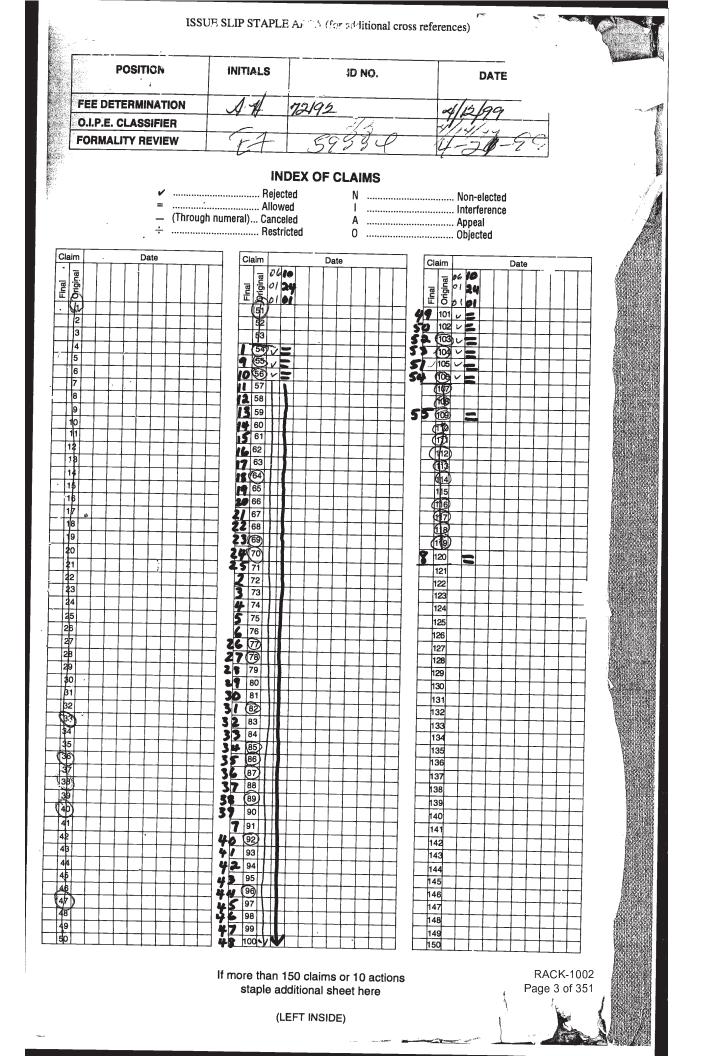
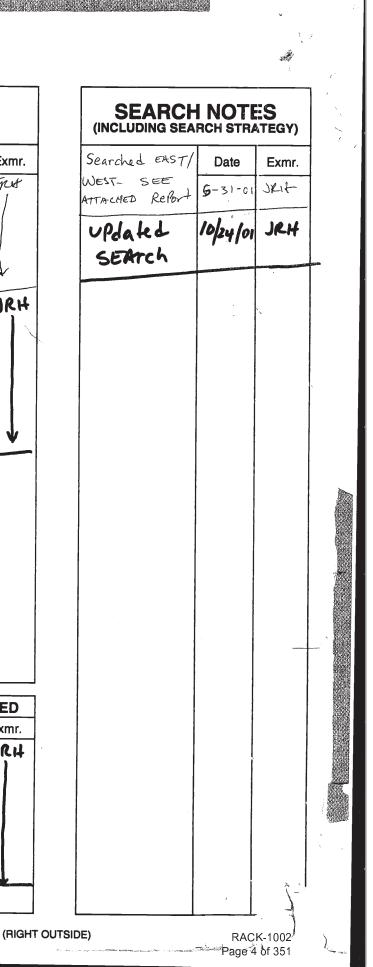
66/TO/BU	P3 2 Class S. briass Class S. briass IISSUE CLASSIFICATION	- <u>U.S.</u> 1		<u>ч РАТЕ</u>			ON	6415280
		SCANN	Ing	9.1.P.E.	Bui	JUL 0		
SECTOR	CLASS 707	SUBCLA	ASS Â.	AR	T UNIT	FILED WIT		K (CRF) FIC
	PF	REPARE	DAND	APPRO	OVED FO	OR ISS	UE	
		ISS	UING	CLASS	FICATIO	ON	NUC X	
					CHOSS I			
CLASS 707			D7	3	10	101	ASS PEN DL	
	IONAL CLASSIFICAT		09	203	219	229		
406F	17 / 30				ļ			
					-			
						Continued	on Issue Slip	Inside File Jacket
	and the second			а ом	<u>.</u>			· · ·
	MINAL CLAIMER			RAWINGS			×	S ALLOWED
		Sheets Di	wg. F	igs. Drwg.	Print Fig.		Claims	Print Claim for
		<u> ~7</u>	5/1	JO	IA, 1B	-		OWANCE MAILE
subsequent	erm of this patent to (date)							
has been dis	erm of this patent shall	A)	Assistant Exan	mmer)	(Date)	1 10	5-2.5-	01
hot extend b	beyond the expiration date $178,791$. J		OMERE			ISSI	
		PH		EXAMINER		8	ount Due	Date Paid
		Jean	Primary Exam		: <u>10/24</u>	\$64	0.00	11-21-01
			AIN.	n site and the second	-, i /		SSUE BAT	TCH NUMBER
an o por provanta	urminal months of		111. 11	ife	10/260		299	7
c) The te	erminalmonths of ave been disclaimed.	(Legia	I Instruments	Examiner)	(Date)			
c) The te this patent h WARNING:	ave been disclaimed. ; n disclosed herein may be r	restricted. Unaut	horized disc	closure may be	(Date)	e United State	es Code Title 3	5, Sections 122, 181 a
C) The te this patent h WARNING : The informatio Possession ou	ave been disclaimed. n disclosed herein may be r tside the U.S. Patent & Trad	restricted. Unaut	horized disc	closure may be	(Date)	e United State actors only.	es Code Title 3	5, Sections 122, 181 a
C) The te this patent h WARNING: The informatio Possession ou Form PTO-436A	ave been disclaimed. n disclosed herein may be r tside the U.S. Patent & Trad	restricted. Unaut lemark Office is ro	horized disc estricted to a	closure may be authorized empl	byees and conti	e United State ractors only.	as Code Title 3	5, Sections 122, 181 a
c) The te this patent h WARNING:	ave been disclaimed. n disclosed herein may be r tside the U.S. Patent & Trad	restricted. Unaut lemark Office is ro	horized disc estricted to a	closure may be	byees and contra	e United State actors only.		

	I libra i fi	и. 5. рто 283160	-A
PATENT APPLIC			INI. IALS
09283160	CCN	TENTS	
	Date raceived (Incl. C. of M.)		Date received (incl. C. of M.)
,	or Date Mailed		or Date Mailed
1. Application pap		42	
2. Pro andt A		-	<u>.</u> v
3. fre andt B			, <u>,</u>
14. Petition Inpre	wige 3-02-01		· · ·
5. Petitus Drind	4/18/01		· ·
6	· 115/157		· · · · · · · · · · · · · · · · · · ·
147. <u>Rej (31005)</u>	1 8/2.2/01		
8. Unal - TDS	8/22/01		
9. ten the second Discl	19:mes 10-9-0(۰
11. Supp. Amet D	10/10/01	52	
12. Drawing Correct		53	·
273 AllonAnce	10-25-01	54	
14. Communication	11/2/01		
15. Formal Drawing			/
12 to. Mire Affice Action	n <u>1-25-02</u>	57	· · · · · · · · · · · · · · · · · · ·
Drawning (3)	11/2/10/		
18. Letter Re. Lik	, 2/15/02		. 4
19		60 61	· ,
20		61 62	
21			
22 23			
23			
24			
26		67	·
27		68	
28		69	
29			
30	•		
31		-	<u>\$</u>
32			
33			/
. 34			
35 36			
36 37			·
37 38			
39	1		·
~ 40 /			
		82	RACK-1002
	4.5	EFT OUTSIDE)	Page 2 of 351



	SEARCHED					
·	Class	Sub.	Date	Exmr.		
	707	3 10 101 2 203 219 229	5-31-01	Trut		
	707	2 3 10 101 203 219 229	10/24/0	JRH		
	;					
F			ESEAR			
$\left \right $	Class	Sub.	Date	Exmr.		
	707 , 709 ,	2 3 10 101 203 219 229	10/24/01	JRH		



Page 1 of 1

	ed States Patent and	Trademark Office	United		MISSIONER FOR PATENTS AND TRADEMARK OFFICE WASHINGTON, D.C. 2023 www.usplo.gov
Bib Data Sheet				CONFIRM	IATION NO. 9574
SERIAL NUMBE 09/283,160	FILING DATE 04/01/1999 RULE	CLASS 707	GROUP AR 2177		ATTORNEY DOCKET NO. PM252465
RONALD D	ARBER, OJAI, CA; LACHMAN, NORTHBRC				- 21 (orthout) - 2
IF REQUIRED, FO	LICATIONS ************************************		ENTITY **		
GRANTED ** 04/2 Foreign Priority claimed 35 USC 119 (a-d) cond met Verified and Acknowledged	Itions U yes to no Allowance Allowance	ter COUNTRY CA	SHEETS DRAWING 24	TOTAL CLAIMS 5	INDEPENDENT CLAIMS 3
ADDRESS PILLSBURY MAD INTELLECTUAL I 1100 NEW YOUK NINTH FLOOR E WASHINGTON J	PROPERTY GROUP (AVENUE NW AST TOWER			4	
TITLE IDENTIFYING DA CONTENTS OF I	TA REQUESTING DATA			S WHICH A	RE BASED ON
RECEIVED	EES: Authority has been (No to charge/c No for following	redit DEPOSIT ACCC	DUNT		ocessing Ext. of

RACK-1002 Page 5 of 351

PATENT APPLICATION SERIAL NO.

U.S. DEPARTMENT OF COMMERCE PATENT AND TRADEMARK OFFICE FEE RECORD SHEET

MADH 6-0-10

14/12/1999 ANAYES 00000010 09283160 11 FC:201 380.00 DP

PTO-1556 (5/87) *U.S. GPO: 1998-433-214/80404

6

ł

·

			REQUEST	ENT AND TRADEN <u>FOR FILING</u> 53(b)(1))	MARK OFFICE
Rule		ENT APPLICATION:	<u>For Design or L</u>	Itility Applications	
	Continuatio Divisional		er 37 CFR 1.53(b)	(1)	9/28 9/28
4)		Group Art Unit: 2	2776
	-	prior application of		Examiner: Home	ere, J.
	ntor(s): FA nt Appln. No.:	ARBER et al. 08	960,079	Atty. Dkt. PM 25	2465
'his (Case Filed:	Series Code ☆ October 24, 1997 April, 1999	û Serial No.	Our Deposit Accor (Our Order No. 7	w M# Client Ref
itle:	IDENTIFYI	NG DATA IN A DATA	PROCESSING SY		C# / <u>New</u> M#
				Date: April 1,	1999
.sst. /ash	Commissione nington, DC 20	r of Patents 231		(Parent Matter No.	· · · · ·
ir:				(i alon mater 140.)
	Тс	effect the above-requ	ested filing today:		
		a copy (<u>which must b</u>			
	_		<u>e mee</u> y of this upp	neadon, meluding.	
	Abstrac	ation and claims (94 p	ages) (<u>must</u> be a	ttached)	
		s (must be attached	if originally filed)	: <u>24</u> sheet(s)/set: ⊠ 1 ☐ Formal of si	set informal; ize ☐ A4 ┌┐ 11"
A. (1)	Always X on Signed	declaration or oath as	originally filed in p	rior application attache	ed
(2)		laration or fee is enclose	sed; therefore, this	s is a filing under Rule	53(f).
	nerep	y made requesting del	etion as inventor(s	of the inventors name b) of the following who	d in the prior application. Petition is is/are not inventor(s) of the
	inven	ion being claimed in th	is application:	-	
	1. 3.			2. 4.	
	5.			6.	
	7			8.	
	invent 1. 3.	ion being claimed in th	etion as inventor(s is application:	2. 46	d in the prior application. Petition is/are <u>not</u> inventor(s) of the
	7			8.	
	7. 9.			10.	

		۵	pplication N				(co	untry)	of
	(1)				Filing Date	(4)	Application No	<u>0.</u>	Filing Date
	(2)					(5)			
	(3)					(6)			
		a. 🗌		(No.) Ce	ertified copy/copies	ottoohad			
		b. 📋	Certified c	opy/copies pr	eviously filed on / code û û <u>ser</u>			i	'n
			0.5. Ap	oplication No.			, filed on		·
			Certified co	opy/copies file	ed during Internation	nal stage	of PCT/	1	
4.	(a) (b)	Don Don	nestic priorit	ly is claimed fi	rom PCT//		filed		•
	(0)	Ber	netit is claim	ned of Provisio	onal Application No.	60/	, filed	_ ·	
5.	\boxtimes				kiNETech, Inc.				
	by as	signment	recorded	June 23, 19	95 (Date)	Reel	7593	Frame	0036.
6.	\boxtimes				(Date) of Assignments (in				
		differen	t assignors)): <u>1</u> and	respective <u>new</u> Co	over Shee	ets. (Do NO T f	later succ	ver sheets)
									<i>:</i>
		(Assign want it/t	ments in pa	rent <u>must be</u> ed against the	refiled with new Co continuing applica	over Shee	ets in this conti	nuing ap	plication if you
	Please				to the undersigned.				
-	_								
7.	\boxtimes	i ne po	wer of attor	ney in the pric	or application is to <u>D</u>	ale S. La	zar, Reg. No.	<u>28.872</u>	
	(Name	and Reg	g. No.)						
				as in item 8 be					
	а. 🖂	Recogr	nize as asso	ociate attorney	/ Brian Siritzky, Reg	. No. 37,	497		
			o. and Addr						
				•					
0	Aaare	ss all fui shurv Me	ture commi adison & Si	unications to utro LLP. Nin	Intellectual Prope th Floor, East Tow	rty Grou	p Now York Ave		
8.	of Pills			3918	and loon, East 104		New TURAV	enue, N.	/ v .,
8.		ington, D).C. 20005-3						
8. 9.	Of Pills Washi	end the	specificatio	on by insertin	a before the first line	e the sen	tence This is	а	
	Of Pills Washi	end the	specificatio	on by insertin	g before the first line ication No08/960	D,079,	filed	a <u>Octob</u>	per 24. 1997
	of Pills Washi ⊠ Am ⊡ con	ngton, D end the itinuation	specificatio	on by insertin on of Appl	series code fr	0,079,	filed	Octob	per 24. 1997
9.	Washi	ington, D iend the stinuation is a contin	specification ⊠ division nuation of 0	on by insertin on of Appl 8/425,160, file	ication No. <u>08/96(</u> <u>series code</u> û ed April 11, 1995, no	0,079, ☆ <u>seria</u> ow abanc	filed I <u>l no.</u> Ioned.	<u>Octob</u>	-
8. 9. 9.	Washi	ngton, D nend the stinuation is a contin Amend t	specification	on by insertin on of Appl 18/425,160, file cation by inse	ication No. <u>08/960</u> <u>series code</u> û ed April 11, 1995, no arting before the firs	0,079, ☆ <u>seria</u> ow abanc	filed Il no. Ioned.	<u>Octob</u> 	-
9. 9.	of Pills Washi	ngton, D eend the s tinuation is a contin Amend s Provision	specification division nuation of 0 the specific nal Applicat	on by insertin on of Appl 8/425,160, file cation by inse ion No. 60/	ication No. <u>08/960</u> <u>series code</u> û ed April 11, 1995, no erting before the firs , filed	0,079,	ll no. loned. his application	Octob claims th	- ne benefit of
9.	of Pills Washi	ngton, D end the s itinuation is a contin Amend f Provision has been	specification implication of 0 the specific nal Applicat n recently de	on by insertin on of Appl 8/425,160, file cation by inse ion No. 60/	ication No. <u>08/960</u> <u>series code</u> û ed April 11, 1995, no arting before the firs	0,079,	ll no. loned. his application	Octob claims th	- ne benefit of
9. 9.	of Pills Washi	ngton, D nend the s tinuation is a contin Amend f Provision has been Hence: (No.)	specification division nuation of 0 the specific nal Applicat n recently de Verified St	on by insertin on of Appl 18/425, 160, file cation by inse tion No. 60/ etermined that tatement(s) es	ication No. <u>08/96(</u> <u>series code</u> îr ed April 11, 1995, no erting before the firs , filed, t this new continuing stablishing "small en	0,079,	tino. Ioned. his application	<u>Octob</u> claims th to small e	- ne benefit of entity status.
9. 9.	of Pills Washi	ngton, D nend the s tinuation is a contin Amend f Provision has been Hence: (No.)	specification division nuation of 0 the specific nal Applicat n recently de) Verified St in above pr	on by insertin on of Appl 18/425, 160, file cation by inse tion No. 60/ etermined that tatement(s) es	ication No. <u>08/960</u> <u>series code</u> û ed April 11, 1995, no erting before the firs , filed	0,079,	tino. Ioned. his application	<u>Octob</u> claims th to small e	- ne benefit of entity status.
9. 9. 10.	of Pilit Washi □ con which i (a) □ □ It	ngton, D end the tinuation is a contii Amend Provision has beer Hence: (No.) ∑ filed ∑ attac	specification division nuation of 0 the specific nal Applicat n recently de) Verified St in above pr ched.	on by insertin on of Appl 8/425,160, file cation by inse tion No. 60/ etermined that tatement(s) es ior application	ication No. <u>08/96(</u> <u>series code</u> î) ed April 11, 1995, no erting before the firs , filed t this new continuing stablishing "small en a (and hence applica	0,079, <u>r seria</u> <u>seria</u> <u>seria</u> <u>seria</u> <u>seria</u> <u>seria</u> <u>seria</u> <u>seria</u> <u>seria</u> <u>seria</u> <u>seria</u> <u>seria</u> <u>seria</u> <u>seria</u> <u>seria</u> <u>seria</u> <u>seria</u> <u>seria</u> <u>seria</u> <u>seria</u> <u>seria</u> <u>seria</u> <u>seria</u> <u>seria</u> <u>seria</u> <u>seria</u> <u>seria</u> <u>seria</u> <u>seria</u> <u>seria</u> <u>seria</u> <u>seria</u> <u>seria</u> <u>seria</u> <u>seria</u> <u>seria</u> <u>seria</u> <u>seria</u> <u>seria</u> <u>seria</u> <u>seria</u> <u>seria</u> <u>seria</u> <u>seria</u> <u>seria</u> <u>seria</u> <u>seria</u> <u>seria</u> <u>seria</u>	his application tion is entitled is under Rules to)	<u>Octob</u> claims th to small e	- ne benefit of entity status.
9. 9.	Of Pilit Washi Con which i (a) □ [1 [1] Petitior x) □ it	angton, D end the : trinuation is a contin Provision has beer Hence: (No.) ∑ filed ∑ attac to exten s being c	specification division nuation of 0 the specific nal Applicat n recently de) Verified St in above pr ched. ad the life of oncurrently	on by insertin on of Appl 18/425, 160, file cation by inse ion No. 60/ etermined that tatement(s) es ior application the above pri filed in that p	ication No. <u>08/96(</u> <u>series code</u> îr ed April 11, 1995, no erting before the firs , filed, t this new continuing stablishing "small en	0,079,	tion is entitled uned. his application tion is entitled us under Rules to) date hereof	<u>Octob</u> claims th to small e	- ne benefit of entity status.

,

.

27

PAT-108 12/98

RACK-1002 Page 8 of 351

Page 3 of 4 INFORMATION DISCLOSURE STATEMENT: Attached is Form PTO-1449 listing all of the documents 12. \boxtimes cited by Applicant and the PTO in the parent application(s) relied upon under 35 USC 120 and referenced in item 9 above. Per Rule 98(d) copies of those documents are not required now. Please consider those documents and advise that they have been considered in this new application as by returning a copy of the enclosed Form PTO-1449 with the Examiner's initials in the left column per MPEP 609. Attached is a Rule 103(a) Petition to Suspend Action. 13. PRELIMINARY AMENDMENT to be entered before fee calculation: (Do not make amendments here 14. \boxtimes except for correction of improper multiple dependencies or cancellation of whole claims or multiple dependencies for purpose of reducing the filing fee per MPEP §§ 506 and 607; do not cancel all claims). Please cancel claims 1-45 and 50-53 without prejudice. The remaining claims correspond to non-elected Groups III & IV from the Examiner's Restriction Requirement of June 4, 1996. FILING FEE THE FOLLOWING FILING FEE IS BASED ON ->->-><u>CLAIMS AS FILED AND CHANGED BY PRELIMINARY AMENDMENT IN ITEM 14</u><-<-<---NOTE: If box 1A2 is X'd, do not pay fees, but leave lines 15-22 and 27-32 blank. Large/Small Fee Entity Code \$310/\$155 106/26 \$760/\$380 +380 101/201 17. Total Effective Claims 5 minus 20 = 0 x \$18/\$9 +0 103/203 18. Independent Claims 3 minus 3 = 0 x \$78/\$39 +0 102/202 19. If any proper multiple dependent claim (ignore improper) is present, \$260/\$130 +0 104/204 20. Subtotal = \$380 +0 122 +40 581 22. TOTAL FILING FEE ATTACHED = \$420 (carry forward to Item 31) 23. ATTACHED: 24.

Preliminary Amendment attached (to be entered after assigning Appln. No.)

Z.

The following PRELIMINARY AMENDMENT is to be entered after assigning Appln. No.: 25.

PAT-108 12/98

RACK-1002 Page 9 of 351

Page 4 of 4

ADDITIONAL FEE CALCULATION FOR PRELIMINARY AMENDMENT <u>PER BOXES 24/25</u>

	Claims remaining after amendment	Highest number previously paid for	Present Extra	Additional Fee
			Large/Small Entity	E File Code
27.	Total Effective Claims	minus ** _20 =	_0x \$18/\$9 =	= \$_0 (103/203)
28.	Independent Claims _*	minus *** <u>3</u> =	_0x \$78/\$39 =	= + 0 (102/202)
29.	If amendment enters proper mu <u>first time</u> , add (per application)			+ 0 (104/204)
30.			ADDITIONAL FEI	= \$ <u>0</u>
31.		<u>plus</u> FEE	from item 22 on page 3	+ 420
32.			TOTAL FEE ATTACHED	\$ 420
33.	*If the entry in this space is less than the entry in	n the next space, the "Present Extra"	' result is "0"	

34. **If the "Highest number previously paid for" (see item 17 above) is less than 20, write "20" in this space

35. If the "Highest number previously paid for" (see item 18 above) is less than 3, write "3" in this space

<u>CHARGE STATEMENT</u>: Upon the filing of a Declaration pursuant to Rule 60(b) or 60(d), the Commissioner is hereby authorized to charge any fee specifically authorized hereafter, or any missing or insufficient fee(s) filed, or asserted to be filed, or which should have been filed herewith or concerning any paper filed hereafter, and which may be required under Rules 16-18 (missing or insufficient fee only) now or hereafter relative to this application and the resulting Official document under Rule 20, or credit any overpayment, to our Account/Order Nos. shown in the heading hereof for which purpose a <u>duplicate</u> copy of this sheet is attached.

This CHARGE STATEMENT does not authorize charge of the issue fee until/unless an issue fee transmittal form is filed.

		Madison & Sutro LLP al Property Group		
1100 New York Avenue, N.W.	By Atty:	Dale S. Lazat/	Reg. No.	28872
Ninth Floor, East Tower Washington, D.C. 20005-3918	Sig:		Fax:	(202) 822-0944
Tel: (202) 861-3000 DSL/BS:kim			Tel:	(202) 861-3527
Atty./Sec.				

NOTE No. 1: File this Request in <u>duplicate</u> with 2 postcard receipts (PAT-103) & attachments **NOTE No. 2:** Is extension in parent necessary for copendency? **DOUBLE CHECK Item 11 above.**

26.

PAT-108 12/98

RACK-1002 Page 10 of 351

APPLICATION UNDER UNITED STATES PATENT LAWS

Invention: David A. Farber and Ronald D. Lachman Inventor(s): IDENTIFYING DATA IN A DATA PROCESSING SYSTEM

> Cushman Darby & Cushman, L.L.P. 1100 New York Avenue, N.W. Ninth Floor, East Tower Washington, D.C. 20005-3918 Attorneys Telephone: (202) 861-3000

<u>This is a</u>:

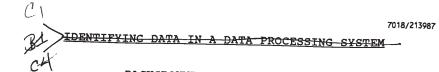
F.F.

