions yet includes unnecessary structure, should be rejected. *See also* Decl. of Dr. Mark Jones at ¶¶ 14-17.

4. “means, utilizing the record search means, for accessing the linked list and, at the same time, removing at least some of the expired ones of the records in the linked list” [claim 1]

Bedrock’s Proposed Construction	Defendants’ Proposed Construction
<p>The <u>recited function</u> is utilizing the record search means, accessing the linked list and, at the same time, removing at least some of the expired ones of the records in the linked list.</p> <p>The <u>corresponding structure</u> is: (1) Portions of the application software, user access software or operating system software, as described at col. 4, lines 30-48 and illustrated in FIG. 2, of a computer system that includes at least a CPU 10 and RAM 11, see FIG. 1 and col. 3 lines 52-56; and (2) Executable software which provides the insert, retrieve, or delete record capability illustrated in the flowchart of FIG. 5, FIG. 6, or FIG. 7, respectively, and/or as pseudo-code of Insert Procedure (cols. 9 and 10), Retrieve Procedure (cols. 9, 10, 11, and 12), or Delete Procedure (cols. 11 and 12), respectively, and/or described in col. 7, line 65 - col. 8, line 32, col. 8, lines 33-44, or col. 8 lines 45-59, or the equivalents thereof.</p>	<p><u>Function</u>: using the records search means defined above, accessing and during the same traversal of the linked list removing at least some of the expired ones of the records in the linked list.</p> <p>“at the same time” means during the same traversal of the linked list.</p> <p>This limitation requires that the referenced means remove at least one of the expired records in the linked list while utilizing a search key to access the linked list.</p> <p>Removing requires, while traversing the linked list, both adjusting pointers in the linked list to bypass the previously identified expired records and de-allocating the memory occupied by those records.</p> <p><u>Means</u>: Boxes 10 and 11 of Fig. 1; Figs. 4, 5, 6, and 7, pseudocode in the Search Procedure (cols. 11-14), Insert Procedure (cols. 9 and 10), Retrieve Procedure (cols. 9 and 10), Delete Procedure (cols. 11-12), and Remove Procedure (cols. 13-14), and corresponding</p>

	<p>portions of the specification. Inserting, retrieving, and deleting are all required.</p> <p>The inclusion of “utilizing the records search means,” however, renders this limitation indefinite as the “record search means” limitation is indefinite.</p>
--	--

5. “mea[n]s, utilizing the record search means, for inserting, retrieving, and deleting records from the system and, at the same time, removing at least some expired ones of the records in the accessed linked list of records’ [claim 5]

Bedrock’s Proposed Construction	Defendants’ Proposed Construction
<p>The <u>recited function</u> is utilizing the record search means, inserting, retrieving, and deleting records from the system and, at the same time, removing at least some expired ones of the records in the accessed linked list of records.</p> <p>The <u>corresponding structure</u> is: (1) Portions of the application software, user access software or operating system software, as described at col. 4, lines 30-48 and illustrated in FIG. 2, of a computer system that includes at least a CPU 10 and RAM 11, see FIG. 1 and col. 3 lines 52-56; and (2) Executable software which provides the insert, retrieve, and delete record capability illustrated in the flowchart of FIG. 5, FIG. 6, and FIG. 7, respectively, and/or as pseudo-code of Insert Procedure (cols. 9 and 10), Retrieve Procedure (cols. 9, 10, 11, and 12), and Delete Procedure (cols. 11 and 12), respectively, and/or described in col. 7, line 65 - col. 8, line 32, col. 8, lines 33-44, and col. 8 lines 45-59, or the equivalents thereof.</p>	<p><u>Function:</u> Using the record search means defined above, inserting, retrieving, and deleting during the same traversal of the linked list removing at least some expired ones of the records in the accessed linked list of records.</p> <p>“at the same time” means during the same traversal of the linked list.</p> <p>This limitation requires that the referenced means remove at least one of the expired ones of the records in the linked list while utilizing a search key to insert, retrieve, and delete records having the same hash address from the system.</p> <p>Removing requires, while traversing the linked list, both adjusting the pointers in the linked list to bypass the previously identified expired records and de-allocating those records from memory.</p> <p><u>Means:</u> Boxes 10 and 11 of Fig. 1; Figs. 4, 5, 6, and 7, psuedocode in the Search Procedure (cols. 11-14), Insert Procedure (cols. 9 and 10), Retrieve Procedure (cols. 9 and 10), Delete Procedure (cols. 11-12), and Remove Procedure (cols. 13-14), and corresponding portions of the specification. Inserting, retrieving, and deleting are all required.</p> <p>The inclusion of “utilizing the records search means,” however, renders this limitation indefinite as the “record search means”</p>