- [] Provisional Application
- [X] Regular Utility Application
- [] Continuing Application
- [] PCT National Phase Application
- [] Design Application
- [] Reissue Application
- [] Plant Application

SPECIFICATION

CDC-100 3/95

RACK-1002 Page 11 of 351



BACKGROUND OF THE INVENTION

1.

2.

Field of the invention

This invention relates to data processing systems and, more particularly, to data processing systems wherein data items are identified by substantially unique identifiers which depend on all of the data in the data items and only on the data in the data items.

10

15

20

25

30

5

Background of the Invention

Data processing (DP) systems, computers, networks of computers, or the like, typically offer users and programs various ways to identify the data in the systems.

Users typically identify data in the data processing system by giving the data some form of name. For example, a typical operating system (OS) on a computer provides a file system in which data items are named by alphanumeric identifiers. Programs typically identify data in the data processing system using a location or address. For example, a program may identify a record in a file or database by using a record number which serves to locate that record.

In all but the most primitive operating systems, users and programs are able to create and use collections of named data items, these collections themselves being named by identifiers. These named collections can then, themselves, be made part of other named collections. For example, an OS may provide mechanisms to group files (data items) into directories (collections). These directories can then, themselves be made part of other directories. A data item may thus be identified relative to these nested directories using a

1

0

RACK-1002 Page 12 of 351 sequence of names, or a so-called pathname, which defines a path through the directories to a particular data item (file or directory).

As another example, a database management system may group data records (data items) into tables and then group these tables into database files (collections). The complete address of any data record can then be specified using the database file name, the table name, and the record number of that data record. Other examples of identifying data items

include: identifying files in a network file system, identifying objects in an object-oriented database, identifying images in an image database, and identifying articles in a text database.

In general, the terms "data" and "data item" as used herein refer to sequences of bits. Thus a data item may be the contents of a file, a portion of a file, a page in memory, an object in an object-oriented program, a digital message, a digital scanned image, a part of a video or audio signal, or any other entity which can be represented by a sequence of bits. The term "data processing" herein refers to the processing of data items, and is sometimes dependent on the type of data item being processed. For example, a data processor for a digital image may differ from a data processor for an audio signal.

In all of the prior data processing systems the names or identifiers provided to identify data items (the data items being files, directories, records in the database, objects in object-oriented programming, locations in memory or on a physical device, or the like) are always defined relative to a specific context. For instance, the file identified by a particular file name can only be determined when the directory containing the file (the context) is known. The file identified by a pathname can be determined only when the file system (context) is known. Similarly, the addresses in a

2

]-é cerchu.

5

10

15

20

25

30

process address space, the keys in a database table, or domain names on a global computer network such as the Internet are meaningful only because they are specified relative to a context.

In prior art systems for identifying data items there is no direct relationship between the data names and the data item. The same data name in two different contexts may refer to different data items, and two different data names in the same context may refer to the same data item.

In addition, because there is no correlation between a data name and the data it refers to, there is no a priori way to confirm that a given data item is in fact the one named by a data name. For instance, in a DP system, if one processor requests that another processor deliver a data item with a given data name, the requesting processor cannot, in general, verify that the data delivered is the correct data (given only the name). Therefore it may require further processing, typically on the part of the requestor, to verify that the data item it has obtained is, in fact, the item it requested.

A common operation in a DP system is adding a new data item to the system. When a new data item is added to the system, a name can be assigned to it only by updating the context in which names are defined. Thus such systems require a centralized mechanism for the management of names. Such a mechanism is required even in a multi-processing system when data items are created and identified at separate processors in distinct locations, and in which there is no other need for communication when data items are added.

In many data processing systems or environments, data items are transferred between different locations in the system. These locations may be processors in the data processing system, storage devices, memory, or the like. For example, one processor may obtain a data item from another processor or from an

3

10

15

20

j j

đ

مىيە. 1. ئىلىم 5

.

25

30

35

RACK-1002 Page 14 of 351 external storage device, such as a floppy disk, and may incorporate that data item into its system (using the name provided with that data item).

However, when a processor (or some location) obtains a data item from another location in the DP system, it is possible that this obtained data item is already present in the system (either at the location of the processor or at some other location accessible by the processor) and therefore a duplicate of the data item is created. This situation is common in a network data processing environment where proprietary software products are installed from floppy disks onto several processors sharing a common file server. In these systems, it is often the case that the same product will be installed on several systems, so that several copies of each file will reside on the common file server. In some data processing systems in which

several processors are connected in a network, one system is designated as a cache server to maintain master copies of data items, and other systems are designated as cache clients to copy local copies of the master data items into a local cache on an as-needed basis. Before using a cached item, a cache client must either reload the cached item, be informed of changes to the cached item, or confirm that the master item corresponding to the cached item has not changed. In other words, a cache client must synchronize its data items with those on the cache server. This synchronization may involve reloading data items onto the cache client. The need to keep the cache synchronized or reload it adds significant overhead to existing caching mechanisms.

In view of the above and other problems with prior art systems, it is therefore desirable to have a mechanism which allows each processor in a multiprocessor system to determine a common and substantially unique identifier for a data item, using only the data in the data item and not relying on any sort of context.

4

5

10

15

20

25

35

It is further desirable to have a mechanism for reducing multiple copies of data items in a data processing system and to have a mechanism which enables the identification of identical data items so as to reduce multiple copies. It is further desirable to determine whether two instances of a data item are in fact the same data item, and to perform various other systems' functions and applications on data items without relying on any context information or properties of the data item.

It is also desirable to provide such a mechanism in such a way as to make it transparent to users of the data processing system, and it is desirable that a single mechanism be used to address each of the problems described above.

SUMMARY OF THE INVENTION

This invention provides, in a data processing system, a method and apparatus for identifying a data item in the system, where the identity of the data item depends on all of the data in the data item and only on the data in the data item. Thus the identity of a data item is independent of its name, origin, location, address, or other information not derivable directly from the data, and depends only on the data itself.

This invention further provides an apparatus and a method for determining whether a particular data item is present in the system or at a location in the system, by examining only the data identities of a plurality of data items.

30

35

5

10

15

20

25

. I., I.,

ŵ

đ

e,

Using the method or apparatus of the present invention, the efficiency and integrity of a data processing system can be improved. The present invention improves the design and operation of a data storage system, file system, relational database, object-oriented database, or the like that stores a plurality of data items, by making possible or improving the design and

5

operation of at least some or all of the following features:

5

10

15

20

30

35

1.1

.

octore de la c

the system stores at most one copy of any data item at a given location, even when multiple data names in the system refer to the same contents;

the system avoids copying data from source to destination locations when the destination locations already have the data;

the system provides transparent access to any data item by reference only to its identity and independent of its present location, whether it be local, remote, or offline;

the system caches data items from a server, so that only the most recently accessed data items need be retained;

when the system is being used to cache data items, problems of maintaining cache consistency are avoided;

the system maintains a desired level of redundancy of data items in a network of servers, to protect against failure by ensuring that multiple copies of the data items are present at different locations in the system;

the system automatically archives data items as 25 they are created or modified;

the system provides the size, age, and location of groups of data items in order to decide whether they can be safely removed from a local file system;

the system can efficiently record and preserve any collection of data items;

the system_can efficiently make a copy of any collection of data items, to support a version control mechanism for groups of the data items;

the system can publish data items, allowing other, possibly anonymous, systems in a network to gain access to the data items and to rely on the availability of the data items; the system can maintain a local inventory of all the data items located on a given removable medium, such as a diskette or CD-ROM, the inventory is independent of other properties of the data items such as their name, location, and date of creation;

the system allows closely related sets of data items, such as matching or corresponding directories on disconnected computers, to be periodically resynchronized with one another;

10

15

20

25

5

the system can verify that data retrieved from another location is the désired or requested data, using only the data identifier used to retrieve the data;

the system can prove possession of specific data items by content without disclosing the content of the data items, for purposes of later legal verification and to provide anonymity;

the system tracks possession of specific data items according to content by owner, independent of the name, date, or other properties of the data item, and tracks the uses of specific data items and files by content for accounting purposes.

Other objects, features, and characteristics of the present invention as well as the methods of operation and functions of the related elements of structure, and the combination of parts and economies of manufacture, will become more apparent upon consideration of the following description and the appended claims with reference to the accompanying drawings, all of which form a part of this specification.

BRIEF DESCRIPTION OF THE DRAWINGS

D1 FIGURE 1 depicts a typical data processing system in which a preferred embodiment of the present invention operates;

FIGURE 2 depicts a hierarchy of data items 35 stored at any location in such a data processing system;

7

FIGURES 3-9 depict data structures used to implement an embodiment of the present invention; and FIGURES 10(a)-28 are flow charts depicting operation of various aspects of the present invention.

 DETAILED DESCRIPTION OF THE PRESENTLY PREFERRED

 \
 EXEMPLARY EMBODIMENTS

D2 An embodiment of the present invention is now described with reference to a typical data processing system 100, which, with reference to FIGURE 1, includes one or more processors (or computers) 102 and various storage devices 104 connected in some way, for example by a bus 106.

Each processor 102 includes a CPU 108, a memory 110 and one or more local storage devices 112. The CPU 108, memory 110, and local storage device 112 may be internally connected, for example by a bus 114. Each processor 102 may also include other devices (not shown), such as a keyboard, a display, a printer, and the like.

In a data processing system 100, wherein more than one processor 102 is used, that is, in a multiprocessor system, the processors may be in one of various relationships. For example, two processors 102 may be in a client/server, client/client, or a server/server relationship. These inter-processor relationships may be dynamic, changing depending on particular situations and functions. Thus, a particular processor 102 may change its relationship to other processors as needed, essentially setting up a peer-topeer relationship with other processors. In a peer-topeer relationship, sometimes a particular processor 102 acts as a client processor, whereas at other times the same processor acts as a server processor. In other words, there is no hierarchy imposed on or required of processors 102.

In a multiprocessor system, the processors 102 may be homogeneous or heterogeneous. Further, in a

5

10

15

20

25

30

35

8

 c_{i}

multiprocessor data processing system 100, some or all of the processors 102 may be disconnected from the network of processors for periods of time. Such disconnection may be part of the normal operation of the system 100 or it may be because a particular processor 102 is in need of repair.

Within a data processing system 100, the data may be organized to form a hierarchy of data storage elements, wherein lower level data storage elements are combined to form higher level elements. This hierarchy can consist of, for example, processors, file systems, regions, directories, data files, segments, and the like. For example, with reference to FIGURE 2, the data items on a particular processor 102 may be organized or structured as a file system 116 which comprises regions 117, each of which comprises directories 118, each of which can contain other directories 118 or files 120. Each file 120 being made up of one or more data segments 122.

In a typical data processing system, some or all of these elements can be named by users given certain implementation specific naming conventions, the name (or pathname) of an element being relative to a context. In the context of a data processing system 100, a pathname is fully specified by a processor name, a filesystem name, a sequence of zero or more directory names identifying nested directories, and a final file name. (Usually the lowest level elements, in this case segments 122, cannot be named by users.)

30

5

10

15

20

25

35

In other words, a file system 116 is a collection of directories 118. A directory 118 is a collection of named files 120 -- both data files 120 and other directory files 118. A file 120 is a named data item which is either a data file (which may be simple or compound) or a directory file 118. A simple file 120 consists of a single data segment 122. A compound file 120 consists of a sequence of data segments 122. A data

RACK-1002 Page 20 of 351

9

j (

segment 122 is a fixed sequence of bytes. An important property of any data segment is its size, the number of bytes in the sequence.

A single processor 102 may access one or more file systems 116, and a single storage device 104 may contain one or more file systems 116, or portions of a file system 116. For instance, a file system 116 may span several storage devices 104.

5

10

15

20

25

30

35

5

In order to implement controls in a file system, file system 116 may be divided into distinct regions, where each region is a unit of management and control. A region consists of a given directory 118 and is identified by the pathname (user defined) of the directory.

In the following, the term "location", with respect to a data processing system 100, refers to any of a particular processor 102 in the system, a memory of a particular processor, a storage device, a removable storage medium (such as a floppy disk or compact disk), or any other physical location in the system. The term "local" with respect to a particular processor 102 refers to the memory and storage devices of that particular processor.

In the following, the terms "True Name", "data identity" and "data identifier" refer to the substantially unique data identifier for a particular data item. The term "True File" refers to the actual file, segment, or data item identified by a True Name.

A file system for a data processing system 100 is now described which is intended to work with an existing operating system by augmenting some of the operating system's file management system codes. The embodiment provided relies on the standard file management primitives for actually storing to and retrieving data items from disk, but uses the mechanisms of the present invention to reference and access those data items.

The processes and mechanisms (services) provided in this embodiment are grouped into the following categories: primitive mechanisms, operating system mechanisms, remote mechanisms, background 5 mechanisms, and extended mechanisms. Primitive mechanisms provide fundamental capabilities used to support other mechanisms. The following primitive mechanisms are described: 1. Calculate True Name; 10 2. Assimilate Data Item; 3. New True File; 4. Get True Name from Path; 5. Link path to True Name; Realize True File from Location; 6. 15 7. Locate Remote File; 8. Make True File Local; 9. Create Scratch File; 10. Freeze Directory; 11. Expand Frozen Directory; 20 12. Delete True File; 13. Process Audit File Entry; 14. Begin Grooming; 15. Select For Removal; and 16. End Grooming. 25 Operating system mechanisms provide typical familiar file system mechanisms, while maintaining the data structures required to offer the mechanisms of the present invention. Operating system mechanisms are designed to augment existing operating systems, and in this way to make the present invention compatible with, 30 and generally transparent to, existing applications. The following operating system mechanisms are described: 1. Open File; 2. Close File; 35 3. Read File; Write File; 4. 5. Delete File or Directory;

11

12-

Copy File or Directory; 6, Move File or Directory; 7. Get File Status; and 8. Get Files in Directory. 9. Remote mechanisms are used by the operating 5 system in responding to requests from other processors. These mechanisms enable the capabilities of the present invention in a peer-to-peer network mode of operation. The following remote mechanisms are described: Locate True File; 10 1. Reserve True File; 2. Request True File; 3. Retire True File; 4. Cancel Reservation; 5. Acquire True File; 6. 15 Lock Cache; 7. Update Cache; and 8. Check Expiration Date. 9. Background mechanisms are intended to run occasionally and at a low priority. These provide m 20 automated management capabilities with respect to the present invention. The following background mechanisms are described: Mirror True File; 1. Groom Region; 2. 25 Check for Expired Links; and Ē 3. Verify Region; and 4. Groom Source List. 5. Extended mechanisms run within application programs over the operating system. These mechanisms 30 provide solutions to specific problems and applications. The following extended mechanisms are described: Inventory Existing Directory; 1. Inventory Removable, Read-only Files; 2. Synchronize directories; 3. 35 Publish Region; 4. Retire Directory; 5.

12

- 6. Realize Directory at location;
- 7. Verify True File;
- 8. Track for accounting purposes; and
- 9. Track for licensing purposes.

The file system herein described maintains sufficient information to provide a variety of mechanisms not ordinarily offered by an operating system, some of which are listed and described here. Various processing performed by this embodiment of the present invention will now be described in greater detail.

In some embodiments, some files 120 in a data processing system 100 do not have True Names because they have been recently received or created or modified, and thus their True Names have not yet been computed. A file that does not yet have a True Name is called a scratch file. The process of assigning a True Name to a file is referred to as assimilation, and is described later. Note that a scratch file may have a user provided name. Some of the processing performed by the present invention can take place in a background mode or on a delayed or as-needed basis. This background processing is used to determine information that is not immediately required by the system or which may never be required. As an example, in some cases a scratch file is being changed at a rate greater than the rate at which it is useful to determine its True Name. In these cases, determining the True Name of the file can be postponed or

Data Structures

performed in the background.

30

5

10

15

20

25

C

Ű

Ð

The following data structures, stored in memory 110 of one of more processors 102 are used to implement the mechanisms described herein. The data structures can be local to each processor 102 of the system 100, or they can reside on only some of the processors 102.

13

RACK-1002 Page 24 of 351 The data structures described are assumed to reside on individual peer processors 102 in the data processing system 100. However, they can also be shared by placing them on a remote, shared file server (for

10

15

20

5

by placing them on a remote, shared file server (for instance, in a local area network of machines). In order to accommodate sharing data structures, it is necessary that the processors accessing the shared database use the appropriate locking techniques to ensure that changes to the shared database do not interfere with one another but are appropriately serialized. These locking techniques are well understood by ordinarily skilled programmers of distributed applications.

It is sometimes desirable to allow some regions to be local to a particular processor 102 and other regions to be shared among processors 102. (Recall that a region is a unit of file system management and control consisting of a given directory identified by the pathname of the directory.) In the case of local and shared regions, there would be both local and shared versions of each data structure. Simple changes to the processes described below must be made to ensure that appropriate data structures are selected for a given operation.

The local directory extensions (LDE) table 124 is a data structure which provides information about files 120 and directories 118 in the data processing system 100. The local directory extensions table 124 is indexed by a pathname or contextual name (that is, a user provided name) of a file and includes the True Name for most files. The information in local directory extension table 124 is in addition to that provided by the native file system of the operating system.

The True File registry (TFR) 126 is a data store for listing actual data items which have True Names, both files 120 and segments 122. When such data items occur in the True File registry 126 they are known as True Files. True Files are identified in True File

đ

25

30

35

registry 126 by their True Names or identities. The table True File registry 126 also stores location, dependency, and migration information about True Files.

The region table (RT) 128 defines areas in the network storage which are to be managed separately. Region table 128 defines the rules for access to and migration of files 120 among various regions with the local file system 116 and remote peer file systems.

The source table (ST) 130 is a list of the sources of True Files other than the current True File registry 126. The source table 130 includes removable volumes and remote processors.

The audit file (AF) 132 is a list of records indicating changes to be made in local or remote files, these changes to be processed in background.

The accounting log (AL) 134 is a log of file transactions used to create accounting information in a manner which preserves the identity of files being tracked independent of their name or location.

The license table (LT) 136 is a table identifying files, which may only be used by licensed users, in a manner independent of their name or location, and the users licensed to use them.

Detailed Descriptions of the Data Structures

The following table summarizes the fields of an local directory extensions table entry, as illustrated by record 138 in FIGURE 3.

Field	Description
Region ID	identifies the region in which this file is contained.
Pathname	the user provided name or contextual name of the file or directory, relative to the region in which it occurs.
True Name	the computed True Name or identity of the file or directory. This True Name is not always up to date, and it is set to a special value when a file is modified and is later recomputed in the background.

15

20

15

5

The Party

ų.

ñ

ويجا ألمنو

25

30

Field	Description
Туре	indicates whether the file is a data file or a directory.
Scratch File ID	the physical location of the file in the file system, when no True Name has been calculated for the file. As noted above, such a file is called a scratch file.
Time of last access	the last access time to this file. If thi file is a directory, this is the last access time to any file in the directory.
Time of last modi- fication	the time of last change of this file. If this file is a directory, this is the last modification time of any file in the directory.
Safe flag	indicates that this file (and, if this fill is a directory, all of its subordinate files) have been backed up on some other system, and it is therefore safe to remove them.
Lock flag	indicates whether a file is locked, that is, it is being modified by the local pro- cessor or a remote processor. Only one processor may modify a file at a time.
Size	the full size of this directory (including all subordinate files), if all files in it were fully expanded and duplicated. For a file that is not a directory this is the size of the actual True File.
Owner	the identity of the user who owns this file, for accounting and license tracking purposes.

Each record of the True File registry 126 has 15 the fields shown in the True File registry record 140 in FIGURE 4. The True File registry 126 consists of the database described in the table below as well as the actual True Files identified by the True File IDs below. _____

Field	Description	
True Name	computed True Name or identity of the file.	

5

10

(0^{11)^{1, 20}}

RACK-1002 Page 27 of 351

Field	Description
Compressed File ID	compressed version of the True File may be stored instead of, or in addition to, an uncompressed version. This field provides the identity of the actual representation of the compressed version of the file.
Grooming delete count	tentative count of how many references have been selected for deletion during a grooming operation.
Time of last access	most recent date and time the content of this file was accessed.
Expiration	date and time after which this file may be deleted by this server.
Dependent processors	processor IDs of other processors which contain references to this True File.
Source IDs	source ID(s) of zero or more sources from which this file or data item may be retrieved.
True File ID	identity or disk location of the actual physical representation of the file or file segment. It is sufficient to use a filename in the registration directory of the underlying operating system. The True File ID is absent if the actual file is not currently present at the current location.
Use count	number of other records on this processor which identify this True File.

OGESIEC _OHOIGO

5

10

A region table 128, specified by a directory pathname, records storage policies which allow files in 15 the file system to be stored, accessed and migrated in different ways. Storage policies are programmed in a configurable way using a set of rules described below. Each region table record 142 of region table 128 includes the fields described in the following table 20 (with reference to FIGURE 5):

17

15

RACK-1002 Page 28 of 351

Field	Description
Region ID	internally used identifier for this region.
Region file system	file system on the local processor of which this region is a part.
Region pathname	a pathname relative to the region fil system which defines the location of this region. The region consists of all files and directories subordinate to this pathname, except those in a region subordinate to this region.
Mirror processor(s)	zero or more identifiers of processor which are to keep mirror or archival copies of all files in the current region. Multiple mirror processors can be defined to form a mirror group
Mirror duplication count	number of copies of each file in this region that should be retained in a mirror group.
Region status	specifies whether this region is local to a single processor 102, shared by several processors 102 (if, for instance, it resides on a shared file server), or managed by a remote processor.
Policy	the migration policy to apply to this region. A single region might participate in several policies. The policies are as follows (parameters in brackets are specified as part of the policy): region is a cached version from [processor ID]; region is a member of a mirror set defined by [processor ID]. region is to be archived on [processor ID]. region is to be backed up locally, by placing new copies in [region ID]. region is read only and may not be changed. region is published and expires on [date]. Files in this region should be compressed.

10

K UN

5

A source table 130 identifies a source location for True Files. The source table 130 is also used to

identify client processors making reservations on the current processor. Each source record 144 of the source table 130 includes the fields summarized in the following table, with reference to FIGURE 6:

5	Field	Description
2	source ID	internal identifier used to identify a particular source.
	source type	type of source location: Removable Storage Volume Local Region Cache Server Mirror Group Server Cooperative Server Publishing Server Client
10	source rights	includes information about the rights of this processor, such as whether it can ask the local processor to store data items for it.
	source availabil- ity	measurement of the bandwidth, cost, and reliability of the connection to this source of True Files. The avail- ability is used to select from among several possible sources.
15	source location	information on how the local processor is to access the source. This may be, for example, the name of a removable storage volume, or the processor ID and region path of a region on a remote processor.

The audit file 132 is a table of events ordered by timestamp, each record 146 in audit file 132 including the fields summarized in the following table (with reference to FIGURE 7):

20

лŪ

1.0

Field	Description
Original Name	path of the file in question.
Operation	whether the file was created, read, written, copied or deleted.
Туре	specifies whether the source is a file or a directory.

19

Field	Description
Processor ID	ID of the remote processor generating this event (if not local).
Timestamp	time and date file was closed (required only for accessed/modified files).
Pathname	Name of the file (required only for rename).
True Name	computed True Name of the file. This is used by remote systems to mirror changes to the directory and is filled in during background processing.

Each record 148 of the accounting log 134 records an event which may later be used to provide information for billing mechanisms. Each accounting log entry record 148 includes at least the information, summarized in the following table, with reference to FIGURE 8:

5

10

210

15

20

< C211

Ũ

Field	Description
date of entry	date and time of this log entry.
type of entry	Entry types include create file, delete file, and transmit file.
True Name	True Name of data item in question.
owner	identity of the user responsible for this action.

Each record 150 of the license table 136 records a relationship between a licensable data item and the user licensed to have access to it. Each license table record 150 includes the information summarized in the following table, with reference to FIGURE 9:

Field	Description
True Name	True Name of a data item subject to license validation.

20

21

RACK-1002 Page 31 of 351

Field	Description	
licensee	identity of a user authorized to have access to this object.	

Various other data structures are employed on some or all of the processors 102 in the data processing system 100. Each processor 102 has a global freeze lock (GFL) 152 (FIGURE 1), which is used to prevent synchronization errors when a directory is frozen or copied. Any processor 102 may include a special archive directory (SAD) 154 into which directories may be copied for the purposes of archival. Any processor 102 may include a special media directory (SMD) 156, into which the directories of removable volumes are stored to form a media inventory. Each processor has a grooming lock 158, which is set during a grooming operation. During this period the grooming delete count of True File registry entries 140 is active, and no True Files should be deleted until grooming is complete. While grooming is in effect, grooming information includes a table of pathnames selected for deletion, and keeps track of the amount of space that would be freed if all of the files were deleted.

Primitive Mechanisms

5

10

15

20

25

30

The first of the mechanisms provided by the present invention, primitive mechanisms, are now described. The mechanisms described here depend on underlying data management mechanisms to create, copy, read, and delete data items in the True File registry 126, as identified by a True File ID. This support may be provided by an underlying operating system or disk storage manager.

The following primitive mechanisms are described:

1. Calculate True Name;

2. Assimilate Data Item;

21

22

RACK-1002 Page 32 of 351

3. New True File; 4. Get True Name from Path; Link Path to True Name; 5. 6. Realize True File from Location; 5 7. Locate Remote File; 8. Make True File Local; 9. Create Scratch File; 10. Freeze Directory; 11. Expand Frozen Directory; 10 12. Delete True File; 13. Process Audit File Entry; 14. Begin Grooming; 15. Select For Removal; and 16. End Grooming. 15 1. Calculate True Name A True Name is computed using a function, MD, which reduces a data block B of arbitrary length to a relatively small, fixed size identifier, the True Name of the data block, such that the True Name of the data block 20 is virtually guaranteed to represent the data block B and only data block B. The function MD must have the following properties: The domain of the function MD is the set 1. 25 of all data items. The range of the function MD is the set of True Names. The function MD must take a data item of 2. arbitrary length and reduce it to an integer value in the range 0 to N-1, where 30 N is the cardinality of the set of True Names. That is, for an arbitrary length data block B, $0 \leq MD(B) < N$. The results of MD(B) must be evenly and 3. randomly distributed over the range of N, 35 in such a way that simple or regular

22

27

RACK-1002 Page 33 of 351 changes to B are virtually guaranteed to produce a different value of MD(B).

- It must be computationally difficult to find a different value B' such that MD(B)=MD(B').
- The function MD(B) must be efficiently computed.

A family of functions with the above properties are the so-called message digest functions, which are used in digital security systems as techniques for authentification of data. These functions (or algorithms) include MD4, MD5, and SHA.

In the presently preferred embodiments, either MD5 or SHA is employed as the basis for the computation of True Names. Whichever of these two message digest functions is employed, that same function must be employed on a system-wide basis.

It is impossible to define a function having a unique output for each possible input when the number of elements in the range of the function is smaller than the number of elements in its domain. However, a crucial observation is that the actual data items that will be encountered in the operation of any system embodying this invention form a very sparse subset of all the possible inputs.

A colliding set of data items is defined as a set wherein, for one or more pairs x and y in the set, MD(x) = MD(y). Since a function conforming to the requirements for MD must evenly and randomly distribute its outputs, it is possible, by making the range of the function large enough, to make the probability arbitrarily small that actual inputs encountered in the operation of an embodiment of this invention will form a colliding set.

35

5

10

15

20

25

30

To roughly quantify the probability of a collision, assume that there are no more than 2^{30} storage devices in the world, and that each storage device has an

1

23

 $\overset{n}{\prec}$

RACK-1002 Page 34 of 351 average of at most 2^{20} different data items. Then there are at most 2^{50} data items in the world. If the outputs of MD range between 0 and 2^{128} , it can be demonstrated that the probability of a collision is approximately 1 in 2^{29} . Details on the derivation of these probability values are found, for example, in P. Flajolet and A.M. Odlyzko, "Random Mapping Statistics," Lecture Notes in Computer Science 434: Advances in Cryptology -- Eurocrypt '89 Proceedings, Springer-Verlag, pp. 329-354.

Note that for some less preferred embodiments of the present invention, lower probabilities of uniqueness may be acceptable, depending on the types of applications and mechanisms used. In some embodiments it may also be useful to have more than one level of True Names, with some of the True Names having different degrees of uniqueness. If such a scheme is implemented, it is necessary to ensure that less unique True Names are not propagated in the system.

While the invention is described herein using only the True Name of a data item as the identifier for the data item, other preferred embodiments use tagged, typed, categorized or classified data items and use a combination of both the True Name and the tag, type, category or class of the data item as an identifier. Examples of such categorizations are files, directories, and segments; executable files and data files, and the like. Examples of classes are classes of objects in an object-oriented system. In such a system, a lower degree of True Name uniqueness is acceptable over the entire universe of data items, as long as sufficient uniqueness is provided per category of data items. This is because the tags provide an additional level of uniqueness.

A mechanism for calculating a True Name given a data item is now described, with reference to FIGURES 10(a) and 10(b).

A simple data item is a data item whose size is less than a particular given size (which must be defined

25

30

5

10

15

20

25

35

in each particular implementation of the invention). To determine the True Name of a simple data item, with reference to FIGURE 10(a), first compute the MD function (described above) on the given simple data item (Step S212). Then append to the resulting 128 bits, the byte length modulo 32 of the data item (Step S214). The resulting 160-bit value is the True Name of the simple data item.

A compound data item is one whose size is greater than the particular given size of a simple data item. To determine the True Name of an arbitrary (simple or compound) data item, with reference to FIGURE 10(b), first determine if the data item is a simple or a compound data item (Step S216). If the data item is a simple data item, then compute its True Name in step S218 (using steps S212 and S214 described above), otherwise partition the data item into segments (Step S220) and assimilate each segment (Step S222) (the primitive mechanism, Assimilate a Data Item, is described below), computing the True Name of the segment. Then create an indirect block consisting of the computed segment True Names (Step S224). An indirect block is a data item which consists of the sequence of True Names of the segments. Then, in step S226, assimilate the indirect block and compute its True Name. Finally, replace the final thirty-two (32) bits of the resulting True Name (that is, the length of the indirect block) by the length modulo 32 of the compound data item (Step S228). The result is the True Name of the compound data item.

Note that the compound data item may be so large that the indirect block of segment True Names is itself a compound data item. In this case the mechanism is invoked recursively until only simple data items are being processed.

Both the use of segments and the attachment of a length to the True Name are not strictly required in a system using the present invention, but are currently

25

24

RACK-1002 Page 36 of 351

81

tion durit three light

1

5

10

15

20

25

30

considered desirable features in the preferred embodiment.

<u>Assimilate Data Item</u>

5

10

15

20

25

30

2.

reference to FIGURE 11. The purpose of this mechanism is to add a given data item to the True File registry 126. If the data item already exists in the True File registry 126, this will be discovered and used during this process, and the duplicate will be eliminated.

Thereby the system stores at most one copy of any data item or file by content, even when multiple names refer to the same content.

A mechanism for assimilating a data item (scratch file or segment) into a file system, given the

scratch file ID of the data item, is now described with

First, determine the True Name of the data item corresponding to the given scratch File ID using the Calculate True Name primitive mechanism (Step S230). Next, look for an entry for the True Name in the True File registry 126 (Step S232) and determine whether a True Name entry, record 140, exists in the True File registry 126. If the entry record includes a corresponding True File ID or compressed File ID (Step S237), delete the file with the scratch File ID (Step S238). Otherwise store the given True File ID in the entry record (step S239).

If it is determined (in step S232) that no True Name entry exists in the True File registry 126, then, in Step S236, create a new entry in the True File registry 126 for this True Name. Set the True Name of the entry to the calculated True Name, set the use count for the new entry to one, store the given True File ID in the entry and set the other fields of the entry as appropriate.

26

27

RACK-1002 Page 37 of 351

Because this procedure may take some time to compute, it is intended to run in background after a file has ceased to change. In the meantime, the file is considered an unassimilated scratch file.

3. <u>New True File</u>

The New True File process is invoked when processing the audit file 132, some time after a True File has been assimilated (using the Assimilate Data Item primitive mechanism). Given a local directory extensions table entry record 138 in the local directory extensions table 124, the New True File process can provide the following steps (with reference to FIGURE 12), depending on how the local processor is configured:

First, in step S238, examine the local directory extensions table entry record 138 to determine whether the file is locked by a cache server. If the file is locked, then add the ID of the cache server to the dependent processor list of the True File registry table 126, and then send a message to the cache server to update the cache of the current processor using the Update Cache remote mechanism (Step 242).

If desired, compress the True File (Step S246), and, if desired, mirror the True File using the Mirror True File background mechanism (Step S248).

25

30

35

4.

5

10

15

20

<u>Get True Name from Path</u>

The True Name of a file can be used to identify a file by contents, to confirm that a file matches its original contents, or to compare two files. The mechanism to get a True Name given the pathname of a file is now described with reference to FIGURE 13.

First, search the local directory extensions table 124 for the entry record 138 with the given pathname (Step S250). If the pathname is not found, this process fails and no True Name corresponding to the given pathname exists. Next, determine whether the local

27

. 25

RACK-1002 Page 38 of 351 directory extensions table entry record 138 includes a True Name (Step S252), and if so, the mechanism's task is complete. Otherwise, determine whether the local directory extensions table entry record 138 identifies a directory (Step S254), and if so, freeze the directory (Step S256) (the primitive mechanism Freeze Directory is described below).

Otherwise, in step S258, assimilate the file (using the Assimilate Data Item primitive mechanism) defined by the File ID field to generate its True Name and store its True Name in the local directory extensions entry record. Then return the True Name identified by the local directory extensions table 124.

5. Link Path to True Name

The mechanism to link a path to a True Name provides a way of creating a new directory entry record identifying an existing, assimilated file. This basic process may be used to copy, move, and rename files without a need to copy their contents. The mechanism to link a path to a True Name is now described with reference to FIGURE 14.

First, if desired, confirm that the True Name exists locally by searching for it in the True Name registry or local directory extensions table 135 (Step S260). Most uses of this mechanism will require this form of validation. Next, search for the path in the local directory extensions table 135 (Step S262). Confirm that the directory containing the file named in the path already exists (Step S264). If the named file itself exists, delete the File using the Delete True File operating system mechanism (see below) (Step S268).

Then, create an entry record in the local directory extensions with the specified path (Step S270) and update the entry record and other data structures as follows: fill in the True Name field of the entry with the specified True Name; increment the use count for the

10

25

5

10

15

20

30

28

29

RACK-1002 Page 39 of 351

True File registry entry record 140 of the corresponding True Name; note whether the entry is a directory by reading the True File to see if it contains a tag (magic number) indicating that it represents a frozen directory 5 (see also the description of the Freeze Directory primitive mechanism regarding the tag); and compute and set the other fields of the local directory extensions appropriately. For instance, search the region table 128 to identify the region of the path, and set the time of last access and time of last modification to the current time.

6.

10

15

20

25

1.1.1.1

Û Ĩ

28

Realize True File from Location

This mechanism is used to try to make a local copy of a True File, given its True Name and the name of a source location (processor or media) that may contain the True File. This mechanism is now described with reference to FIGURE 15.

First, in step S272, determine whether the location specified is a processor. If it is determined that the location specified is a processor, then send a Request True File message (using the Request True File remote mechanism) to the remote processor and wait for a response (Step S274). If a negative response is received or no response is received after a timeout period, this mechanism fails. If a positive response is received, enter the True File returned in the True File registry 126 (Step S276). (If the file received was compressed, enter the True File ID in the compressed File ID field.)

30

35

If, on the other hand, it is determined in step S272 that the location specified is not a processor, then, if necessary, request the user or operator to mount the indicated volume (Step S278). Then (Step S280) find the indicated file on the given volume and assimilate the file using the Assimilate Data Item primitive mechanism. If the volume does not contain a True File registry 126, search the media inventory to find the path of the file

29

Zi

on the volume. If no such file can be found, this mechanism fails.

Locate Remote File

follows (with reference to FIGURE 16).

At this point, whether or not the location is determined (in step S272) to be a processor, if desired, verify the True File (in step S282).

7.

5

10

Q

The second second

160.04010

Ő

35

D3 This mechanism allows a processor to locate a file or data item from a remote source of True Files, when a specific source is unknown or unavailable. A client processor system may ask one of several or many sources whether it can supply a data object with a given True Name. The steps to perform this mechanism are as

The client processor 102 uses the source table 15 145 to select one or more source processors (Step S284). If no source processor can be found, the mechanism fails. Next, the client processor 102 broadcasts to the selected sources a request to locate the file with the given True Name using the Locate True File remote mechanism (Step S286). The request to locate may be augmented by asking 20 to propagate this request to distant servers. The client processor then waits for one or more servers to respond positively (Step S288). After all servers respond negatively, or after a timeout period with no positive 25 response, the mechanism repeats selection (Step S284) to attempt to identify alternative sources. If any selected source processor responds, its processor ID is the result of this mechanism. Store the processor ID in the source field of the True File registry entry record 140 of the 30 given True Name (Step. S290).

If the source location of the True Name is a different processor or medium than the destination (Step S290a), perform the following steps:

(i) Look up the True File registry entry record 140 for the corresponding True Name, and add the

source location ID to the list of sources for the True Name (Step S290b); and

(ii) If the source is a publishing system, determine the expiration date on the publishing system
for the True Name and add that to the list of sources. If the source is not a publishing system, send a message to reserve the True File on the source processor (Step S290c).

Source selection in step S284 may be based on optimizations involving general availability of the source, access time, bandwidth, and transmission cost, and ignoring previously selected processors which did not respond in step S288.

8. <u>Make True File Local</u>

10

15

20

25

30

35

(D)

١Ū

D⁴ This mechanism is used when a True Name is known and a locally accessible copy of the corresponding file or data item is required. This mechanism makes it possible to actually read the data in a True File. The mechanism takes a True Name and returns when there is a local, accessible copy of the True File in the True File registry 126. This mechanism is described here with reference to the flow chart of FIGURE 17.

First, look in the True File registry 126 for a True File entry record 140 for the corresponding True Name (Step S292). If no such entry is found this mechanism fails. If there is already a True File ID for the entry (Step S294), this mechanism's task is complete. If there is a compressed file ID for the entry (Step S296), decompress the file corresponding to the file ID (Step S298) and store the decompressed file ID in the entry (Step S300). This mechanism is then complete.

If there is no True File ID for the entry (Step S294) and there is no compressed file ID for the entry (Step S296), then continue searching for the requested file. At this time it may be necessary to notify the user that the system is searching for the requested file.

31

32

RACK-1002 Page 42 of 351 If there are one or more source IDs, then select an order in which to attempt to realize the source ID (Step S304). The order may be based on optimizations involving general availability of the source, access time, bandwidth, and transmission cost. For each source in the order chosen, realize the True File from the source location (using the Realize True File from Location primitive mechanism), until the True File is realized (Step S306). If it is realized, continue with step S294. If no known source can realize the True File, use the Locate Remote File primitive mechanism to attempt to find the True File (Step S308). If this succeeds, realize the True File from the identified source location and continue with step S296.

9. <u>Create Scratch File</u>

5

10

15

20

25

30

35

Ő

1.1

Ц Ц

đ

D5 A scratch copy of a file is required when a file is being created or is about to be modified. The scratch copy is stored in the file system of the underlying operating system. The scratch copy is eventually assimilated when the audit file record entry 146 is processed by the Process Audit File Entry primitive mechanism. This Create Scratch File mechanism requires a local directory extensions table entry record 138. When it succeeds, the local directory extensions table entry record 138 contains the scratch file ID of a scratch file that is not contained in the True File registry 126 and that may be modified. This mechanism is now described with reference to FIGURE 18.

First determine whether the scratch file should be a copy of the existing True File (Step S310). If so, continue with step S312. Otherwise, determine whether the local directory extensions table entry record 138 identifies an existing True File (Step S316), and if so, delete the True File using the Delete True File primitive mechanism (Step S318). Then create a new, empty scratch file and store its scratch file ID in the local directory

32

30

RACK-1002 Page 43 of 351 extensions table entry record 138 (Step S320). This mechanism is then complete.

If the local directory extensions table entry record 138 identifies a scratch file ID (Step S312), then the entry already has a scratch file, so this mechanism succeeds.

If the local directory extensions table entry record 138 identifies a True File (S316), and there is no True File ID for the True File (S312), then make the True File local using the Make True File Local primitive mechanism (Step S322). If there is still no True File ID, this mechanism fails.

There is now a local True File for this file. If the use count in the corresponding True File registry entry record 140 is one (Step S326), save the True File ID in the scratch file ID of the local directory extensions table entry record 138, and remove the True File registry entry record 140 (Step S328). (This step makes the True File into a scratch file.) This mechanism's task is complete.

Otherwise, if the use count in the corresponding True File registry entry record 140 is not one (in step S326), copy the file with the given True File ID to a new scratch file, using the Read File OS mechanism and store its file ID in the local directory extensions table entry record 138 (Step S330), and reduce the use count for the True File by one. If there is insufficient space to make a copy, this mechanism fails.

10. Freeze Directory

5

10

15

20

25

30

35

CD D C C C C C C

This mechanism freezes a directory in order to calculate its True Name. Since the True Name of a directory is a function of the files within the directory, they must not change during the computation of the True Name of the directory. This mechanism requires the pathname of a directory to freeze. This mechanism is described with reference to FIGURE 19.

In step S332, add one to the global freeze lock. Then search the local directory extensions table 124 to find each subordinate data file and directory of the given directory, and freeze each subordinate directory found using the Freeze Directory primitive mechanism (Step S334). Assimilate each unassimilated data file in the directory using the Assimilate Data Item primitive mechanism (Step S336). Then create a data item which begins with a tag or marker (a "magic number") being a unique data item indicating that this data item is a frozen directory (Step S337). Then list the file name and True Name for each file in the current directory (Step S338). Record any additional information required, such as the type, time of last access and modification, and size (Step S340). Next, in step S342, using the Assimilate Data Item primitive mechanism, assimilate the data item created in step S338. The resulting True Name is the True Name of the frozen directory. Finally, subtract one from the global freeze lock (Step S344).

Expand Frozen Directory

5

10

15

20

25

30

11.

Ű

TU TU

L.

.

This mechanism expands a frozen directory in a given location. It requires a given pathname into which to expand the directory, and the True Name of the directory and is described with reference to FIGURE 20.

First, in step S346, make the True File with the given True Name local using the Make True File Local primitive mechanism. Then read each directory entry in the local file created in step S346 (Step S348). For each such directory entry, do the following:

Create a full pathname using the given pathname and the file name of the entry (Step S350); and

link the created path to the True Name (Step S352) using the Link Path to True Name primitive mechanism.

12. <u>Delete True File</u>

This mechanism deletes a reference to a True Name. The underlying True File is not removed from the True File registry 126 unless there are no additional references to the file. With reference to FIGURE 21, this mechanism is performed as follows:

If the global freeze lock is on, wait until the global freeze lock is turned off (Step S354). This prevents deleting a True File while a directory which might refer to it is being frozen. Next, find the True File registry entry record 140 given the True Name (Step S356). If the reference count field of the True File registry 126 is greater than zero, subtract one from the reference count field (Step S358). If it is determined (in step S360) that the reference count field of the True File registry entry record 140 is zero, and if there are no dependent systems listed in the True File registry entry record 140, then perform the following steps: (i) If the True File is a simple data item,

20

25

5

10

15

0

đ.

then delete the True File, otherwise,
 (ii) (the True File is a compound data item)
for each True Name in the data item, recursively delete
the True File corresponding to the True Name (Step S362).

(iii) Remove the file indicated by the True
 File ID and compressed file ID from the True File
 registry 126, and remove the True File registry entry
 record 140 (Step S364).

13. <u>Process Audit File Entry</u>

30

35

This mechanism performs tasks which are required to maintain information in the local directory extensions table 124 and True File registry 126, but which can be delayed while the processor is busy doing more time-critical tasks. Entries 142 in the audit file 132 should be processed at a background priority as long as there are entries to be processed. With reference to

35

-74

FIGURE 22, the steps for processing an entry are as follows:

Determine the operation in the entry 142 currently being processed (Step S365). If the operation indicates that a file was created or written (Step S366), then assimilate the file using the Assimilate Data Item primitive mechanism (Step S368), use the New True File primitive mechanism to do additional desired processing (such as cache update, compression, and mirroring) (Step S369), and record the newly computed True Name for the file in the audit file record entry (Step S370).

Otherwise, if the entry being processed indicates that a compound data item or directory was copied (or deleted) (Step S376), then for each component True Name in the compound data item or directory, add (or subtract) one to the use count of the True File registry entry record 140 corresponding to the component True Name (Step S378).

In all cases, for each parent directory of the given file, update the size, time of last access, and time of last modification, according to the operation in the audit record (Step S379).

Note that the audit record is not removed after processing, but is retained for some reasonable period so that it may be used by the Synchronize Directory extended mechanism to allow a disconnected remote processor to update its representation of the local system.

14. Begin Grooming

30

35

5

10

15

20

25

, É

This mechanism makes it possible to select a set of files for removal and determine the overall amount of space to be recovered. With reference to FIGURE 23, first verify that the global grooming lock is currently unlocked (Step S382). Then set the global grooming lock, set the total amount of space freed during grooming to zero and empty the list of files selected for deletion

36

(Step S384). For each True File in the True File registry 126, set the delete count to zero (Step S386).

15.

5

20

25

30

0

Q

<u>ة</u> القر

Select For Removal

This grooming mechanism tentatively selects a pathname to allow its corresponding True File to be removed. With reference to FIGURE 24, first find the local directory extensions table entry record 138 corresponding to the given pathname (Step S388). Then find the True File registry entry record 140 corresponding to the True File name in the local 10 directory extensions table entry record 138 (Step S390). Add one to the grooming delete count in the True File registry entry record 140 and add the pathname to a list of files selected for deletion (Step S392). If the grooming delete count of the True File registry entry 15 record 140 is equal to the use count of the True File registry entry record 140, and if the there are no entries in the dependency list of the True File registry entry record 140, then add the size of the file indicated by the True File ID and or compressed file ID to the

total amount of space freed during grooming (Step S394).

End Grooming 16.

This grooming mechanism ends the grooming phase and removes all files selected for removal. With reference to FIGURE 25, for each file in the list of files selected for deletion, delete the file (Step S396) and then unlock the global grooming lock (Step S398).

Operating System Mechanisms

1.

The next of the mechanisms provided by the present invention, operating system mechanisms, are now described.

The following operating system' mechanisms are described:

Open File;

37

35

RACK-1002 Page 48 of 351

2.	Close File;
3.	Read File;
4.	Write File;
5.	Delete File or Directory;
6.	Copy File or Directory;
7.	Move File or Directory;

8. Get File Status; and

9. Get Files in Directory.

1. <u>Open File</u>

D1 A mechanism to open a file is described with reference to FIGURE 26. This mechanism is given as input a pathname and the type of access required for the file (for example, read, write, read/write, create, etc.) and produces either the File ID of the file to be opened or an indication that no file should be opened. The local directory extensions table record 138 and region table record 142 associated with the opened file are associated with the open file for later use in other processing functions which refer to the file, such as read, write, and close.

First, determine whether or not the named file exists locally by examining the local directory extensions table 124 to determine whether there is an entry corresponding to the given pathname (Step S400). If it is determined that the file name does not exist locally, then, using the access type, determine whether or not the file is being created by this opening process (Step S402). If the file is not being created, prohibit the open (Step S404). If the file is being created, create a zero-length-scratch file using an entry in local directory extensions table 124 and produce the scratch file ID of this scratch file as the result (Step S406).

If, on the other hand, it is determined in step S400 that the file name does exist locally, then determine the region in which the file is located by searching the region table 128 to find the record 142

5

10

15

20

25

30

1

Ű

ñ

35

38

34

RACK-1002 Page 49 of 351 with the longest region path which is a prefix of the file pathname (Step S408). This record identifies the region of the specified file.

Next, determine using the access type, whether the file is being opened for writing or whether it is being opened only for reading (Step S410). If the file is being opened for reading only, then, if the file is a scratch file (Step S419), return the scratch File ID of the file (Step S424). Otherwise get the True Name from the local directory extensions table 124 and make a local version of the True File associated with the True Name using the Make True File Local primitive mechanism, and then return the True File ID associated with the True Name (Step S420).

If the file is not being opened for reading only (Step S410), then, if it is determined by inspecting the region table entry record 142 that the file is in a read-only directory (Step S416), then prohibit the opening (Step S422).

If it is determined by inspecting the region table 128 that the file is in a cached region (Step S423), then send a Lock Cache message to the corresponding cache server, and wait for a return message (Step S418). If the return message says the file is already locked, prohibit the opening.

If the access type indicates that the file being modified is being rewritten completely (Step S419), so that the original data will not be required, then Delete the File using the Delete File OS mechanism (Step S421) and perform step S406. Otherwise, make a scratch copy of the file (Step S417) and produce the scratch file ID of the scratch file as the result (Step S424).

2. <u>Close File</u>

35

5

10

15

20

25

30

10

This mechanism takes as input the local directory extensions table entry record 138 of an open file and the data maintained for the open file. To close

39

40

RACK-1002 Page 50 of 351 a file, add an entry to the audit file indicating the time and operation (create, read or write). The audit file processing (using the Process Audit File Entry primitive mechanism) will take care of assimilating the file and thereby updating the other records.