limitation is indefinite.

These claim terms are taken together because Bedrock proposes the same structures across the two claim terms. The difference between Bedrock’s proposed constructions is that, in the first construction above, the identified structures are combined in disjunctive form, i.e., with the conjunction “or,” but in the second construction above, the identified structures are combined in the conjunctive form, i.e., with the conjunction “and.” Bedrock’s proposed recited functions are close restatements of the claim language. In contrast and again, the Defendants attempt to import additional limitations—some of which are not even recast as language describing functionality—in an improper attempt to confine this term.

The structures that performs the recited function in the second term above are the portions of the application software, user access software or operating system software, described at ’120::4:30-48 and illustrated in FIG. 2, of a computer system that includes at least a CPU 10 and RAM 11 of FIG. 1 as described at ’120::3:52-56, where the portion of the software includes executable software instructions which provide the insert, retrieve, and delete record capability illustrated in the flowchart of FIG. 5, FIG. 6, and FIG. 7, respectively, and/or as pseudo-code of Insert Procedure (cols. 9 and 10), Retrieve Procedure (cols. 9, 10, 11, and 12), and Delete Procedure (cols. 11 and 12) or the equivalents thereof. With respect to the first term above, any one of the insert, retrieve, or delete structures can serve as the corresponding structure, and so there the identified structures are joined together as above, but with the conjunction “or.” The specification provides a clear link between these terms and the corresponding structures above. *See* ’120::7:65- 8:32, 8:33-44, and 8:45-59. Bedrock’s construction, which pairs the recited function with the corresponding structures disclosed in the specification, should be adopted, and the Defendants’ construction, which ignores disclosed structures for some functions yet includes

unnecessary structure, should be rejected. *See also* Decl. of Dr. Mark Jones at ¶¶ 18-21 and 28-30.

6. “a hashing means to provide access to records stored in a memory of the system and using an external chaining technique to store the records with same hash address, at least some of the records automatically expiring” [claim 5]

Bedrock’s Proposed Construction	Defendants’ Proposed Construction
<p>The <u>recited function</u> is using hashing to provide access to records stored in a memory of the system and using an external chaining technique to store the records with same hash address, at least some of the records automatically expiring.</p> <p>The <u>corresponding structure</u> is: (1) Portions of the application software, user access software or operating system software, as described at col. 4, lines 30-48 and illustrated in FIG. 2, of a computer system that includes at least a CPU 10 and RAM 11, see FIG. 1 and col. 3 lines 52-56; and (2) Executable software instructions corresponding to pseudo-code “var table: array [0 . . . table_size - 1] of list_element_pointer /* Hash table.*/” which point to records of type “list_element” in cols. 9-10 that allocates in memory an external chaining hash table, and/or as described in col. 5, lines 16-41, or the equivalents thereof.</p>	<p>Indefinite</p>

The recited function is “using hashing to provide access to records stored in a memory of the system and using an external chaining technique to store the records with same hash address, at least some of the records automatically expiring.” The Defendants have proposed no recited function. The structure that performs this function is the portions of the application software, user access software or operating system software, described at ’120::4:30-48 and illustrated in FIG. 2, of a computer system that includes at least a CPU 10 and RAM 11 of FIG. 1 as described at ’120::3:52-56, where the portion of the software includes executable software instructions as disclosed in the pseudocode:

Definitions

The following formal definitions are required for specifying the insertion, retrieval, and deletion procedures. They are global to all procedures and functions shown below.

```

1. const table_size                                     /* Size of hash table. */
2. type list_element_pointer = ↑ list_element          /* Pointer to elements of linked list. */
3. type list_element =                                 /* Each element of linked list. */
    record
        record_contents: record_type;
        next: list_element_pointer                    /* Singly-linked list. */
    end
4. var table: array [0 .. table_size - 1] of list_element_pointer /* Hash table. */
    /* Each array entry is pointer to head of list. */
    Initial state of table: table[i] = nil ∀ i 0 ≤ i < table_size /* Initially empty. */

```

or the equivalents thereof. The specification provides a clear link between this term and the corresponding structures above. *See* '120::1:34-2:5; *see also* '120::4:53-5:33.

The Defendant’s contention that this term is indefinite is without merit. An ordinary artisan would understand the bounds of this disputed term in light of the disclosed structures cited above. *See also* Decl. of Dr. Mark Jones at ¶¶ 25-27.

7. “means for dynamically determining maximum number for the record search means to remove in the accessed linked list of records [claims 2 and 6]”

Bedrock’s Proposed Construction	Defendants’ Proposed Construction
<p>The <u>recited function</u> is dynamically determining maximum number of records for the record search means to remove in the accessed linked list of records.</p> <p>The <u>corresponding structure</u> is: (1) Portions of the application software, user access software or operating system software, as described at col. 4, lines 30-48 and illustrated in FIG. 2, of a computer system that includes at least a CPU 10 and RAM 11, see FIG. 1 and col. 3 lines 52-56; and (2) Executable software, as described in col. 6, line 56 - col. 7, line 15, that dynamically chooses among removal strategies (e.g., chooses whether to execute Search Table Procedure [cols. 11-12] or Alternate Version of Search Table Procedure [cols. 11-14]) “at the time the record search means is invoked by the caller, thus sometimes removing all expired records, at other times removing some but not all of them, and yet at other times choosing to remove none of them. Such a dynamic decision can be based on factors</p>	<p>Indefinite</p>

such as, for example, how much memory is available in the system storage pool, general system load, time of day, the number of records currently residing in the information system, and other factors both internal and external to the information storage and retrieval system itself” (col. 7, lines 1-10), or the equivalent thereof.	
--	--

The recited function of this disputed term is “dynamically determining maximum number of records for the record search means to remove in the accessed linked list of records.” This is a close restatement of language in the claims. The Defendants have proposed no recited function. The structure that performs the recited functions is the portions of the application software, user access software or operating system software, described at ’120::4:30-48 and illustrated in FIG. 2, of a computer system that includes at least a CPU 10 and RAM 11 of FIG. 1 as described at ’120::3:52-56, where the portion of the software includes executable software instructions which, at the time the record search means is invoked by the caller, chooses whether to execute the Search Table Procedure or the Alternate Version of Search Table Procedure. See ’120::6:56-7:15. This choice is based on factors such as how much memory is available in the system storage pool, general system load, time of day, the number of records currently residing in the information system. See *id.*

The Defendant’s contention that these terms are indefinite is without merit. To be sure, there is no flowchart or pseudocode in the ’120 patent that entirely discloses the corresponding structure for this claim term. There is, however, an algorithm specified in prose form in the specification, which is just as capable a structure. See *Finisar Corp. v. DirecTV Group, Inc.*, 523 F.3d 1323, 1340 (Fed. Cir. 2008) (“This court permits a patentee to express that 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.”) (inner citation removed). *See also* Decl. of Dr. Mark Jones at ¶¶ 22-24.