3. <u>Read File</u>

5

10

15

20

30

35

To read a file, a program must provide the offset and length of the data to be read, and the location of a buffer into which to copy the data read.

The file to be read from is identified by an open file descriptor which includes a File ID as computed by the Open File operating system mechanism defined above. The File ID may identify either a scratch file or a True File (or True File segment). If the File ID identifies a True File, it may be either a simple or a compound True File. Reading a file is accomplished by the following steps:

In the case where the File ID identifies a scratch file or a simple True File, use the read capabilities of the underlying operating system.

In the case where the File ID identifies a compound file, break the read operation into one or more read operations on component segments as follows:

A. Identify the segment(s) to be read by dividing the specified file offset and length each by the fixed size of a segment (a system dependent parameter), to determine the segment number and number of segments that must be read.

B. For each segment number computed above, do the following:

i. Read the compound True File index block to determine the True Name of the segment to be read.

ii. Use the Realize True File from Location primitive mechanism to make the True File

40

2

ortor carrade

Û

41

RACK-1002 Page 51 of 351 segment available locally. (If that mechanism fails, the Read File mechanism fails).

iii. Determine the File ID of the True File specified by the True Name corresponding to this segment.

iv. Use the Read File mechanism (recursively) to read from this segment into the corresponding location in the specified buffer.

4. Write File

10

15

20

25

5.

Ő

ų,

Ð

5

File writing uses the file ID and data management capabilities of the underlying operating system. File access (Make File Local described above) can be deferred until the first read or write.

<u>Delete File or Directory</u>

 \mathfrak{S} The process of deleting a file, for a given pathname, is described here with reference to FIGURE 27.

First, determine the local directory extensions table entry record 138 and region table entry record 142 for the file (Step S422). If the file has no local directory extensions table entry record 138 or is locked or is in a read-only region, prohibit the deletion.

Identify the corresponding True File given the True Name of the file being deleted using the True File registry 126 (Step S424). If the file has no True Name, (Step S426) then delete the scratch copy of the file based on its scratch file ID in the local directory extensions table 124 (Step S427), and continue with step S428.

30

If the file has a True Name and the True File's use count is one (Step S429), then delete the True File (Step S430), and continue with step S428.

If the file has a True Name and the True File's use count is greater than one, reduce its use count by one (Step S431). Then proceed with step S428.

In Step S428, delete the local directory extensions table entry record, and add an entry to the audit file 132 indicating the time and the operation performed (delete).

5

10

15

20

25

35

6.

<u>Copy File or Directory</u>

A mechanism is provided to copy a file or directory given a source and destination processor and pathname. The Copy File mechanism does not actually copy the data in the file, only the True Name of the file. This mechanism is performed as follows:

(A) Given the source path, get the True Name from the path. If this step fails, the mechanism fails.
(B) Given the True Name and the destination path, link the destination path to the True Name.

(C) If the source and destination processors have different True File registries, find (or, if necessary, create) an entry for the True Name in the True File registry table 126 of the destination processor.

Enter into the source ID field of this new entry the source processor identity.

(D) Add an entry to the audit file 132 indicating the time and operation performed (copy).

This mechanism addresses capability of the system to avoid copying data from a source location to a destination location when the destination already has the data. In addition, because of the ability to freeze a directory, this mechanism also addresses capability of the system immediately to make a copy of any collection of files, thereby to support an efficient version control mechanisms for groups of files.

7. <u>Move File or Directory</u>

A mechanism is described which moves (or renames) a file from a source path to a destination path. The move operation, like the copy operation, requires no actual transfer of data, and is performed as follows:

30

42

(A) Copy the file from the source path to the destination path.

(B) If the source path is different from the destination path, delete the source path.

8. <u>Get File Status</u>

5

10

15

20

25

30

9.

This mechanism takes a file pathname and provides information about the pathname. First the local directory extensions table entry record 138 corresponding to the pathname given is found. If no such entry exists, then this mechanism fails, otherwise, gather information about the file and its corresponding True File from the local directory extensions table 124. The information can include any information shown in the data structures, including the size, type, owner, True Name, sources, time of last access, time of last modification, state (local or not, assimilated or not, compressed or not), use count, expiration date, and reservations.

<u>Get Files in Directory</u>

This mechanism enumerates the files in a directory. It is used (implicitly) whenever it is necessary to determine whether a file exists (is present) in a directory. For instance, it is implicitly used in the Open File, Delete File, Copy File or Directory, and Move File operating system mechanisms, because the files operated on are referred to by pathnames containing directory names. The mechanism works as follows:

The local directory extensions table 124 is searched for an entry 138 with the given directory pathname. If no such entry is found, or if the entry found is not a directory, then this mechanism fails.

If there is a corresponding True File field in the local directory extensions table record, then it is assumed that the True File represents a frozen directory. The Expand Frozen Directory primitive mechanism is used

43

44

RACK-1002 Page 54 of 351 to expand the existing True File into directory entries in the local directory extensions table.

Finally, the local directory extensions table 124 is again searched, this time to find each directory subordinate to the given directory. The names found are provided as the result.

Remote Mechanisms

The remote mechanisms provided by the present invention are now described. Recall that remote mechanisms are used by the operating system in responding to requests from other processors. These mechanisms enable the capabilities of the present invention in a peer-to-peer network mode of operation.

In a presently preferred embodiment, processors communicate with each other using a remote procedure call (RPC) style interface, running over one of any number of communication protocols such as IPX/SPX or TCP/IP. Each peer processor which provides access to its True File registry 126 or file regions, or which depends on another peer processor, provides a number of mechanisms which can be used by its peers.

The following remote mechanisms are described:

1. Locate True File;

25

5

10

15

20

1

ų

D H H H H

3. Request True File;

Reserve True File;

- 4. Retire True File;
- 5. Cancel Reservation;
- 6. Acquire True File;
- 7. Lock Gache;

30

1.

- 8. Update Cache; and
- 9. Check Expiration Date.
- Locate True File

2.

This mechanism allows a remote processor to determine whether the local processor contains a copy of

RACK-1002 Page 55 of 351

a specific True File. The mechanism begins with a True Name and a flag indicating whether to forward requests for this file to other servers. This mechanism is now described with reference to FIGURE 28.

5

10

15

20

25

30

35

A line find

First determine if the True File is available locally or if there is some indication of where the True File is located (for example, in the Source IDs field). Look up the requested True Name in the True File registry 126 (Step S432).

If a True File registry entry record 140 is not found for this True Name (Step S434), and the flag indicates that the request is not to be forwarded (Step S436), respond negatively (Step S438). That is, respond to the effect that the True File is not available.

One the other hand, if a True File registry entry record 140 is not found (Step S434), and the flag indicates that the request for this True File is to be forwarded (Step S436), then forward a request for this True File to some other processors in the system (Step S442). If the source table for the current processor identifies one or more publishing servers which should have a copy of this True File, then forward the request to each of those publishing servers (Step S436).

If a True File registry entry record 140 is found for the required True File (Step S434), and if the entry includes a True File ID or Compressed File ID (Step S440), respond positively (Step S444). If the entry includes a True File ID then this provides the identity or disk location of the actual physical representation of the file or file segment required. If the entry include a Compressed File ID, then a compressed version of the True File may be stored instead of, or in addition to, an uncompressed version. This field provides the identity of the actual representation of the compressed version of the file.

If the True File registry entry record 140 is found (Step S434) but does not include a True File ID

46

RACK-1002 Page 56 of 351 (the File ID is absent if the actual file is not currently present at the current location) (Step S440), and if the True File registry entry record 140 includes one or more source processors, and if the request can be forwarded, then forward the request for this True File to one or more of the source processors (Step S444).

2. <u>Reserve True File</u>

This mechanism allows a remote processor to indicate that it depends on the local processor for access to a specific True File. It takes a True Name as input. This mechanism is described here.

(A) Find the True File registry entry record140 associated with the given True File. If no entryexists, reply negatively.

(B) If the True File registry entry record 140 does not include a True File ID or compressed File ID, and if the True File registry entry record 140 includes no source IDs for removable storage volumes, then this processor does not have access to a copy of the given file. Reply negatively.

(C) Add the ID of the sending processor to the list of dependent processors for the True File registry entry record 140. Reply positively, with an indication of whether the reserved True File is on line or off line.

25

30

3.

5

10

15

20

And And And

Ð

<u>Request True File</u>

This mechanism allows a remote processor to request a copy of a True File from the local processor. It requires a True Name and responds positively by sending a True File back to the requesting processor. The mechanism operates as follows:

(A) Find the True File registry entry record 140 associated with the given True Name. If there is no such True File registry entry record 140, reply negatively. (B) Make the True File local using the Make True File Local primitive mechanism. If this mechanism fails, the Request True File mechanism also fails.

(C) Send the local True File in either it is uncompressed or compressed form to the requesting remote processor. Note that if the True File is a compound file, the components are not sent.

(D) If the remote file is listed in the dependent process list of the True File registry entry record 140, remove it.

4. <u>Retire True File</u>

This mechanism allows a remote processor to indicate that it no longer plans to maintain a copy of a given True File. An alternate source of the True File can be specified, if, for instance, the True File is being moved from one server to another. It begins with a True Name, a requesting processor ID, and an optional alternate source. This mechanism operates as follows:

(A) Find a True Name entry in the True File registry 126. If there is no entry for this True Name, this mechanism's task is complete.

(B) Find the requesting processor on the source list and, if it is there, remove it.

(C) If an alternate source is provided, add it to the source list for the True File registry entry record 140.

(D) If the source list of the True File registry entry record 140 has no items in it, use the Locate Remote File primitive mechanism to search for another copy of the file. If it fails, raise a serious error.

5.

5

10

15

20

25

30

35

ā

<u>Cancel Reservation</u>

This mechanism allows a remote processor to indicate that it no longer requires access to a True File stored on the local processor. It begins with a True

47

:4\$

Name and a requesting processor ID and proceeds as follows:

(A) Find the True Name entry in the True File registry 126. If there is no entry for this True Name, this mechanism's task is complete.

(B) Remove the identity of the requesting processor from the list of dependent processors, if it appears.

(C) If the list of dependent processors becomes zero and the use count is also zero, delete the True File.

6. <u>Acquire True File</u>

This mechanism allows a remote processor to insist that a local processor make a copy of a specified True File. It is used, for example, when a cache client wants to write through a new version of a file. The Acquire True File mechanism begins with a data item and an optional True Name for the data item and proceeds as follows:

(A) Confirm that the requesting processor has the right to require the local processor to acquire data items. If not, send a negative reply.

(B) Make a local copy of the data item transmitted by the remote processor.

25

5

10

15

20

5

(C) Assimilate the data item into the True File registry of the local processor.

(D) If a True Name was provided with the file, the True Name calculation can be avoided, or the mechanism can verify that the file received matches the True Name sent.

(E) Add an entry in the dependent processor list of the true file registry record indicating that the requesting processor depends on this copy of the given True File.

35

30

(F) Send a positive reply.

48

7. Lock Cache

This mechanism allows a remote cache client to lock a local file so that local users or other cache clients cannot change it while the remote processor is using it. The mechanism begins with a pathname and proceeds as follows:

(A) Find the local directory extensions table entry record 138 of the specified pathname. If no such entry exists, reply negatively.

(B) If an local directory extensions table entry record 138 exists and is already locked, reply negatively that the file is already locked.

(C) If an local directory extensions table entry record 138 exists and is not locked, lock the entry. Reply positively.

8. <u>Update Cache</u>

This mechanism allows a remote cache client to unlock a local file and update it with new contents. It begins with a pathname and a True Name. The file corresponding to the True Name must be accessible from the remote processor. This mechanism operates as follows:

Find the local directory extensions table entry record 138 corresponding to the given pathname. Reply negatively if no such entry exists or if the entry is not locked.

Link the given pathname to the given True Name using the Link Path to True Name primitive mechanism. Unlock the local directory extensions table

entry record 138 and return positively.

9.

5

10

15

20

25

30

The line was in the

the true and

đ

ē.]

Check Expiration Date

Return current or new expiration date and possible alternative source to caller.

49

42

RACK-1002 Page 60 of 351

Background Processes and Mechanisms

The background processes and mechanisms provided by the present invention are now described. Recall that background mechanisms are intended to run occasionally and at a low priority to provide automated management capabilities with respect to the present invention.

The following background mechanisms are described:

10

5

- 1. Mirror True File;
- 2. Groom Region;
- 3. Check for Expired Links;
- 4. Verify Region; and
- 5. Groom Source List.

15

20

25

30

35

1.

ar iedi

<u>Mirror True File</u>

This mechanism is used to ensure that files are available in alternate locations in mirror groups or archived on archival servers. The mechanism depends on application-specific migration/archival criteria (size, time since last access, number of copies required, number of existing alternative sources) which determine under what conditions a file should be moved. The Mirror True File mechanism operates as follows, using the True File specified, perform the following steps:

(A) Count the number of available locations of the True File by inspecting the source list of the True File registry entry record 140 for the True File. This step determines how many copies of the True File are available in the system.

(B) If the True File meets the specified migration criteria, select a mirror group server to which a copy of the file should be sent. Use the Acquire True File remote mechanism to copy the True File to the selected mirror group server. Add the identity of the selected system to the source list for the True File.

50

L,

<u>Groom Region</u>

2.

5

10

15

20

25

C

11

Ċ.

This mechanism is used to automatically free up space in a processor by deleting data items that may be available elsewhere. The mechanism depends on application-specific grooming criteria (for instance, a file may be removed if there is an alternate online source for it, it has not been accessed in a given number of days, and it is larger than a given size). This mechanism operates as follows:

Repeat the following steps (i) to (iii) with more aggressive grooming criteria until sufficient space is freed or until all grooming criteria have been exercised. Use grooming information to determine how much space has been freed. Recall that, while grooming is in effect, grooming information includes a tablé of pathnames selected for deletion, and keeps track of the amount of space that would be freed if all of the files were deleted.

(i) Begin Grooming (using the primitive mechanism).

(ii) For each pathname in the specified region, for the True File corresponding to the pathname, if the True File is present, has at least one alternative source, and meets application specific grooming criteria for the region, select the file for removal (using the primitive mechanism).

(iii) End Grooming (using the primitive mechanism).

If the region is used as a cache, no other

30

35

processors are dependent on True Files to which it refers, and all such True Files are mirrored elsewhere. In this case, True Files can be removed with impunity. For a cache region, the grooming criteria would ordinarily eliminate the least recently accessed True Files first. This is best done by sorting the True Files in the region by the most recent access time before performing step (ii) above. The application specific

51

criteria would thus be to select for removal every True File encountered (beginning with the least recently used) until the required amount of free space is reached.

5

10

15

20

25

SSTURD DEFERSE

3.

<u>Check for Expired Links</u>

This mechanism is used to determine whether dependencies on published files should be refreshed. The following steps describe the operation of this mechanism:

For each pathname in the specified region, for each True File corresponding to the pathname, perform the following step:

If the True File registry entry record 140 corresponding to the True File contains at least one source which is a publishing server, and if the expiration date on the dependency is past or close, then perform the following steps:

(A) Determine whether the True File registry entry record contains other sources which have not expired.

(B) Check the True Name expiration of the server. If the expiration date has been extended, or an alternate source is suggested, add the source to the True File registry entry record 140.

(C) If no acceptable alternate source was found in steps (A) or (B) above, make a local copy of the True File.

(D) Remove the expired source.

4.

<u>Verify Region</u>

30

This mechanism can be used to ensure that the data items in the True File registry 126 have not been damaged accidentally or maliciously. The operation of this mechanism is described by the following steps:

(A) Search the local directory extensions table 124 for each pathname in the specified region and then perform the following steps:

5**2**

53

RACK-1002 Page 63 of 351

(i) Get the True File name corresponding to the pathname;

(ii) If the True File registry entry 140 for the True File does not have a True File ID or compressed file ID, ignore it.

(iii) Use the Verify True File mechanism (see extended mechanisms below) to confirm that the True File specified is correct.

5.

10

15

20

25

<u>cruto</u> corrato

5

Groom Source List

The source list in a True File entry should be groomed sometimes to ensure there are not too many mirror or archive copies. When a file is deleted or when a region definition or its mirror criteria are changed, it may be necessary to inspect the affected True Files to determine whether there are too many mirror copies. This can be done with the following steps:

For each affected True File,

(A) Search the local directory extensions

table to find each region that refers to the True File.

(B) Create a set of "required sources",

initially empty.

(C) For each region found,

determine the mirroring criteria for (a) that region,

(b) determine which sources for the True File satisfy the mirroring criteria, and

(c) add these sources to the set of required sources.

30

35

(D) For each source in the True File registry entry, if the source_identifies a remote processor (as opposed to removable media), and if the source is not a publisher, and if the source is not in the set of required sources, then eliminate the source, and use the Cancel Reservation remote mechanism to eliminate the given processor from the list of dependent processors

53

recorded at the remote processor identified by the source.

Extended Mechanisms

The extended mechanisms provided by the present invention are now described. Recall that extended mechanisms run within application programs over the operating system to provide solutions to specific problems and applications.

The following extended mechanisms are described:

10 descri

1.

5

- 1. Inventory Existing Directory;
- 2. Inventory Removable, Read-only Files;
- 3. Synchronize Directories;
- 4. Publish Region;
- 5. Retire Directory;
- 6. Realize Directory at Location;
- 7. Verify True File;
- 8. Track for Accounting Purposes; and
- 9. Track for Licensing Purposes.
- 20

15

Inventory Existing Directory

This mechanism determines the True Names of files in an existing on-line directory in the underlying operating system. One purpose of this mechanism is to install True Name mechanisms in an existing file system.

25

30

An effect of such an installation is to eliminate immediately all duplicate files from the file system being traversed. If several file systems are inventoried in a single True File registry, duplicates across the volumes are also eliminated.

(A) Traverse the underlying file system in the operating system. For each file encountered, excluding directories, perform the following:

(i) Assimilate the file encountered (using the Assimilate File primitive mechanism). This

54

process computes its True Name and moves its data into the True File registry 126.

(ii) Create a pathname consisting of the path to the volume directory and the relative path of the file on the media. Link this path to the computed True Name using the Link Path to True Name primitive mechanism.

Inventory Removable, Read-only Files

10

15

20

25

35

Coraz

ģ

ñ

46.2

2.

5

A system with access to removable, read-only media volumes (such as WORM disks and CD-ROMS) can create a usable inventory of the files on these disks without having to make online copies. These objects can then be used for archival purposes, directory overlays, or other needs. An operator must request that an inventory be created for such a volume.

This mechanism allows for maintaining inventories of the contents of files and data items on removable media, such as diskettes and CD-ROMs, independent of other properties of the files such as name, location, and date of creation.

The mechanism creates an online inventory of the files on one or more removable volumes, such as a floppy disk or CD-ROM, when the data on the volume is represented as a directory. The inventory service uses a True Name to identify each file, providing a way to locate the data independent of its name, date of creation, or location.

30

The inventory can be used for archival of data (making it possible to avoid archiving data when that data is already on a separate volume), for grooming (making it possible to delete infrequently accessed files if they can be retrieved from removable volumes), for version control (making it possible to generate a new version of a CD-ROM without having to copy the old version), and for other purposes.

55

El:

The inventory is made by creating a volume directory in the media inventory in which each file named identifies the data item on the volume being inventoried. Data items are not copied from the removable volume during the inventory process.

5

10

15

20

25

30

35

An operator must request that an inventory be created for a specific volume. Once created, the volume directory can be frozen or copied like any other directory. Data items from either the physical volume or

the volume directory can be accessed using the Open File operating system mechanism which will cause them to be read from the physical volume using the Realize True File from Location primitive mechanism.

To create an inventory the following steps are taken:

(A) A volume directory in the media inventory is created to correspond to the volume being inventoried. Its contextual name identifies the specific volume.

(B) A source table entry 144 for the volume is created in the source table 130. This entry 144 identifies the physical source volume and the volume directory created in step (A).

(C) The filesystem on the volume is traversed. For each file encountered, excluding directories, the following steps are taken:

(i) The True Name of the file is computed. An entry is created in the True Name registry 124, including the True Name of the file using the primitive mechanism. The source field of the True Name registry entry 140 identifies the source table entry 144.

(ii) A pathname is created consisting of the path to the volume directory and the relative path of the file on the media. This path is linked to the computed True Name using Link Path to True Name primitive mechanism.

(D) After all files have been inventoried, the volume directory is frozen. The volume directory serves

5**6**

as a table of contents for the volume. It can be copied using the Copy File or Directory primitive mechanism to create an "overlay" directory which can then be modified, making it possible to edit a virtual copy of a read-only medium.

3. <u>Synchronize Directories</u>

5

10

15

20

25

30

35

Derester of the set

Ð

Given two versions of a directory derived from the same starting point, this mechanism creates a new, synchronized version which includes the changes from

each. Where a file is changed in both versions, this mechanism provides a user exit for handling the discrepancy. By using True Names, comparisons are instantaneous, and no copies of files are necessary. This mechanism lets a local processor

synchronize a directory to account for changes made at a remote processor. Its purpose is to bring a local copy of a directory up to date after a period of no communication between the local and remote processor. Such a period might occur if the local processor were a mobile processor detached from its server, or if two distant processors were run independently and updated nightly.

An advantage of the described synchronization process is that it does not depend on synchronizing the clocks of the local and remote processors. However, it does require that the local processor track its position in the remote processor's audit file.

This mechanism does not resolve changes made simultaneously to the same file at several sites. If that occurs, an external resolution mechanism such as, for example, operator intervention, is required.

The mechanism takes as input a start time, a local directory pathname, a remote processor name, and a remote directory pathname name, and it operates by the following steps:

57

(A) Request a copy of the audit file 132 from the remote processor using the Request True File remote mechanism.

(B) For each entry 146 in the audit file 132 after the start time, if the entry indicates a change to a file in the remote directory, perform the following steps:

(i) Compute the pathname of the corresponding file in the local directory. Determine the True Name of the corresponding file.

(ii) If the True Name of the local file is the same as the old True Name in the audit file, or if there is no local file and the audit entry indicates a new file is being created, link the new True Name in the audit file to the local pathname using the Link Path to True Name primitive mechanism.

(iii) Otherwise, note that there is a problem with the synchronization by sending a message to the operator or to a problem resolution program, indicating the local pathname, remote pathname, remote processor, and time of change.

(C) After synchronization is complete, record the time of the final change. This time is to be used as the new start time the next time this directory is synchronized with the same remote processor.

4. <u>Publish Region</u>

The publish region mechanism allows a processor to offer the files in a region to any client processors for a limited period of time.

30

5

10

15

20

25

0.63

× 85

The purpose of the service is to eliminate any need for client processors to make reservations with the publishing processor. This in turn makes it possible for the publishing processor to service a much larger number of clients.

35

When a region is published, an expiration date is defined for all files in the region, and is propagated

58

into the publishing system's True File registry entry record 140 for each file.

When a remote file is copied, for instance using the Copy File operating system mechanism, the expiration date is copied into the source field of the client's True File registry entry record 140. When the source is a publishing system, no dependency need be created.

The client processor must occasionally and in background, check for expired links, to make sure it still has access to these files. This is described in the background mechanism Check for Expired Links.

5. <u>Retire Directory</u>

This mechanism makes it possible to eliminate safely the True Files in a directory, or at least dependencies on them, after ensuring that any client processors depending on those files remove their dependencies. The files in the directory are not actually deleted by this process. The directory can be deleted with the Delete File operating system mechanism.

The mechanism takes the pathname of a given directory, and optionally, the identification of a preferred alternate source processor for clients to use. The mechanism performs the following steps:

(A) Traverse the directory. For each file in the directory, perform the following steps:

(i) Get the True Name of the file from its path and find the True File registry entry 140 associated with the True Name.

30

35

5

10

15

20

25

<u>cetter cetter</u>

(ii) Determine an alternate source for the True File. If the gource IDs field of the TFR entry includes the preferred alternate source, that is the alternate source. If it does not, but includes some other source, that is the alternate source. If it contains no alternate sources, there is no alternate source.

5**9**

le i

(iii) For each dependent processor in the True File registry entry 140, ask that processor to retire the True File, specifying an alternate source if one was determined, using the remote mechanism.

5

10

15

20

25

1

6.

Realize Directory at Location

This mechanism allows the user or operating system to force copies of files from some source location to the True File registry 126 at a given location. The purpose of the mechanism is to ensure that files are accessible in the event the source location becomes

inaccessible. This can happen for instance if the source or given location are on mobile computers, or are on removable media, or if the network connection to the source is expected to become unavailable, or if the source is being retired.

This mechanism is provided in the following steps for each file in the given directory, with the exception of subdirectories:

(A) Get the local directory extensions table entry record 138 given the pathname of the file. Get the True Name of the local directory extensions table entry record 138. This service assimilates the file if it has not already been assimilated.

(B) Realize the corresponding True File at the given location. This service causes it to be copied to the given location from a remote system or removable media.

7. <u>Verify True File</u>

problems.

30

This mechanism is used to verify that the data item in a True File registry 126 is indeed the correct data item given its True Name. Its purpose is to guard against device errors, malicious changes, or other

35

If an error is found, the system has the ability to "heal" itself by finding another source for

60

 $\left| o \right|$

RACK-1002 Page 71 of 351

the True File with the given name. It may also be desirable to verify that the error has not propagated to other systems, and to log the problem or indicate it to the computer operator. These details are not described here.

5

To verify a data item that is not in a True File registry 126, use the Calculate True Name primitive mechanism described above.

The basic mechanism begins with a True Name, and operates in the following steps: 10

> (A) Find the True File registry entry record 140 corresponding to the given True Name.

(B) If there is a True File ID for the True File registry entry record 140 then use it. Otherwise, indicate that no file exists to verify.

(C) Calculate the True Name of the data item given the file ID of the data item.

(D) Confirm that the calculated True Name is equal to the given True Name.

20

1

15

(E) If the True Names are not equal, there is an error in the True File registry 126. Remove the True File ID from the True File registry entry record 140 and place it somewhere else. Indicate that the True File registry entry record 140 contained an error.

25

30

35

8.

Track for Accounting Purposes

This mechanism provides a way to know reliably which files have been stored on a system or transmitted from one system to another. The mechanism can be used as a basis for a value-based accounting system in which charges are based on the identity of the data stored or transmitted, rather than simply on the number of bits.

This mechanism allows the system to track possession of specific data items according to content by owner, independent of the name, date, or other properties of the data item, and tracks the uses of specific data items and files by content for accounting purposes. True

61

names make it possible to identify each file briefly yet uniquely for this purpose.

Tracking the identities of files requires maintaining an accounting log 134 and processing it for accounting or billing purposes. The mechanism operates in the following steps:

(A) Note every time a file is created or deleted, for instance by monitoring audit entries in the Process Audit File Entry primitive mechanism. When such an event is encountered, create an entry 148 in the accounting log 134 that shows the responsible party and the identity of the file created or deleted.

(B) Every time a file is transmitted, for instance when a file is copied with a Request True File remote mechanism or an Acquire True File remote mechanism, create an entry in the accounting log 134 that shows the responsible party, the identity of the file, and the source and destination processors.

(C) Occasionally run an accounting program to process the accounting log 134, distributing the events to the account records of each responsible party. The account records can eventually be summarized for billing purposes.

9. <u>Track for Licensing Purposes</u>

This mechanism ensures that licensed files are not used by unauthorized parties. The True Name provides a safe way to identify licensed material. This service allows proof of possession of specific files according to their contents without disclosing their contents.

30

5

10

15

20

25

Enforcing use of valid licenses can be active (for example, by refusing to provide access to a file without authorization) or passive (for example, by creating a report of users who do not have proper authorization).

35

One possible way to perform license validation is to perform occasional audits of employee systems. The

service described herein relies on True Names to support such an audit, as in the following steps:

(A) For each licensed product, record in the license table 136 the True Name of key files in the product (that is, files which are required in order to use the product, and which do not occur in other products) Typically, for a software product, this would include the main executable image and perhaps other major files such as clip-art, scripts, or online help. Also record the identity of each system which is authorized to have a copy of the file.

(B) Occasionally, compare the contents of each user processor against the license table 136. For each True Name in the license table do the following:

(i) Unless the user processor is authorized to have a copy of the file, confirm that the user processor does not have a copy of the file using the Locate True File mechanism.

(ii) If the user processor is found to have a file that it is not authorized to have, record the user processor and True Name in a license violation table.

The System in Operation

Given the mechanisms described above, the operation of a typical DP system employing these mechanisms is now described in order to demonstrate how the present invention meets its requirements and capabilities.

In operation, data items (for example, files, database records, messages, data segments, data blocks, directories, instances of object classes, and the like) in a DP system employing the present invention are identified by substantially unique identifiers (True Names), the identifiers depending on all of the data in the data items and only on the data in the data items.

63

RACK-1002 Page 74 of 351

£

5

10

15

20

25

30

The primitive mechanisms Calculate True Name and Assimilate Data Item support this property. For any given data item, using the Calculate True Name primitive mechanism, a substantially unique identifier or True Name for that data item can be determined.

5

Щ

1

Õ

Ó

30

35

Further, in operation of a DP system incorporating the present invention, multiple copies of data items are avoided (unless they are required for some reason such as backups or mirror copies in a faulttolerant system). Multiple copies of data items are

10 tolerant system). Multiple copies of data items are avoided even when multiple names refer to the same data item. The primitive mechanisms Assimilate Data Items and New True File support this property. Using the Assimilate Data Item primitive mechanism, if a data item

already exists in the system, as indicated by an entry in 15 the True File registry 126, this existence will be discovered by this mechanism, and the duplicate data item (the new data item) will be eliminated (or not added). Thus, for example, if a data file is being copied onto a system from a floppy disk, if, based on the True Name of 20 the data file, it is determined that the data file already exists in the system (by the same or some other name), then the duplicate copy will not be installed. If the data item was being installed on the system by some name other than its current name, then, using the Link 25 Path to True Name primitive mechanism, the other (or new) name can be linked to the already existing data item.

In general, the mechanisms of the present invention operate in such a way as to avoid recreating an actual data item at a location when a copy of that data item is already present at that location. In the case of a copy from a floppy disk, the data item (file) may have to be copied (into a scratch file) before it can be determined that it is a duplicate. This is because only one processor is involved. On the other hand, in a multiprocessor environment or DP system, each processor has a record of the True Names of the data items on that

64

65

RACK-1002 Page 75 of 351 processor. When a data item is to be copied to another location (another processor) in the DP system, all that is necessary is to examine the True Name of the data item prior to the copying. If a data item with the same True

5 Name already exists at the destination location (processor), then there is no need to copy the data item. Note that if a data item which already exists locally at a destination location is still copied to the destination location (for example, because the remote system did not have a True Name for the data item or because it arrives as a stream of un-named data), the Assimilate Data Item

as a stream of un-named data), the Assimilate bata feel primitive mechanism will prevent multiple copies of the data item from being created.

Since the True Name of a large data item (a compound data item) is derived from and based on the True Names of components of the data item, copying of an entire data item can be avoided. Since some (or all) of the components of a large data item may already be present at a destination location, only those components which are not present there need be copied. This property derives from the manner in which True Names are determined.

When a file is copied by the Copy File or Directory operating system mechanism, only the True Name of the file is actually replicated.

When a file is opened (using the Open File operating system mechanism), it uses the Make True File Local primitive mechanism (either directly or indirectly through the Create Scratch File primitive mechanism) to create a local copy of the file. The Open File operating system mechanism uses the Make True File Local primitive mechanism, which uses the Realize True File from Location primitive mechanism, which, in turn uses the Request True File remote mechanism.

35

25

30

Ű

The Request True File remote mechanism copies only a single data item from one processor to another. If the data item is a compound file, its component

65

segments are not copied, only the indirect block is copied. The segments are copied only when they are read (or otherwise needed).

The Read File operating system mechanism actually reads data. The Read File mechanism is aware of compound files and indirect blocks, and it uses the Realize True File from Location primitive mechanism to make sure that component segments are locally available, and then uses the operating system file mechanisms to read data from the local file.

5

10

15

20

25

30

35

1

Ű

Thus, when a compound file is copied from a remote system, only its True Name is copied. When it is opened, only its indirect block is copied. When the corresponding file is read, the required component segments are realized and therefore copied.

In operation data items can be accessed by reference to their identities (True Names) independent of their present location. The actual data item or True File corresponding to a given data identifier or True Name may reside anywhere in the system (that is, locally, remotely, offline, etc). If a required True File is present locally, then the data in the file can be accessed. If the data item is not present locally, there are a number of ways in which it can be obtained from wherever it is present. Using the source IDs field of the True File registry table, the location(s) of copies of the True File corresponding to a given True Name can be determined. The Realize True File from Location primitive mechanism tries to make a local copy of a True File, given its True Name and the name of a source location (processor or media) that may contain the True File. If, on the other hand, for some reason it is not known where there is a copy of the True File, or if the processors identified in the source IDs field do not respond with the required True File, the processor requiring the data item can make a general request for the data item using the Request True File remote

66

(c')

RACK-1002 Page 77 of 351 mechanism from all processors in the system that it can contact.

As a result, the system provides transparent access to any data item by reference to its data identity, and independent of its present location.

5

10

15

20

25

30

35

Ð

In operation, data items in the system can be verified and have their integrity checked. This is from the manner in which True Names are determined. This can be used for security purposes, for instance, to check for viruses and to verify that data retrieved from another location is the desired and requested data. For example, the system might store the True Names of all executable applications on the system and then periodically redetermine the True Names of each of these applications to ensure that they match the stored True Names. Any change in a True Name potentially signals corruption in the system and can be further investigated. The Verify Region background mechanism and the Verify True File extended mechanisms provide direct support for this mode of operation. The Verify Region mechanism is used to ensure that the data items in the True File registry have not been damaged accidentally or maliciously. The Verify True File mechanism verifies that a data item in a True File registry is indeed the correct data item given its True Name.

Once a processor has determined where (that is, at which other processor or location) a copy of a data item is in the DP system, that processor might need that other processor or location to keep a copy of that data item. For example, a processor might want to delete local copies of data, items to make space available locally while knowing that it can rely on retrieving the data from somewhere else when needed. To this end the system allows a processor to Reserve (and cancel the reservation of) True Files at remote locations (using the remote mechanism). In this way the remote locations are

67

108

RACK-1002 Page 78 of 351 put on notice that another location is relying on the presence of the True File at their location.

A DP system employing the present invention can be made into a fault-tolerant system by providing a certain amount of redundancy of data items at multiple locations in the system. Using the Acquire True File and Reserve True File remote mechanisms, a particular processor can implement its own form of fault-tolerance by copying data items to other processors and then reserving them there. However, the system also provides the Mirror True File background mechanism to mirror (make copies) of the True File available elsewhere in the system. Any degree of redundancy (limited by the number of processors or locations in the system) can be implemented. As a result, this invention maintains a desired degree or level of redundancy in a network of processors, to protect against failure of any particular processor by ensuring that multiple copies of data items exist at different locations.

The data structures used to implement various features and mechanisms of this invention store a variety of useful information which can be used, in conjunction with the various mechanisms, to implement storage schemes and policies in a DP system employing the invention. For example, the size, age and location of a data item (or of groups of data items) is provided. This information can be used to decide how the data items should be treated. For example, a processor may implement a policy of deleting local copies of all data items over a certain age if other copies of those data items are present elsewhere in the system. The age (or variations on the age) can be determined using the time of last access or modification in the local directory extensions table, and the presence of other copies of the data item can be determined either from the Safe Flag or the source IDs, or by checking which other processors in the system have

6**8**

69

10

5

15

20

~ 미난미 1

1 1.1

25

30

35

RACK-1002 Page 79 of 351 copies of the data item and then reserving at least one of those copies.

In operation, the system can keep track of data items regardless of how those items are named by users (or regardless of whether the data items even have names). The system can also track data items that have different names (in different or the same location) as well as different data items that have the same name. Since a data item is identified by the data in the item, without regard for the context of the data, the problems of inconsistent naming in a DP system are overcome.

In operation, the system can publish data items, allowing other, possibly anonymous, systems in a network to gain access to the data items and to rely on the availability of these data items. True Names are globally unique identifiers which can be published simply by copying them. For example, a user might create a textual representation of a file on system A with True Name N (for instance as a hexadecimal string), and post it on a computer bulletin board. Another user on system B could create a directory entry F for this True Name N by using the Link Path to True Name primitive mechanism. (Alternatively, an application could be developed which hides the True Name from the users, but provides the same public transfer service.)

When a program on system B attempts to open pathname F linked to True Name N, the Locate Remote File primitive mechanism would be used, and would use the Locate True File remote mechanism to search for True Name N on one or more remote processors, such as system A. If system B has access to system A, it would be able to realize the True File (using the Realize True File from Location primitive mechanism) and use it locally. Alternatively, system B could find True Name N by accessing any publicly available True Name server, if the server could eventually forward the request to system A.

69

10

15

20

25

30

5

, m

Clients of a local server can indicate that they depend on a given True File (using the Reserve True File remote mechanism) so that the True File is not deleted from the server registry as long as some client requires access to it. (The Retire True File remote mechanism is used to indicate that a client no longer needs a given True File.)

5

10

15

20

25

30

35

A publishing server, on the other hand, may want to provide access to many clients, and possibly anonymous ones, without incurring the overhead of tracking dependencies for each client. Therefore, a public server can provide expiration dates for True Files in its registry. This allows client systems to safely maintain references to a True File on the public server. The Check For Expired Links background mechanism allows the client of a publishing server to occasionally confirm that its dependencies on the publishing server are safe.

In a variation of this aspect of the invention, a processor that is newly connected (or reconnected after some absence) to the system can obtain a current version of all (or of needed) data in the system by requesting it from a server processor. Any such processor can send a request to update or resynchronize all of its directories (starting at a root directory), simply by using the Synchronize Directories extended mechanism on the needed directories.

Using the accounting log or some other user provided mechanism, a user can prove the existence of certain data items at certain times. By publishing (in a public place) a list of all True Names in the system on a given day (or at some given time), a user can later refer back to that list to show that a particular data item was present in the system at the time that list was published. Such a mechanism is useful in tracking, for example, laboratory notebooks or the like to prove dates of conception of inventions. Such a mechanism also

70

17/

permits proof of possession of a data item at a particular date and time.

The accounting log file can also track the use of specific data items and files by content for

5 accounting purposes. For instance, an information utility company can determine the data identities of data items that are stored and transmitted through its computer systems, and use these identities to provide bills to its customers based on the identities of the

10 data items being transmitted (as defined by the substantially unique identifier). The assignment of prices for storing and transmitting specific True Files would be made by the information utility and/or its data suppliers; this information would be joined periodically with the information in the accounting log file to 15 produce customer statements.

Backing up data items in a DP system employing the present invention can be done based on the True Names of the data items. By tracking backups using True Names, 20 duplication in the backups is prevented. In operation, the system maintains a backup record of data identifiers of data items already backed up, and invokes the Copy File or Directory operating system mechanism to copy only those data items whose data identifiers are not recorded 25 in the backup record. Once a data item has been backed up, it can be restored by retrieving it from its backup location, based on the identifier of the data item. Using the backup record produced by the backup to identify the data item, the data item can be obtained using, for example, the Make True File Local primitive 30 mechanism.

In operation, the system can be used to cache data items from a server, so that only the most recently accessed data items need be retained. To operate in this way, a cache client is configured to have a local registry (its cache) with a remote Local Directory Extensions table (from the cache server). Whenever a

71

72

RACK-1002 Page 82 of 351

file is opened (or read), the Local Directory Extensions table is used to identify the True Name, and the Make True File Local primitive mechanism inspects the local registry. When the local registry already has a copy,

the file is already cached. Otherwise, the Locate True File remote mechanism is used to get a copy of the file. This mechanism consults the cache server and uses the Request True File remote mechanism to make a local copy, effectively loading the cache.

10

15

25

30

35

ų. Ű

5

The Groom Cache background mechanism flushes the cache, removing the léast-recently-used files from the cache client's True File registry. While a file is being modified on a cache client, the Lock Cache and Update Cache remote mechanisms prevent other clients from trying to modify the same file.

In operation, when the system is being used to cache data items, the problems of maintaining cache consistency are avoided.

To access a cache and to fill it from its server, a key is required to identify the data item 20 desired. Ordinarily, the key is a name or address (in this case, it would be the pathname of a file). If the data associated with such a key is changed, the client's cache becomes inconsistent; when the cache client refers to that name, it will retrieve the wrong data. In order to maintain cache consistency it is necessary to notify every client immediately whenever a change occurs on the server.

By using an embodiment of the present invention, the cache key uniquely identifies the data it represents. When the data associated with a name changes, the key itself changes. Thus, when a cache client wishes to access the modified data associated with a given file name, it will use a new key (the True Name of the new file) rather than the key to the old file contents in its cache. The client will always request the correct data, and the old data in its cache will be

eventually aged and flushed by the Groom Cache background mechanism.

Because it is not necessary to immediately notify clients when changes on the cache server occur, the present invention makes it possible for a single server to support a much larger number of clients than is otherwise possible.

In operation, the system automatically archives data items as they are created or modified. After a file 10 is created or modified, the Close File operating system mechanism creates an audit file record, which is eventually processed by the Process Audit File Entry primitive mechanism. This mechanism uses the New True File primitive mechanism for any file which is newly created, which in turn uses the Mirror True File background mechanism if the True File is in a mirrored or archived region. This mechanism causes one or more copies of the new file to be made on remote processors.

In operation, the system can efficiently record and preserve any collection of data items. The Freeze Directory primitive mechanism creates a True File which identifies all of the files in the directory and its subordinates. Because this True File includes the True Names of its constituents, it represents the exact contents of the directory tree at the time it was frozen. The frozen directory can be copied with its components preserved.

The Acquire True File remote mechanism (used in mirroring and archiving) preserves the directory tree structure by ensuring that all of the component segments and True Files in a compound data item are actually copied to a remote system. Of course, no transfer is necessary for data items already in the registry of the remote system.

35

5

15

20

25

30

2

Ű

in the second se

T

Û Ð

> In operation, the system can efficiently make a copy of any collection of data items, to support a version control mechanism for groups of the data items.

The Freeze Directory primitive mechanism is used to create a collection of data items. The constituent files and segments referred to by the frozen directory are maintained in the registry, without any need to make copies of the constituents each time the directory is frozen.

Whenever a pathname is traversed, the Get Files in Directory operating system mechanism is used, and when it encounters a frozen directory, it uses the Expand Frozen Directory primitive mechanism.

A frozen directory can be copied from one pathname to another efficiently, merely by copying its True Name. The Copy File operating system mechanism is used to copy a frozen directory.

15

20

25

30

35

IJ

i Îl

10

5

Thus it is possible to efficiently create copies of different versions of a directory, thereby creating a record of its history (hence a version control system).

In operation, the system can maintain a local inventory of all the data items located on a given removable medium, such as a diskette or CD-ROM. The inventory is independent of other properties of the data items such as their name, location, and date of creation. The Inventory Existing Directory extended

mechanism provides a way to create True File Registry entries for all of the files in a directory. One use of this inventory is as a way to pre-load a True File registry with backup record information. Those files in the registry (such as previously installed software) which are on the volumes inventoried need not be backed up onto other volumes.

The Inventory Removable, Read-only Files extended mechanism not only determines the True Names for the files on the medium, but also records directory entries for each file in a frozen directory structure. By copying and modifying this directory, it is possible to create an on line patch, or small modification of an

74

existing read-only file. For example, it is possible to create an online representation of a modified CD-ROM, such that the unmodified files are actually on the CD-ROM, and only the modified files are online.

In operation, the system tracks possession of specific data items according to content by owner, independent of the name, date, or other properties of the data item, and tracks the uses of specific data items and files by content for accounting purposes. Using the Track for Accounting Purposes extended mechanism provides a way to know reliably which files have been stored on a

system or transmitted from one system to another.

True Names in Relational and Object-Oriented Databases Although the preferred embodiment of this

invention has been presented in the context of a file system, the invention of True Names would be equally valuable in a relational or object-oriented database. A relational or object-oriented database system using True Names would have similar benefits to those of the file system employing the invention. For instance, such a database would permit efficient elimination of duplicate records, support a cache for records, simplify the process of maintaining cache consistency, provide location-independent access to records, maintain archives and histories of records, and synchronize with distant or disconnected systems or databases.

The mechanisms described above can be easily modified to serve in such a database environment. The True Name registry would be used as a repository of database records. All references to records would be via the True Name of the record. (The Local Directory Extensions table is an example of a primary index that uses the True Name as the unique identifier of the desired records.)

75

76

RACK-1002 Page 86 of 351

ñ

30

5

10

15

20

In such a database, the operations of inserting, updating, and deleting records would be implemented by first assimilating records into the registry, and then updating a primary key index to map the key of the record to its contents by using the True Name as a pointer to the contents.

5

10

15

20

The mechanisms described in the preferred embodiment, or similar mechanisms, would be employed in such a system. These mechanisms could include, for example, the mechanisms for calculating true names, assimilating, locating, réalizing, deleting, copying, and moving True Files, for mirroring True Files, for maintaining a cache of True Files, for grooming True Files, and other mechanisms based on the use of substantially unique identifiers.

While the invention has been described in connection with what is presently considered to be the most practical and preferred embodiments, it is to be understood that the invention is not to be limited to the disclosed embodiment, but on the contrary, is intended to cover various modifications and equivalent arrangements included within the spirit and scope of the appended claims.

76

671

RACK-1002 Page 87 of 351

WHAT IS CLAIMED IS:

 In a data processing system, an apparatus prising:

comprising:

5

10

15

20

identity means for determining, for any of a plurality of data items in the system, a substantially unique identifier, said identifier depending on all of the data in the data item and only on the data in the data item; and

existence means for determining whether a particular data item is present in the system, by examining the identifiers of the plurality of data items.

2. An apparatus as in claim 1, further comprising:

local existence means for determining whether an instance of a particular data item is present at a particular location in the system, based on the identifier of the data item.

3. An apparitus as in claim 2, wherein each location contains a distinct plurality of data items, and wherein said local existence means determines whether a particular data item is present at a particular location in the system by examining the identifiers of the plurality of data items at said particular location in the system.

25

30

4. An apparatus as in claim 2, further comprising:

data associating means for making and maintaining, for a gata item in the system, an association between the data item and the identifier of the data item; and

access means for accessing a particular data item using the identifier of the data item.

5. An apparatus as in claim 2, further comprising:

5

duplication means for copying a data item from a source to a destination in the data processing system, by providing said destination with the data item only if it is determined using the data identifier that the data item is not present at the destination.

6. An apparatus as in claim 4, further comprising:

10

assimilation means for assimilating a new data item into the system, said assimilation means invoking said identity means to determine the identifier of the new data item and invoking said data associating means to associate the new data item with its identifier.

15

20

25

Defizio _ otrigo

7. An apparatus as in claim 4, further comprising:

duplication means for duplicating a data item from a source location to a destination location in the data processing system, based on the identifier of the data item, said duplication means invoking said local existence means to determine whether an instance of the data item is present at the destination location, and invoking said access means to provide said destination with the data item only if said local existence means determines that no instance of the data item is present at the destination.

8. An apparatus as in claim 7, further comprising:

30

backup means for making copies of data items in the system, said backup means maintaining a backup record of identifiers of data items backed up, and invoking duplication means to copy only those data items whose data identifiers are not recorded in the backup record.

9. An apparatus as in claim 8, further comprising:

recovery means for retrieving a data item previously backed up by said backup means, based on the identifier of the data item, said recovery means using the backup record to identify the data item, and invoking access means to retrieve the data item.

10. An apparatus as in claim 2, wherein a location is a computer among a network of computers, the apparatus further comprising:

remote existence means for determining whether a data item is present at a remote location in the system from a current location in the system, based on the identifier of the data item, said remote location using local existence means at the remote location to determine whether the data item is present at the remote location, and providing the current location with an indication of the presence of the data item at the remote location.

11. An apparatus as in claim 4, wherein a location is a computer among a network of computers, the apparatus further comprising:

requesting means for requesting a data item at a current location in the system from a remote location in the system, based on the identifier of the data item, said remote location using access means at the remote location to obtain the data item and to send it to the current location if it is present.

12. An apparatus as in claim 1 further comprising:

30

5

10

15

20

25

DSEELEE D.J.C.L

Ē

context means for making and maintaining a context association between at least one contextual name of a data item in the system and the identifier of the data item; and referencing means for obtaining the identifier of a data item in the system given a contextual name for the data item, using said context association.

13. An apparatus as in claim 12, further comprising:

assignment means for assigning a data item to a contextual name, invoking said identity means to determine the identifier of the data item, and invoking said context means to make or modify the context association between the contextual name of the data item and the identifier of the data item.

14. An apparatus as in claim 12, further comprising:

data associating means for making and maintaining, for a data item in the system, an association between the data item and the identifier of the data item;

access means for accessing a particular data item using the identifier of the particular data item; and

contextual name access means for accessing a data item in the system for a given context name of the data item, determining the data identifier associated with the given context name, and invoking said access means to access the data item using the data identifier.