C. Miscellaneous Construction Issues

1. Ordering of the claims [claims 3 and 7]

Bedrock proposes that the steps of claims 3 and 7 require no construction. The Defendants propose a rigid ordering of the steps. The Federal Circuit has been clear that “[u]nless the steps of a method actually recite an order, the steps are not ordinarily construed to require one.” *See Altiris v. Symantec Corp.*, 318 F.3d 1363, 1369 (Fed. Cir. 2003). The *Altiris* Court went on to hold that the test for determining whether the steps of a method claim that do not recite an order must nonetheless be performed in an order is to look at the logic and grammar of the claim as well as the intrinsic record. *See id.* Here, neither logic nor grammar requires a specific ordering of the steps of claims 3 and 7, with the exception that a first “identifying” step would begin performing before the first “removing” step could be completed. The jury, however, would understand this without construction. Further, the steps of claims 3 and 7 can be performed in a consecutive, repeating, and/or overlapping manner without offending the logic or grammar of the claim. The Defendants’ proposed ordering, however, does not permit these possibilities. As such, the Defendants’ proposed ordering should be rejected.

2. “identifying . . . and removing . . . when the linked list is accessed” [claims 3 and 7]

The Defendants propose a construction for “identifying . . . and removing . . . when the linked list is accessed” separately from the “identifying” and “removing” terms of claims 3 and 7. This suffers from a variety of problems. First, this is not a claim term; these are two claim terms glued together with ellipses. Second, after claims terms are construed, the claims should be capable of being rewritten with the constructions substituted for the terms. It is not clear, then, how the claims would be rewritten since the Defendants also want the “identifying” and

“removing” claim terms construed separately. Also, the Defendants’ proposed construction does not make sense when substituted into the claims. Taking claim 3 as an example:

accessing the linked list of records,
identifying at least some of the automatically expired ones of the records, and
removing at least some of the automatically expired records from the linked list when the linked list is accessed

would read:

accessing the linked list of records,
both identification and removal of the automatically expired record(s) occurs during the same traversal of the linked list.

Finally, the requirement that these steps occur “during the same traversal” is not proper for the same reasons given in section IV.A.4 of this brief. As such, the Defendants’ proposed construction should be rejected.

V. CONCLUSION

For the foregoing reasons, Bedrock respectfully requests that the Court adopt Bedrock’s proposed constructions of the disputed claim terms of the ’120 patent, and refuse Defendants’ repeated invitations to alter the claim language and import limitations from the preferred embodiments.

DATED: August 27, 2010

Respectfully submitted,
McKOOL SMITH, P.C.

/s/ Douglas A. Cawley

Sam F. Baxter
Texas Bar No. 01938000
McKOOL SMITH, P.C.
sbaxter@mckoolsmith.com
104 E. Houston Street, Suite 300
P.O. Box 0
Marshall, Texas 75670
Telephone: (903) 923-9000
Facsimile: (903) 923-9099

Douglas A. Cawley, Lead Attorney
Texas Bar No. 04035500
dcawley@mckoolsmith.com
Theodore Stevenson, III
Texas Bar No. 19196650
tstevenson@mckoolsmith.com
Jason D. Cassady
Texas Bar No. 24045625
jcassady@mckoolsmith.com
J. Austin Curry
Texas Bar No. 24059636
acurry@mckoolsmith.com
McKOOL SMITH, P.C.
300 Crescent Court, Suite 1500
Dallas, Texas 75201
Telephone: 214-978-4000
Facsimile: 214-978-4044

Robert M. Parker
Texas Bar No. 15498000
Robert Christopher Bunt
Texas Bar No. 00787165
PARKER, BUNT & AINSWORTH, P.C.
100 E. Ferguson, Suite 1114
Tyler, Texas 75702
Telephone: 903-531-3535
Facsimile: 903-533-9687
E-mail: rm_parker@pbatyler.com
E-mail: rcbunt@pbatyler.com

**ATTORNEYS FOR PLAINTIFF
BEDROCK COMPUTER
TECHNOLOGIES LLC**

CERTIFICATE OF SERVICE

I hereby certify that all counsel of record who are deemed to have consented to electronic service are being served with a copy of the forgoing document via the Court's CM/ECF system pursuant to the Court's Local Rules this 27th day of August, 2010.

/s/ J. Austin Curry _____
J. Austin Curry