15. An apparatus as in claim 11, further comprising:

30

5

10

15

20

25

- C

transparent access means for accessing a data item from one of several locations, using the identifier of the data item, said transparent access means invoking said local existence means to determine if the particular data item is present at the current location, and, in the case when the particular data item is not present at the current location, invoking said requesting means to obtain the data item from a remote location.

16. An apparatus as in claim 15, further comprising:

5

identifier copy means for copying an identifier of a data item from a source location to a destination location.

17. An apparatus as in claim 15, further comprising:

10

15

20

25

30

Q

ų

Ð

in t

context means for making and maintaining a context association between a contextual name of a data item in the system and the identifier of the data item; context copy means for copying a data item from

a source location to a destination location, given the contextual name of the data item, by copying only the context association between the contextual identifier and the data identifier from the source location to the destination location; and

transparent referencing means for obtaining a data item from one of several locations the system given a contextual name for the data item, said transparent referencing means invoking said context association to determine the data identifier of a data item given a contextual name, and invoking said transparent access means to access the data item from one of several locations given the identifier of the data item.

18. An apparatus as in claim 1, wherein at least some of said data items are compound data items, each compound data item including at least some component data items in a fixed sequence, and wherein the identity means determines the identifier of a compound data item based on each component data item of the compound data item. 19. An apparatus as in claim 18, wherein said compound data items are files and said component data items are segments, and wherein the identity means determines the identifier of a file based on the identifier of each data segment of the file.

5

10

15

20

25

30

 20. An apparatus as in claim 18, wherein said compound data items are directories and said component data items are files or subordinate directories, and wherein the identity means determines the identifier of a given directory based on each file and subordinate directory within the given directory.

21. An apparatus as in claim 11, further comprising:

means for advertising a data item from a location in the system to at least one other location in the system, said means for advertising providing each of said at least one other location with the data identifier of the data item, and providing the data item to only those locations of said other locations that request said data item in response to said providing.

22. An apparatus as in claim 18, further comprising:

local existence means for determining whether a particular data item is present at a particular location in the system, based on the identifier of the data item; and

compound copy means for copying a data item from a source to a destination in the data processing system, said compound copy means invoking said local existence means to determine whether the data item is present at the destination, and to determine, when the data item is a compound data item, whether the component data items of the compound data item are present at the destination, and providing said destination with the data

item only if said local existence means determines that the data item is not present at the destination, and providing said destination with each component data item only if said local existence means determines that the component data item is not present at the destination.

23. An apparatus as in claim 11, further comprising:

means for verifying the integrity a data item obtained from said requesting means in response to providing said requesting with a particular data identifier, to confirm that the data item obtained from the requesting means is the same data item as the data item requested, said verifying means invoking said identity means to determine the data identifier of the obtained data item, and comparing said determined data identifier with said particular data identifier to verify said obtained data item.

24. An apparatus as in claim 2, wherein a location is at least one of a storage location and a processing location, and wherein a storage location is at least one of a data storage device and a data storage volume, and wherein a processing location is at least one of a data processor and a computer.

25

30

5

10

15

20

1

25. An apparatus as in claim 3, wherein at least some of said data items are compound data items, each compound data item including at least some component data items in a fixed sequence, and wherein the identity means determines the identifier of a compound data item based on the identifier of each component data item of the compound data item.

26. An apparatus as in claim 3, further comprising:

context associating means for making and maintaining a context association, for any data item in the system, between the identifier of the data item and at least one contextual name of the data item at a particular location in the system;

means for obtaining the identifier of a data item in the system given a contextual name for the data item at a particular location in the system; and

10

5

logical copy means for associating the data identifier corresponding to a contextual name at a source location with a contextual name at a destination location in the data processing system.

27. An apparatus as in claim 25, wherein said compound data items are files and said component data items are segments, and wherein the identity means determines the identifier of a file based on the identifier of each data segment of the file.

28. An apparatus as in claim 25, further comprising:

20

15

Щ

in the second second m

83

-

25

30

compound copy means for copying a data item from a source location to a destination location in the data processing system, said compound copy means invoking said local existence means to determine whether the data item is present at the destination, and to determine, when the data item is a compound data item, whether the component data items of the compound data item are present at the destination, and providing said destination with the data item only if said local existence means determines that the data item is not present at the destination, and providing said destination with each component data item only if said local existence means determines that the component data item is not present at the destination.

RACK-1002 Page 95 of 351

29. An apparatus as in any of claims 1-28, wherein a data item is at least one of a file, a database record, a message, a data segment, a data block, a directory, and an instance an object class.

30. A method of identifying a data item in a data processing system for subsequent access to the data item, the method comprising the steps of:

determining a substantially unique identifier for the data item, said identifier depending on all of the data in the data item and on the data in the data item; and

accessing a data item in the system using the identifier of the data item.

1

Ū

đ

j.

5

10

15

20

25

31. A method as in claim 30, further comprising the step of:

making and maintaining, for a plurality of data items in the system, an association between each of the data items and the identifier of each of the data items, wherein said accessing step accesses a data item via the association.

32. A method as in claim '31, further comprising the step of:

assimilating a new data item into the system, by determining the identifier of the new data item and associating the new data item with its identifier.

33. A method for duplicating a given data item from a source location to a destination location in a data processing system, the method comprising the steps of:

30

determining a substantially unique identifier for the given data item, said identifier depending on all of the data in the data item and only on the data in the data item; determining, using said data identifier, whether said data item is present at said destination location; and

based on said determining, providing said destination location with said data item only if said data item is not present at said destination.

34. A method as in claim 33, wherein said given data item is a compound data item having a plurality of component data items, the method further comprising the steps of:

for each data item of said component data

items,

5

10

15

20

25

ŝ

obtaining the component data identifier of the data item by determining a substantially unique identifier for the data item, said identifier depending on all of the data in the data item and only on the data in the data item;

determining, using said obtained component data identifier, whether said data item is present at said destination; and

based on said determining, providing said destination with said data item only if said data item is not present at said destination.

35. A method for determining whether a particular data item is present in a data processing system, the method comprising the steps of:

(A) for each data item of a plurality of data items in the system.

86

(i) determining a substantially unique identifier for the data item, said identifier depending on all of the data in the data item and only on the data in the data item; and

(ii) making and maintaining a set of identifiers of said plurality of data items; and

(B) for the particular data item,

(i) determining a particular

substantially unique identifier for the data item, said identifier depending on all of the data in the data item and only on the data in the data item; and

(ii) determining whether said particular identifier is in said set of data items.

36. A method of backing up, of a plurality of data items, data items modified since a previous backup time in a data processing system, the method comprising the steps of:

> (A) maintaining a backup record of identifiers of data items backed up at the previous backup time; and

(B) for each of said plurality of data items,

identifier for the data item, said

20

15

ā kļ

5

10

25

30

identifier depending on all of the data in the data item and only on the data in the data item; (ii) determining those data items of the plurality of data items whose identifiers are not in the backup record; and (iii) based on said determining, copying

determining a substantially unique

only those data items whose data identities are not recorded in the backup record.

37. A method as in claim 36, further comprising the step of:

(i)

recording in the backup record the identifiers of those data items copied in said step of copying.

5

10

15

20

25

30

35

IJ

į...

38. A method of locating a particular data item at a location in a data processing system, the method comprising the steps of:

(A) determining a substantially unique identifier for the data item, said identifier depending on all of the data in the data item and only on the data in the data item;
(B) requesting the particular data item by sending the data identifier of the data item from the requestor location to at least one location of a plurality of provider locations in the system; and

(C) on at least some of said provider locations,

(a) for each data item of a plurality of data items at said provider locations,
(i) determining a substantially unique identifier for the data item, said identifier depending on all of the data in the data item and only on the data in the data item; and

(ii) making and maintaining a set of identifiers of data items,

(b) determining, based on said set of identifiers, whether the data item corresponding to the requested data identifier is present at said provider location; and

(c) ~based on said determining, when said provider location determines that the particular data item is present at the provider location, notifying said requestor that the provider has a copy of the given data item.

39. The method of claim 38, further comprising

(a) for each data item of a plurality of data items at said provider locations,

making and maintaining an association between the data item and the identifier of the data item,

(b) in response to said notifying, said client location copying said data item from one of said responding remote locations, using said association to access the data item given the data identifier.

40. A method of locating a particular data item among a plurality of locations, each of said locations having a plurality of data items, the method comprising the steps of:

determining, for the particular data item and for each data item of the plurality of data items, a substantially unique identifier for the data item, said identifier depending on all of the data in the data item and only on the data in the data item; and

determining the presence of the particular data item in each of said plurality of locations by determining whether the identifier of the particular data item is present at each of said locations.

41. The method of claim 30, wherein said step of accessing further comprises the steps of, for a given data identifier and for a given current location and a remote location in the system:

30

5

10

15

20

25

di la

determining whether the data item corresponding to the given data identifier is present at the current location, and

based on said determining, if said data item is not present at the current location, fetching the data

item from a remote location in the system to the current location.

42. The method of claim 41, further comprising the steps of:

for each contextual name at a location, making and maintaining a context

association between the context name of a data item and the identifier of said data item, and when some context association changes at said current location, and

notifying said remote location of a modification to the context association.

43. The method of claim 42, further comprising the step of:

at said remote location, updating the association between the contextual identifier of the data item and the identifier of the data item.

44. The method of dlaim 43, further comprising the step of:

from said remote location, notifying all other locations that said data item has been modified, by providing the contextual identifier and data identifier of said data item to said other locations.

45. The method of claim 44, further comprising the step of, at each location notified that the data item has been modified:

modifying an association between the contextual identifier of the data item and the data identifier of the data item, to record that the data item has been modified.

30

5

10

15

20

25

10 F. 4

46. A method of eliminating a data item at a given location in a data processing system when said data

item can be obtained from another location in the system, the method comprising the steps of:

determining a substantially unique identifier for the data, said identifier depending on all of the data in the data item and only on the data in the data item;

making and maintaining a source association between the data identifier and at least one location at which said data item is known to be present; and

10

15

20

5

based on said source association, if said data item is present at said other location, removing the data item from the given location.

47. A method of deleting a data item from a location in a data processing system, the method comprising the steps of

for each of a plurality of data items in the system:

determining a substantially unique identifier for the data, said identifier depending on all of the data in the data item and only on the data in the data item; and

making and maintaining, an association between each of the data items and the unique identifier of the data items; and

25

for a given data item:

determining a substantially unique identifier for the data, said identifier depending on all of the data in the data item and only on the data in the data item; and

30

35

determining whether a contextual identifier or a compound data item or a remote processor in the system refers to the unique identifier of the data item, and based on said determining, deleting said data item and its association if no other contextual identifier or compound data item or remote processor refers to said data item.

48. The method of claim 47, wherein said determining is based on a use count for the data item, and wherein said data item is deleted only if said use count indicates that no other contextual identifier or compound data item or remote processor in the system refers to the data item.

49. A method of substantially synchronizing data items at a client location in a data processing system after a period of independent changes on the client and another location in the system, given a context, the method comprising the steps of:

making and maintaining a list of changes to the context association between each context name of a data item and the identifier of said data item, in the given context and during the period of independent change; obtaining the list of changes from the other location for the given context; and,

for each context name in the list of changes updating the context identifier

associations at the client whenever it is determined that the context association of the given context name changed either only at the client or only at the other location during the period if independent changes; and

performing a conflict-resolution task such as notifying an operator of the client location, whenever it is determined that the context association changed at both the client and the other location.

50. A method as in claim 49, wherein said lists are maintained as queues based on a temporal order, and wherein, at said client location, said replacing is based on said temporal order.

51. A method of maintaining at least a predetermined number of copies of a given data item in a data processing system, at different locations in the

92

10

15

20

25

30

5

ي. ويتعاقب

/

data processing system, said data processing system being one wherein data is identified by a substantially unique identifier, said identifier depending on all of the data in the data item and only on the data in the data item, and wherein any data item in the system may be accessed using only the identifier of the data item, the method comprising the steps of:

(i) sending, from a first location in the system, the data identifier of the given data item to other locations in the system; and
(ii) in response to said sending, at each of said other locations,

(A) determining whether the data item corresponding to the data identifier is present at the other location, and based on said determining, and

(B) informing said first location whether said data item is present at the other location; and (iii) in response to said informing from said other locations, at said first location,
(A) determining whether said data item is present in at least the predetermined number of other locations, and based on said determining,
(B) when less than the predetermined number of other locations have a copy of the data item, requesting some locations that do not have a copy of the data item make a copy of the data item.

In the second

(Internet

1. J. J. J. J.

Harry Harris

0

. F 5

10

15

20

25

30

52. A method as in claim 51, wherein said step (iii) further comprises the step of:

(C) when more than the predetermined number of other locations have a copy of the data item, requesting some locations that do have a copy of the data item delete the copy of the data item.

53. A method as in any of claims 30-52, wherein said data items are at least one of a file, a database record, a message, a data segment, a data block, a directory, and an instance of an object class.

ADD B17 ADD C27

al a

OSTOND CALEBOOD

RACK-1002 Page 105 of 351

1

INS. CA ABSTRACT OF THE DISCLOSURE IDENTIFYING DATA IN A DATA PROCESSING SYSTEM

In a data processing system, a mechanism identifies data items by substantially unique identifiers which depend on all of the data in the data items and only on the data in the data items. Existence means determine whether a particular data item is present in the system, by examining the identifiers of the plurality of data items.

5

95

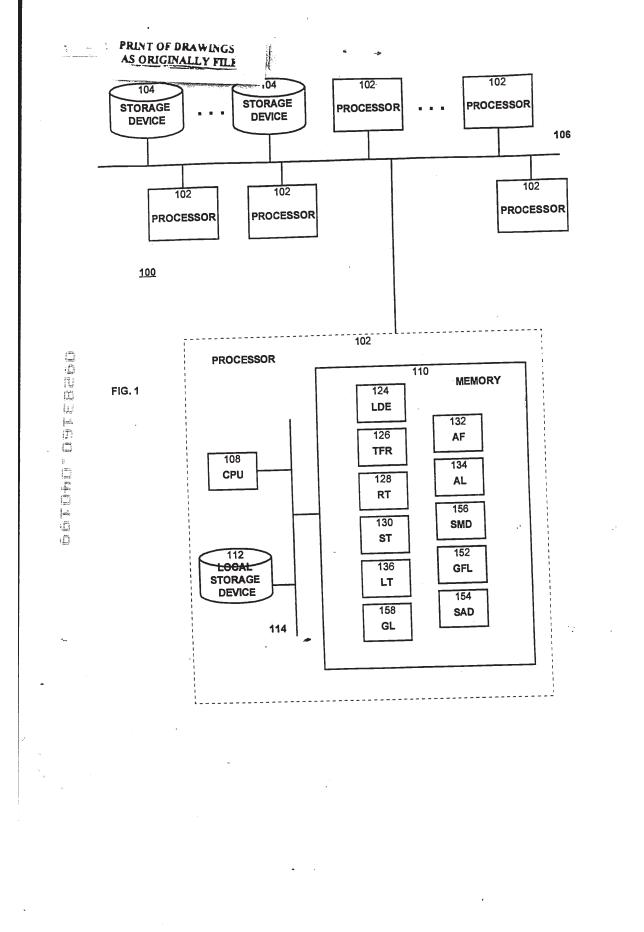
RACK-1002 Page 106 of 351

CIP/PCT I	FILITY/DESIGN	Π	DECLAR	RULE 63 (37 C.F.R ATION AND POWER	OF ATTORNEY		CUSH
ORIGINAL/SUBSTITUTE/SUPPLEMENTAL DECLARATIONS			FOR PATENT APPLICATION IN THE UNITED STATES PATENT AND TRADEMARK OFFICE				FOR
matter which is claim <u>IDENTIFYING DA</u>	ed and for which TA IN A DATA	a patent is sought on the PROCESSING SYSTE	he <u>INVENI</u>	e address and citizens n original, first and jo HON ENITILED	hip are as stated b int inventor (if plu	below next to my name, an aral names are listed below	d I believ v) of the
the specifica	tion of which (CI attached hereto.	<u>IECK</u> applicable BOX	<u>ES)</u>)				
X -> [x] w	as filed on Apr	il 11, 1995		_ as U.S. Application	No. 0 8 / 425.160	1	
BOX(ES) -> [] w -> -> <u>and (if app</u>	as filed as PCT In licable to U.S. or	nternational Application PCT application) was			On		
claim foreign priority below any foreign appl	benefits under 35 ication for patent	U.S.C. 119/365 of any f	foreign appl	ication(s) for patent of	or inventor's certifi	e claims, as amended by a ty as defined in 37 C.F.R. icate listed below and have tter claimed in this applica he filing date of this appli	1.56. I : also ide
PRIOR FOREIGN A	PPLICATION(S)			Date first Laid-	Date Patente		
Number	Country	Day/MONTH/Year	Filed	open or Published	or Granted		<u>No</u>
mprisonment, or both, application or any pate	under Section 10	01 of Title 18 of the L	wledge are knowledge nited States	true and that all state that willful false stat s Code and that such	ements made on ir ements and the li willful false stater	nformation and belief are ike so made are punishal nents may jeopardize the	believed ble by fi validity o
And I hereby appoint C	ushman Darby & hom all communi	Cushman, L.L.P. 1100 N Cations are to be direct	New York A ted), and th	venue, N.W., Ninth Fl he below-named perso	oor, East Tower Vons (of the same a	Washington, D.C. 20005-39 ddress) individually and c	918, telep
attorners to proceedite	hem to act and re	ly on instructions from a	and commu	nicht and Trademark	Office connected	therewith and with the res attorney/firm/ organization o be represented unless/u	sulting pa
attorneys to prosecute and I hereby authorize t	hman in writing	to the contrary.					
attorneys to prosecute and I hereby authorize t first sends/sent this cas Cushman, Darby & Cush Paul N. Kokulis	hman in writing 16773 J	Edward M. Prince	22429	Dale S. Lazar	28872	Michelle N. Lester	325
atorneys to prosecute and I hereby authorize t first sends/sent this cas Cushman, Darby & Cus Paul N. Kokulis Raymond F. Lippitt G. Lloyd Knight	shman in writing 16773 I 17519 I	to the contrary.	23048	Dale S. Lazar Glenn J. Perry	28458	Michelle N. Lester Jeffrey A. Simenauer	319
atlorneys to prosecute and I hereby authorize t first sends/sent this cas Cushman, Darby & Cus Paul N. Kokulis Raymond F. Lippitt G. Lloyd Knight Carl G. Love	hman in writing 16773 I 17519 I 17698 I 18781 C	Edward M. Prince Donald B. Deaver David W. Brinkman George M. Sirilla	23048 20817 18221	Dale S. Lazar Glenn J. Perry Kendrew H. Colto Chris Comuntzis	n 28458 30368 31097	Jeffrey A. Simenauer Robert A. Molan	319 298
atorneys to prosecute and I hereby authorize t first sends/sent this cas Cushman, Darby & Cus Paul N. Kokulis Raymond F. Lippitt G. Lloyd Knight	shman in writing 16773 I 17519 I 17698 I 18781 C 20534 I	Edward M. Prince Donald B. Deaver David W. Brinkman George M. Sirilla Donald J. Bird	23048 20817 18221 25323	Dale S. Lazar Glenn J. Perry Kendrew H. Colto Chris Comuntzis Wallace G. Walter	28458 n 30368 31097 27843	Jeffrey A. Simenauer Robert A. Molan G. Paul Edgell Lynn E. Eccleston	319 298 242 358
atlorneys to prosecute and I hereby authorize t first sends/sent this cas Cushman, Darby & Cus Paul N. Kokulis Raymond F. Lippitt G. Lloyd Knight Carl G. Love Edgar H. Martin William K. West, Jr. Kevin E. Jovce	shman in writing 16773 I 17519 I 17698 I 18781 C 20534 I 22057 V 20508 OF	Edward M. Prince Donald B. Deaver David W. Brinkman George M. Sirilla	23048 20817 18221	Dale S. Lazar Gienn J. Perry Kendrew H. Colto Chris Comuntzis Wallace G. Walter Lawrence Harbin	28458 an 30368 31097 27843 27644	Jeffrey A. Simenauer Robert A. Molan G. Paul Edgell Lynn E. Eccleston Frederick S. Frei	319 298 242 358 271
atorneys to prosecute and I hereby authorize t first sends/sent this cas Cushman, Darby & Cui Paul N. Kokulis Raymond F. Lippitt G. Lloyd Knight Carl G. Love Edgar H. Martin William K. West, Jr. Kevin E. Joyce 1. INVENTOR'S SIGN	shman in writing 16773 II 17519 II 17698 II 18781 C 20534 II 22057 V 20508 II 20508 II 20508 II ATURE: V	Edward M. Prince Donald B. Deaver David W. Brinkman Beorge M. Sirilla Donald J. Bird V. Warren Taltavull teter W. Gowdey	23048 20817 18221 25323 25647 25872	Dale S. Lazar Glenn J. Perry Kendrew H. Colto Chris Comuntzis Wallace G. Walter Lawrence Harbin Paul E. White, Jr.	28458 n 30368 31097 27843 27644 32011 Date_ <u>7677</u>	Jeffrey A. Simenauer Robert A. Molan G. Paul Edgell Lynn E. Eccleston Frederick S. Frei David A. Jakopin	319 298 242 358 271
atlorneys to prosecute and I hereby authorize t first sends/sent this cas Cushman, Darby & Cus Paul N. Kokulis Raymond F. Lippitt G. Lloyd Knight Carl G. Love Edgar H. Martin William K. West, Jr. Kevin E. Jovce	shman in writing 16773 II 17519 I 17698 I 18781 C 20534 I 20534 V 20508 F ATURE: V bad Daw bad Daw	Edward M. Prince Donald B. Deaver David W. Brinkman Beorge M. Sirilla Donald J. Bird V. Warren Taltavull teter W. Gowdey rid	23048 20817 18221 25323 25647 25872 4.	Dale S. Lazar Glenn J. Perry Kendrew H. Colto Chris Comuntzis Wallace G. Walter Lawrence Harbin Paul E. White, Jr.	28458 n 30368 31097 27843 27644 32011 Date	Jeffrey A. Simenauer Robert A. Molan G. Paul Edgell Lynn E. Eccleston Frederick S. Frei David A. Jakopin Mark G. Paulson U.S.A.	319 298 242 358 271 307
atorneys to prosecute and I hereby authorize t first sends/sent this cas Cushman, Darby & Cus Paul N. Kokulis Raymond F. Lippitt G. Lloyd Knight Carl G. Love Edgar H. Martin William K. West, Jr. Kevin E. Joyce 1. INVENTOR'S SIGN Inventor's Name (typ Residence (City)	thman in writing 16773 J 17519 I 17698 I 20534 I 22057 V 20508 F ATURE: V cd) Dav Fi Ojaj	Edward M. Prince Donald B. Deaver David W. Brinkman Beorge M. Sirilla Donald J. Bird V. Warren Taltavull eter W. Gowdey Current G. Fruct id	23048 20817 18221 25323 25647 25872 <u>4.</u> Middle Ini	Dale S. Lazar Glenn J. Perry Kendrew H. Colto Chris Comuntzis Wallace G. Walter Lawrence Harbin Paul E. White, Jr.	28458 n 30368 31097 27843 27644 32011 Date 6/7/ FARBER Family Name	Jeffrey A. Simenauer Robert A. Molan G. Paul Edgell Lynn E. Eccleston Frederick S. Frei David G. Paulson	319 298 242 358 271 307
attorneys to prosecute add I hereby authorize t first sends/sent this cas Cushman, Darby & Cus Paul N. Kokulis Raymond F. Lippitt G. Lloyd Knight Carl G. Love Edgar H. Martin William K. West, Jr. Kevin E. Joyce 1. INVENTOR'S SIGN Inventor's Name (typ	thman in writing 16773 J 17519 I 17698 I 20534 I 20534 I 20534 F 20508 F ATURE: V red) Dav Fi Ojai clude Zip Code) 4	Edward M. Prince Donald B. Deaver David W. Brinkman Beorge M. Sirilla Donald J. Bird V. Warren Taltavull eter W. Gowdey Carrier G. Frank id 1990 Carillo Rdy Ojali, 1990 Carillo Rdy Ojali,	23048 20817 18221 25323 25647 25872 	Dale S. Lazar Glenn J. Perry Kendrew H. Colto Chris Comuntzis Wallace G. Walter Lawrence Harbin Paul E. White, Jr.	28458 30368 31097 27843 27644 32011 	Jeffrey A. Simenauer Robert A. Molan G. Paul Edgell Lynn E. Eccleston Frederick S. Frei David A. Jakopin Mark G. Paulson U.S.A.	32: 319 298 242 358 271 307
attorneys to prosecute and I hereby authorize t first sends/sent this cas Cushman, Darby & Cus Paul N. Kokulis Raymond F. Lippitt G. Lloyd Knight Carl G. Love Edgar H. Martin William K. West, Jr. Kevin E. Joyce 1. INVENTOR'S SIGN Inventor's Name (typ Residence (City)	shman in writing 16773 I 17519 I 17698 I 18781 C 20534 I 22057 V 20508 F ATURE: \vee Dav Fi Ojai Clude Zip Code) 2c:	Edward M. Prince Donald B. Deaver David W. Brinkman George M. Sirilla Donald J. Bird V. Warren Taltavull eter W. Gowdey, Carrier G. G. Frank id 100 100 100 100 100 100 100 10	23048 20817 18221 25323 25647 25872 50 Middle Ini (State	Dale S. Lazar Glenn J. Perry Kendrew H. Colto Chris Comuntzis Wallace G. Walter Lawrence Harbin Paul E. White, Jr.	28458 30368 31097 27843 27644 32011 Date 6 21 FARBER FARBER Family Name CA	Jeffrey A. Simenauer Robert A. Molan G. Paul Edgell Lynn E. Eccleston Frederick S. Frei David A. Jakopin Mark G. Paulson U.S.A.	319 298 242 358 271 307
atorneys to prosecute and I hereby authorize t first sends/sent this cas Cushman, Darby & Cur Paul N. Kokulis Raymond F. Lippitt G. Lloyd Knight Carl G. Love Edgar H. Martin William K. West, Jr. Kevin E. Joyce 1. INVENTOR'S SIGN Inventor's Name (typ Residence (City) Post Office Address (In	16773 1 17519 1 17519 1 17698 1 18781 0 20534 1 22057 V Vature: V Ojai Fi Ojai 2 Clude Zip Code) 2 Atture: 2 ed) Ronald	Edward M. Prince Donald B. Deaver Donald B. Deaver David W. Brinkman Deorge M. Sirilla Donald J. Bird V. Warren Taltavull ver W. Gowdey <u>ver School</u> <u>ver School</u>	23048 20817 18221 25323 25647 25872 4. Middle Ini (State CA 93023 P () D.	Dale S. Lazar Glenn J. Perry Kendrew H. Colto Chris Comuntzis Wallace G. Walter Lawrence Harbin Paul E. White, Jr.	28458 30368 31097 27843 27644 27644 22011 Date <u>671</u> 5 FARBER Family Name CA	Jeffrey A. Simenauer Robert A. Molan G. Paul Edgell Lynn E. Eccleston Frederick S. Frei David G. Palsgoin U.S.A. Country of Cit	319 298 242 358 271 307
atorneys to prosecute and I hereby authorize t first sends/sent this cas Cushman, Darby & Cus Paul N. Kokulis Raymond F. Lippitt G. Lloyd Knight Carl G. Love Edgar H. Martin William K. West, Jr. Kevin E. Joyce 1. INVENTOR'S SIGN Inventor's Name (typ Residence (City) Post Office Address (In 2. INVENTOR'S SIGN Inventor's Name (typ	thman in writing 16773 J 17519 J 17698 J 20534 J 20534 J 20508 T ATURE:	Edward M. Prince Donald B. Deaver Donald B. Deaver David W. Brinkman Deorge M. Sirilla Donald J. Bird V. Warren Taltavull ver W. Gowdey <u>ver School</u> <u>ver School</u>	23048 20817 18221 25323 25647 25872 50 Middle Ini (State CA 93023 P P)	Dale S. Lazar Glenn J. Perry Kendrew H. Colto Chris Comuntzis Wallace G. Walter Lawrence Harbin Paul E. White, Jr.	28458 30368 31097 27843 27644 32011 Date 6 21 FARBER FARBER Family Name CA	Jeffrey A. Simenauer Robert A. Molan G. Paul Edgell Lynn E. Eccleston Frederick S. Frei U.S.A. Country of Cit	319 298 242 358 277 367 367
atorneys to prosecute add I hereby authorize t first sends/sent this cas Cushman, Darby & Cus Paul N. Kokulis Raymond F. Lippitt G. Lloyd Knight Carl G. Love Edgar H. Martin William K. West, Jr. Kevin E. Joyce 1. INVENTOR'S SIGN Inventor's Name (typ Residence (City) Post Office Address (In 2. INVENTOR'S SIGN	shman in writing 16773 I 17519 I 17598 I 17698 I 18781 C 20534 I 20534 I 20534 I 20534 I 20534 I 2053 F ATURE: V Oiai Clude Zip Code) - Clude Zip Code) - Fi ATURE: Fi chuber Sin Code - Fi prthbrook	Edward M. Prince Donald B. Deaver Donald B. Deaver David W. Brinkman George M. Sirilla Donald J. Bird W. Warren Taltavull teter W. Gowdey Charles G. Trail di Teter W. Gowdey Carillo Rd, Olai, 26 M. Garillo Rd, Olai, 26 M. Garillo Rd, Olai, March M. Start March M. Start Start March M. Start Sta	23048 20817 18221 25323 25647 25872 25772	Dale S. Lazar Glenn J. Perry Kendrew H. Colto Chris Comuntzis Wallace G. Walter Lawrence Harbin Paul E. White, Jr.	28458 a) 30368 31097 27843 27644 232011 Date 6/70 FARBER Family Name CA Date 6/70 CA CA CA ACHMAN Family Name	Jeffrey A. Simenauer Robert A. Molan G. Paul Edgell Lynn E. Eccleston Frederick S. Frei David G. Palsgoin U.S.A. Country of Cit	319 298 242 358 277 367 367
 altorneys to prosecute jaid I hereby authorize t first sends/sent this cas first sends/sent this cas Cushman, Darby & Cush Raymond F. Lippitt G. Lloyd Knight Carl G. Love Edgar H. Martin William K. West, Jr. Kevin E. Joyce I. INVENTOR'S SIGN Inventor's Name (typ Residence (City) Post Office Address (In 1. INVENTOR'S SIGN Inventor's Name (typ Residence (City) Name (typ Residence (City)) 	16773 1 17519 1 17519 1 17698 1 18781 0 20534 1 22050 V PATURE: V opiai 1 clude Zip Code) 2 ATURE: 2 ed) Ronald Fi 1 orthbrook 1	Edward M. Prince Donald B. Deaver Donald B. Deaver David W. Brinkman George M. Sirilla Donald J. Bird W. Warren Taltavull teter W. Gowdey Charles G. Frank dd Tst 200 Carillo Rd, Olai, 26 N. Grillo Rd, Olai, 26 N. Grillo Rd, Olai, Market M. Starter Market M. Starte	23048 20817 18221 25323 25647 25872 25772	Dale S. Lazar Glenn J. Perry Kendrew H. Colto Chris Comuntzis Wallace G. Walter Lawrence Harbin Paul E. White, Jr.	28458 a 30368 31097 27843 27644 232011 Date 6/10 FARBER Family Name CA Date 6/10 ACHMAN Family Name	Jeffrey A. Simenauer Robert A. Molan G. Paul Edgell Lynn E. Eccleston Frederick S. Frei U.S.A. Country of Cit	319 298 242 358 271 307 307

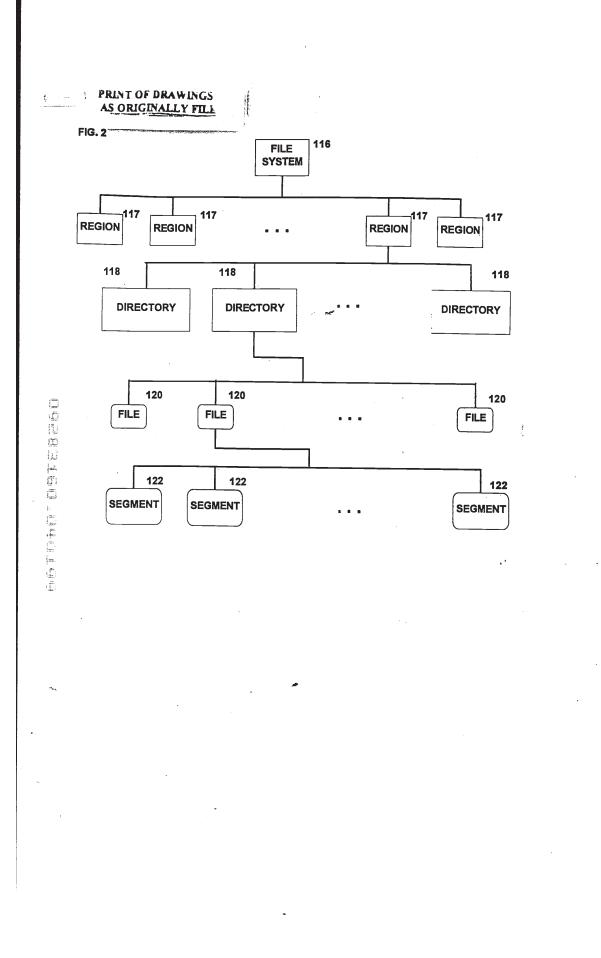
,

 \bigcirc \bigcirc ADDITIONAL INVENTORS, check box [] and attach sheet (CDC-116.2) for same information for each re signature, name, date, citizenship, residence and address.)

Inventor(s): David A. Farber and Ronald D. Lachman Appln. No.: 0 8 /425,160 or Patent No.:	(Atty. Dkt. (213987 /
Filed: April 11, 1995 or Issued:	M# / Client Ref.
VERIFIED STATEMENT (DECLARATION) CLAIN STATUS (37 CFR 1.9(d) and 1.27(c)) - <u>SMALL B</u>	
I hereby declare that I am	
 [w] the owner of the small business concern identified below: [] an official of the small business concern empowered to act 	on behalf of the concern identified below:
NAME OF CONCERN <u>KINETECH INC.</u> ADDRESS OF CONCERN 2 02N Carillo Rd., Ojai, California 9302 3	DF 6/7/95
3140 whisperweeds Ct., North broc	k Illinois 600 62
<u>I hereby declare</u> that the above identified small business concern qualifies a 121.12, and reproduced in 37 CFR 1.9(d), for purposes of paying reduced fe States Code, in that the number of employees of the concern, including the	ees under Section 41(a) and (b) of Title 35, United
For purposes of this statement, (1) the number of employees of the business of	concern is the average over the previous fiscal year
of the concern of the persons employed on a full-time, part-time or temporary year, and (2) <u>concerns are affiliates</u> of each other when either, directly or inc <u>control</u> the other, or a third party or parties controls or has the power to con	directly, one concern controls or has the power to
I hereby declare that rights under contract or law have been conveyed to an above with regard to the invention entitled: IDENTIFYING DATA IN A DATA	d remaln with the small business concern identified
by Inventors(s) David A. FARBER and Ronald D. LACHMAN	
described in	
$X_{-}^{[1]} \rightarrow [$] the Specification filed herewith, one $->[X]$ Application No. <u>0.8 /425,160</u> , filed box $->[$] Patent No,	April 11, 1995
box ->[] Patent No, i, If the rights held by the above Identified small business concern are not e	· · · · · · · · · · · · · · · · · · ·
organization having rights to the invention is listed in (A) and (B) below and other than the inventor, who could not qualify under 37 CFR 1.9(c) as an invention, or by any concern which would not qualify as a small busine organization under 37 CFR 1.9(e).	no rights to the invention are held by any person, ndependent inventor if that person had made the ss concern under 37 CFR 1.9(d) or a nonprofit
(A) FULL NAME of assignee/licensee/grantee/conveyee*	
ADDRESS	
	k 4
(B) FULL NAME of assignee/licensee/grantee/conveyee*	
ADDRESS [] INDIVIDUAL [] SMALL BŰŠINESS CO	
*NOTE: <u>Separate verified statement is required</u> from each person, concern or organization name to his/her/its status as a small entity. (37 CFR 1.27)	ned in (A) and (B) above having rights to the invention, averring
Lacknowledge the duty to file, in this case, notification of any change in status resulting in loss of of paying, the earliest of the issue fee or any maintenance fee due after the date on which statu	f entitlement to small entity status prior to paying, or at the time is as a small entity is no longer appropriate. (37 CFR 1.28(b))
I hereby declare that all statements made herein of my own knowledge are true and that all state and further that these statements were made with the knowledge that willful false statements a or both, under section 1001 of Title 18 of the United States Code, and that such willful false stater issuing thereon, or any patent to which this verified statement is directed.	and the like so made are punishable by fine or imprisonment,
NAME OF PERSON SIGNING . Rohald Luchman	
T OF PERSON OTHER THAN OWNER	
MOLTHBROOK IL. 60062	111
SIGNATURE DATE 6-	/4-95



RACK-1002 Page 109 of 351



RACK-1002 Page 110 of 351

PRINT OF DRAWINGS AS ORIGINALLY FILL

- Andrew

ŧ

138 Region ID Pathname True Name Type File ID . Time of last access Time of last modification Safe flag ~ Lock flag Size Owner

FIG. 3

Region status

Policy ' FIG. 5

Mirror processor(s)

Mirror duplication count

	140
True Name	
File ID	
Compressed File ID	
Source IDs	
Dependent processors	
Use count	
Time of last access	
Expiration	
Grooming delete count	
FIG. 4	142
Region ID	· · ·
Region file system	
Region pathname	

<u>GGEEIEG.O40199</u>

₹..~;

PRINT OF DRAWINGS

The state of the s

144

source ID source type source rights source availability source location FIG. 6

146

 Original Name

 Operation

 Type

 Processor ID

 Timestamp

 Pathname

 True Name

 FIG. 7

 148

 date of entry

 type of entry

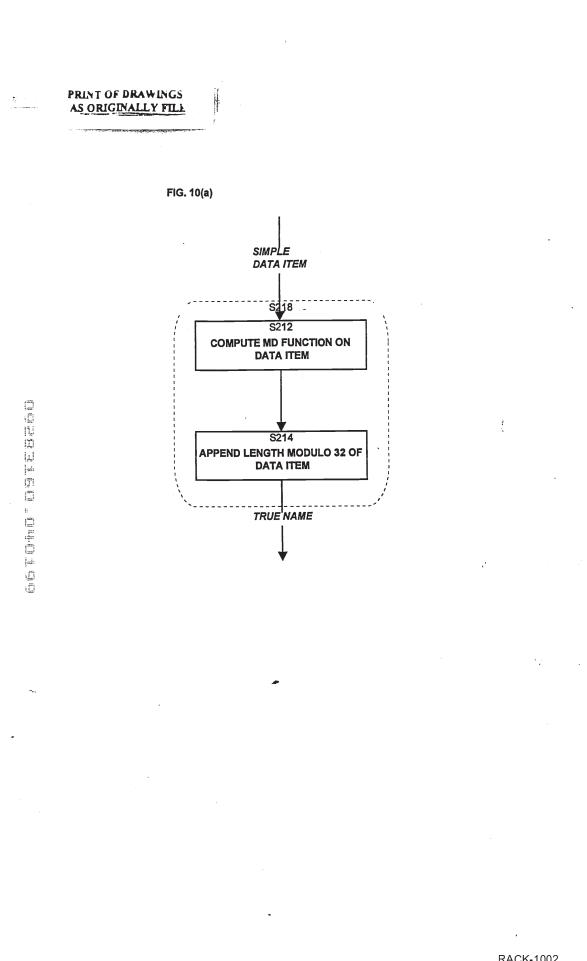
True Name FIG. 8

150

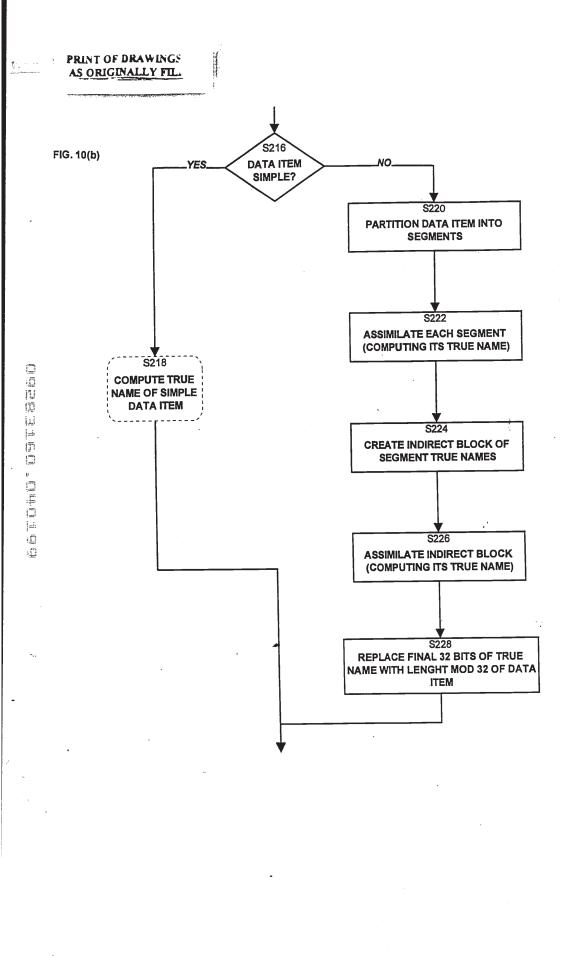
· '

True Name licensee FIG. 9

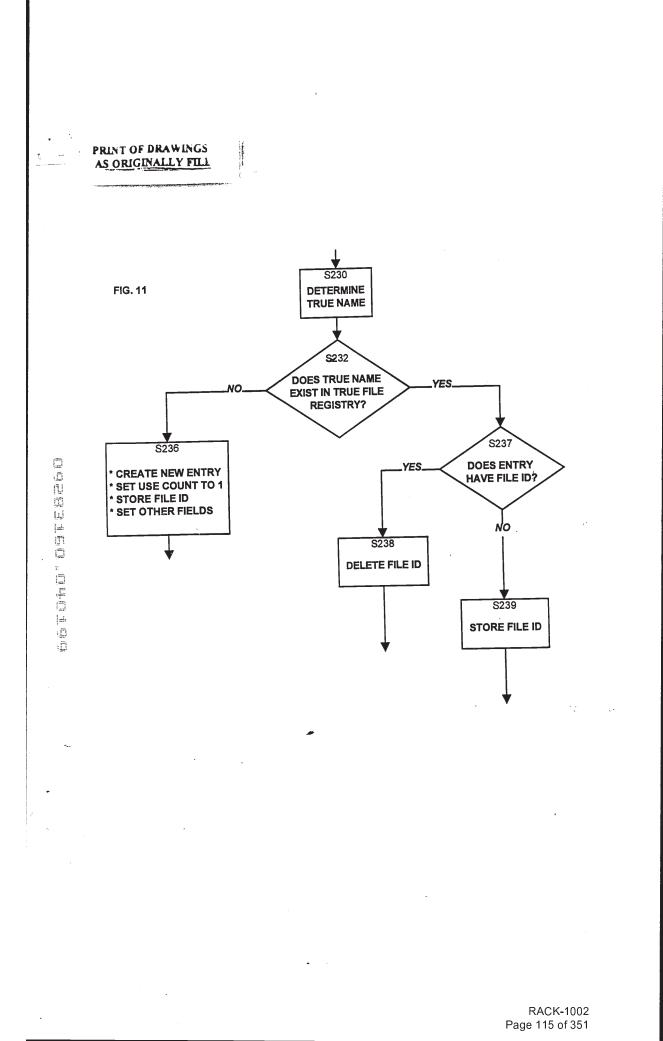
.

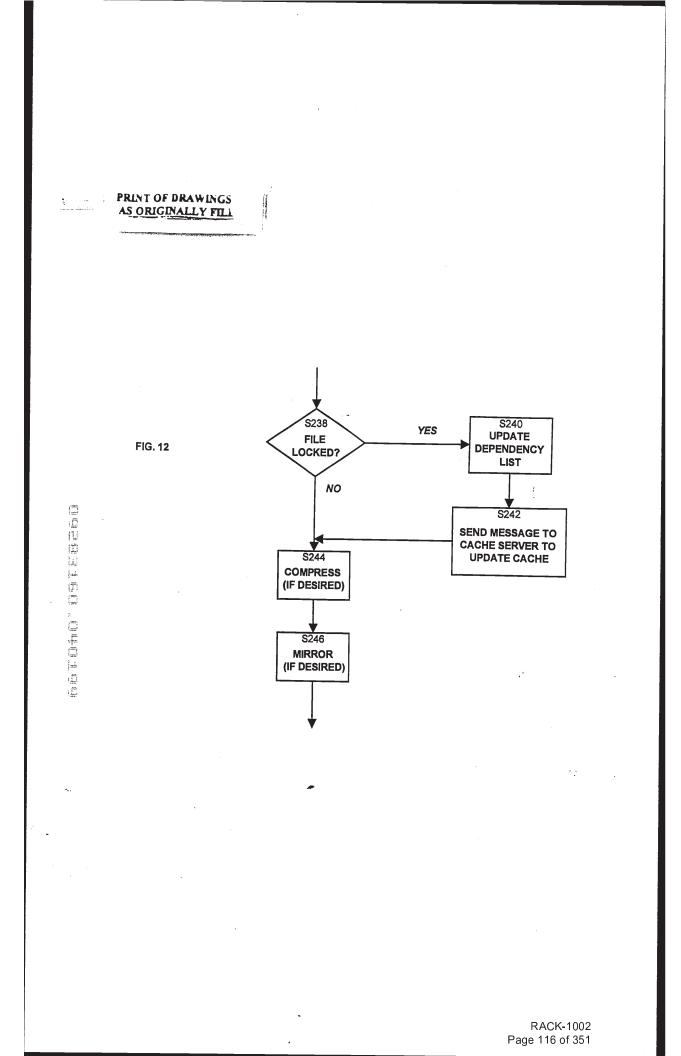


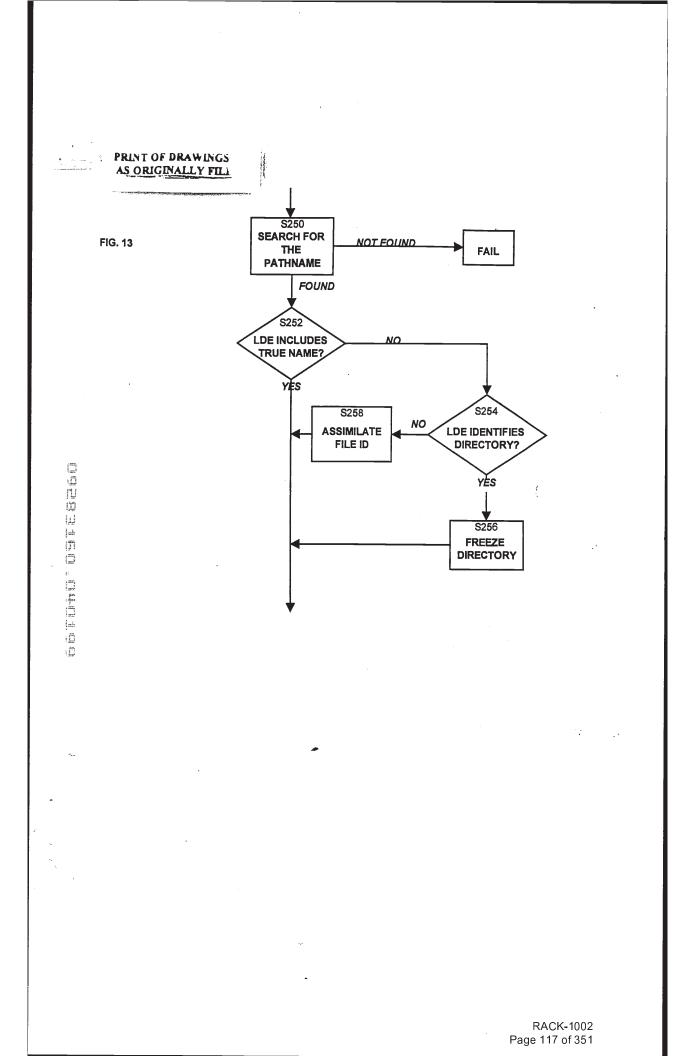
RACK-1002 Page 113 of 351

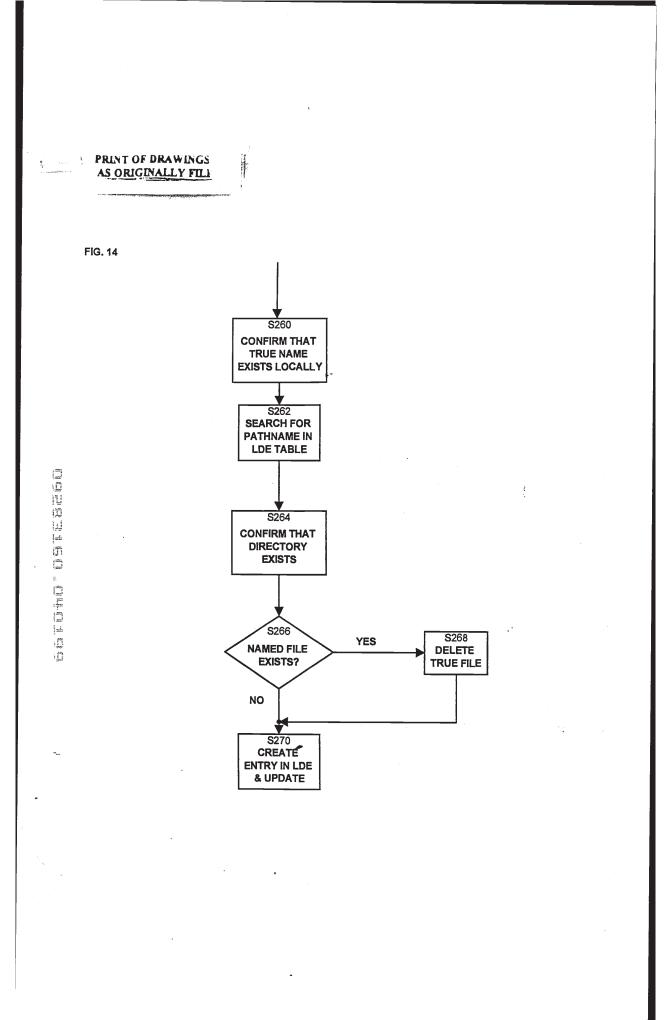


RACK-1002 Page 114 of 351

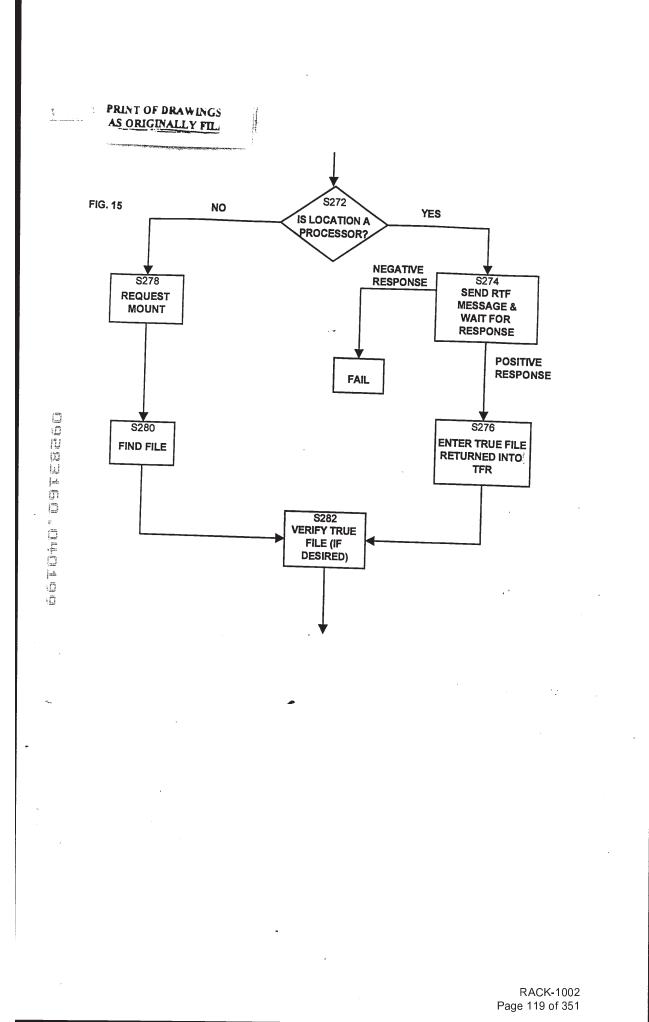


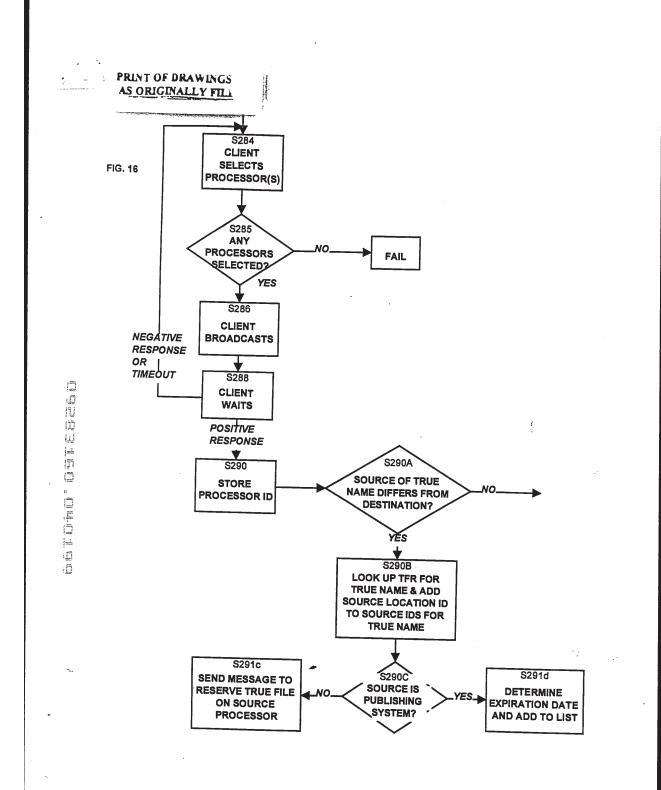




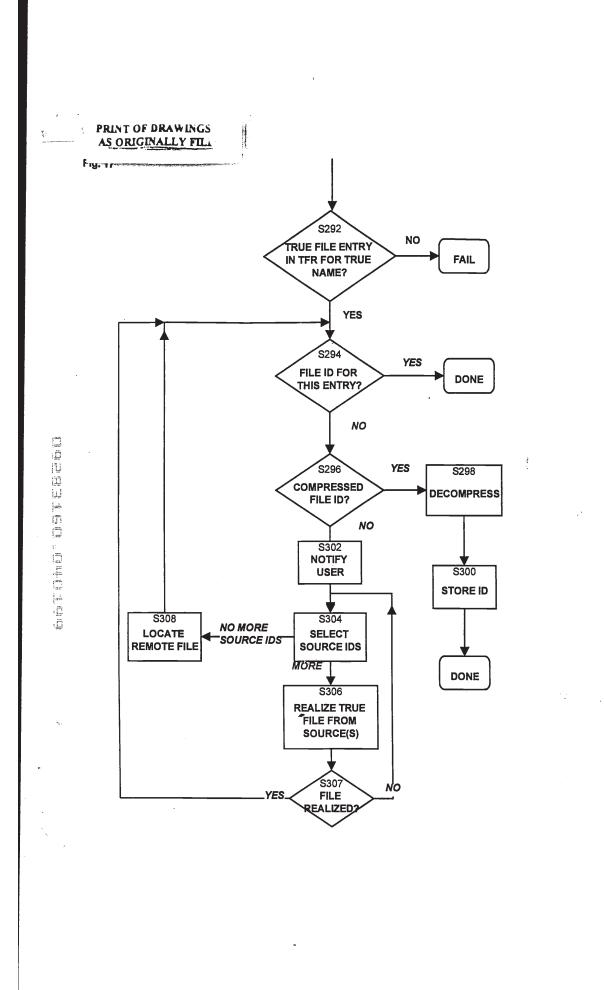


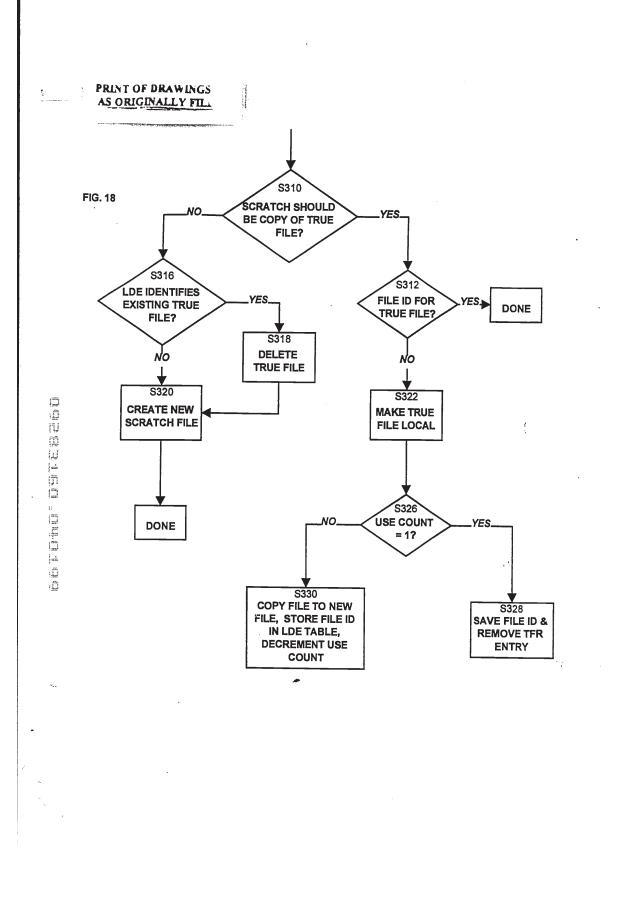
RACK-1002 Page 118 of 351



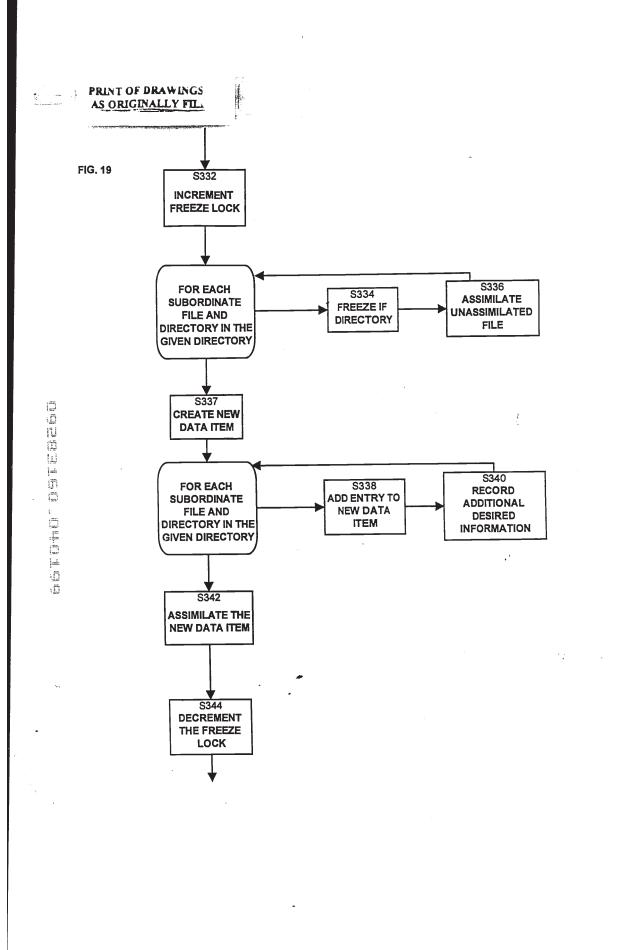


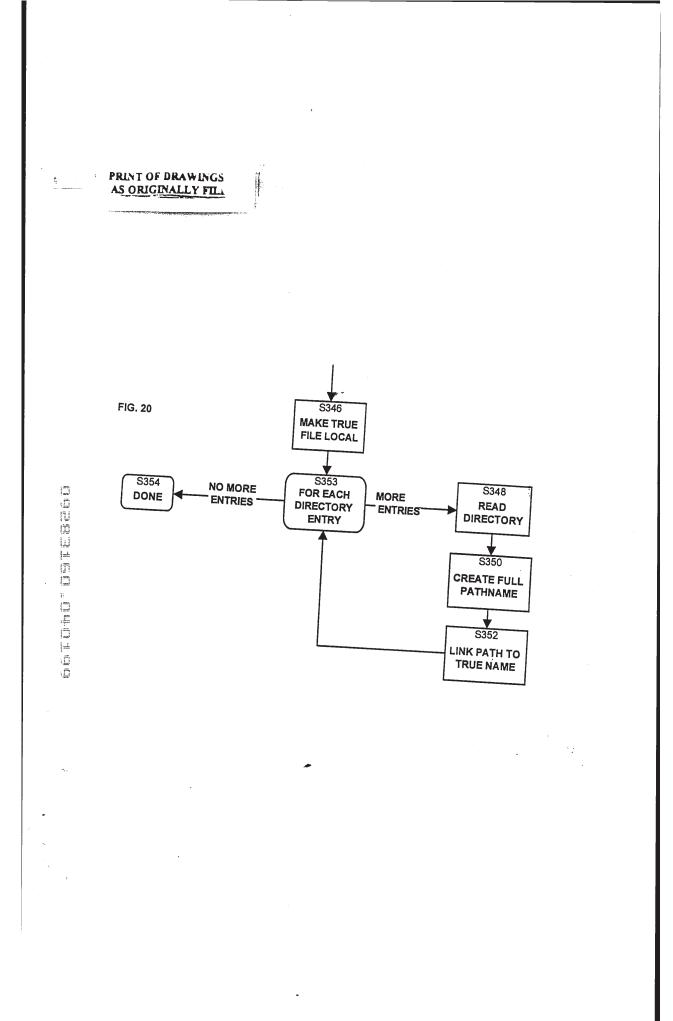
RACK-1002 Page 120 of 351



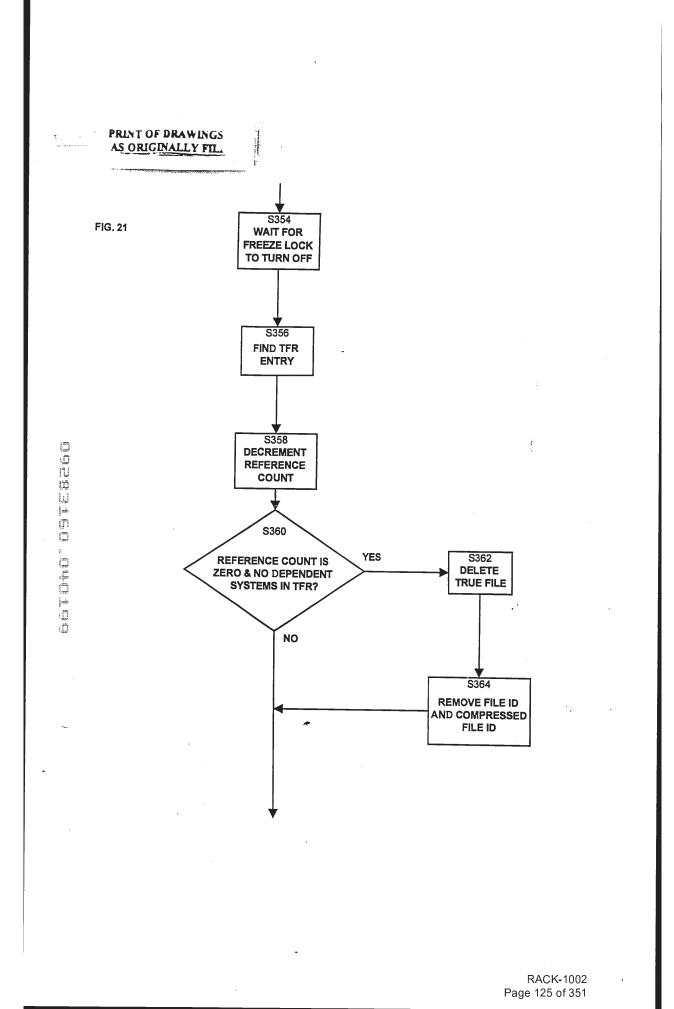


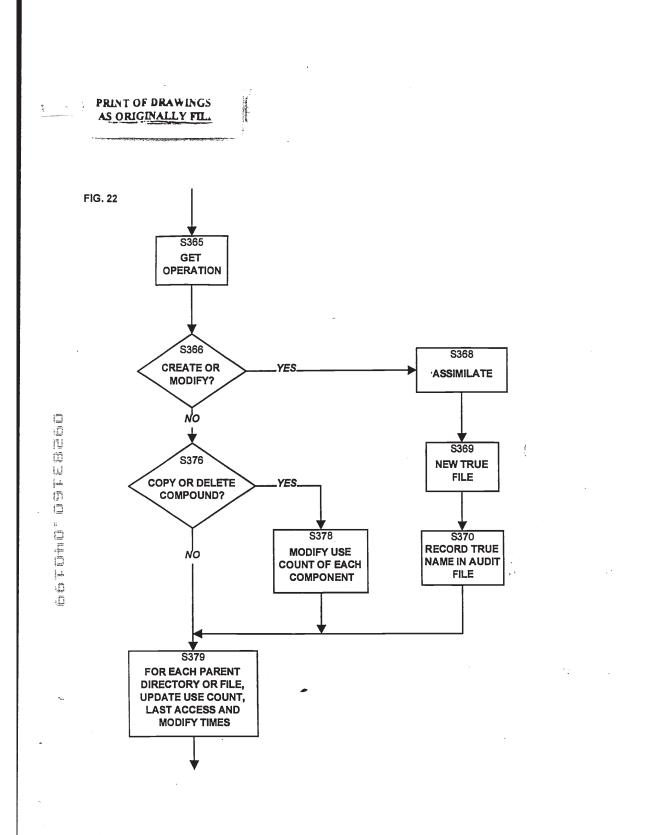
RACK-1002 Page 122 of 351

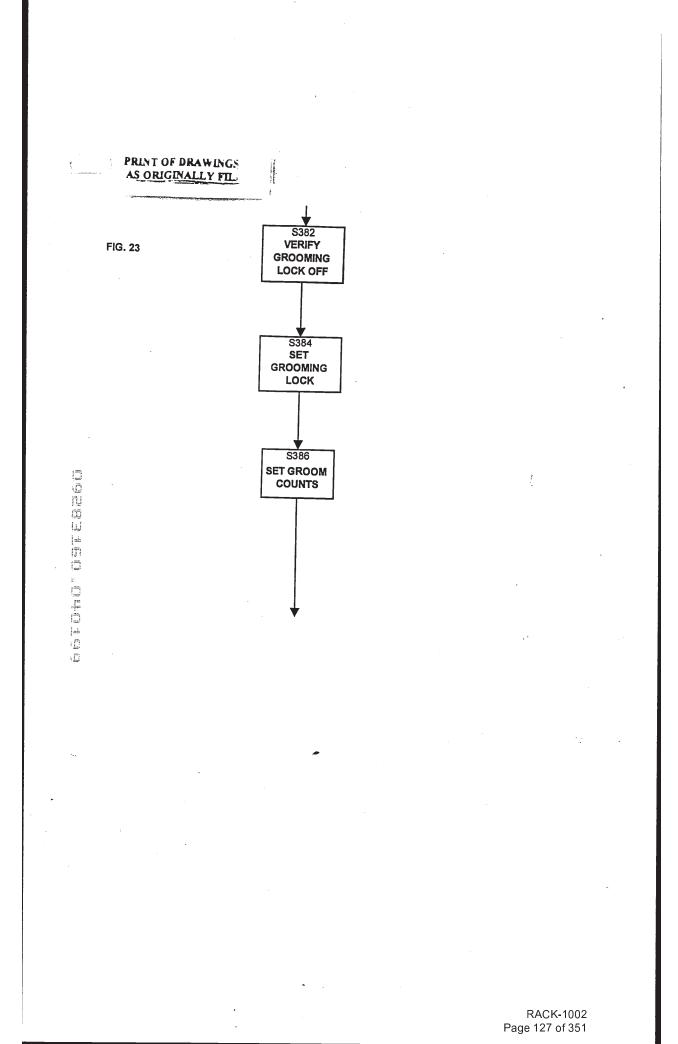


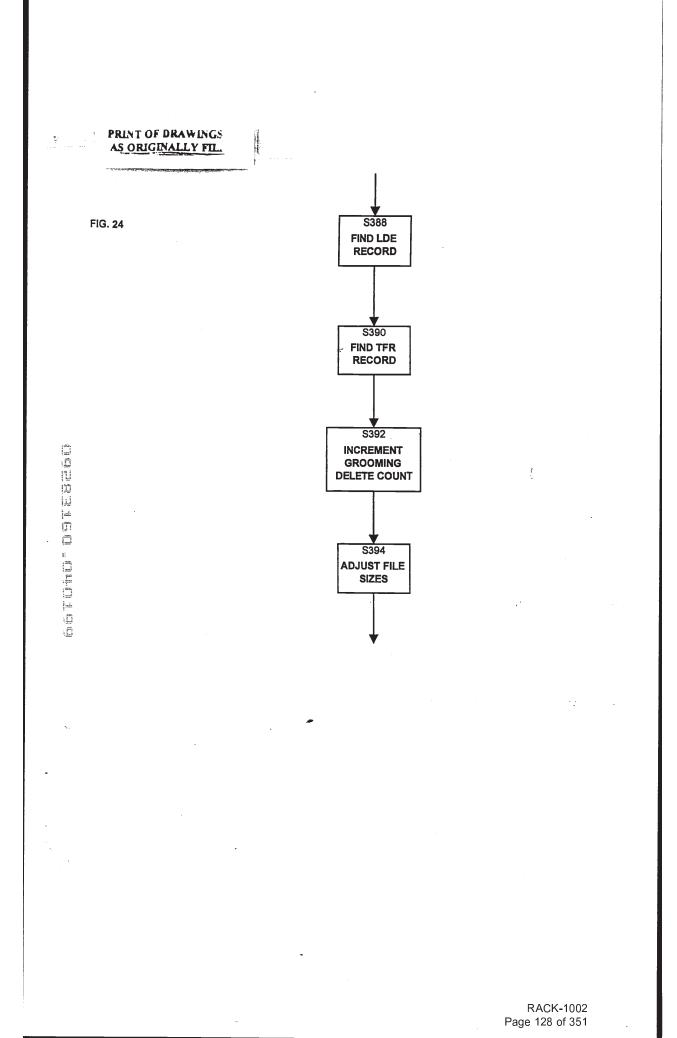


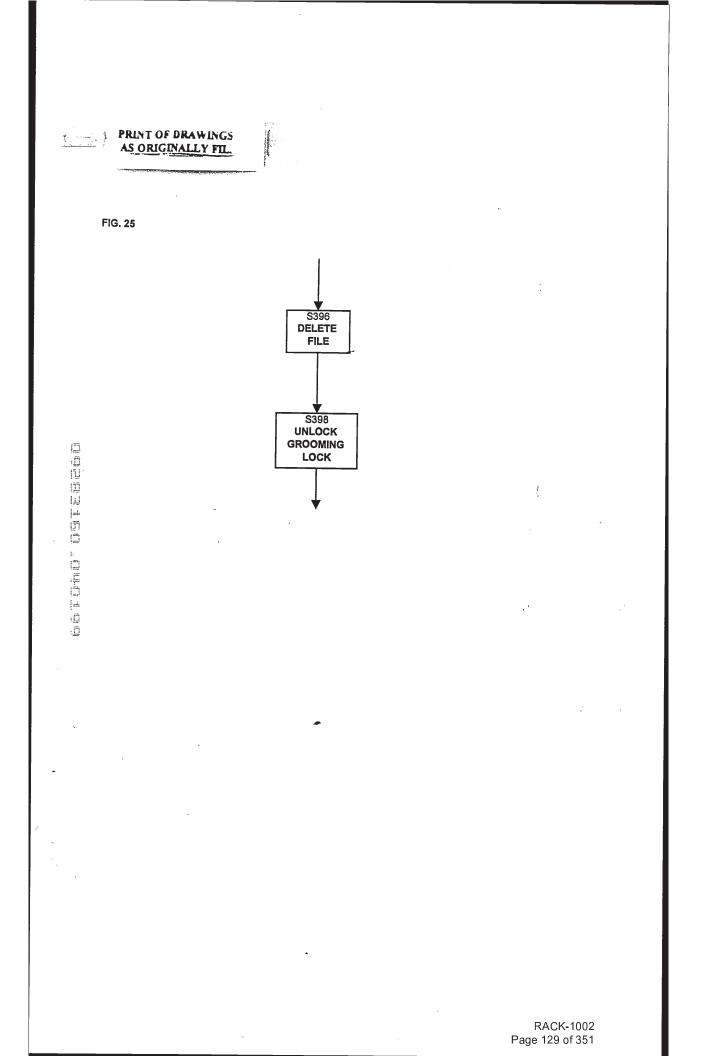
['] RACK-1002 Page 124 of 351

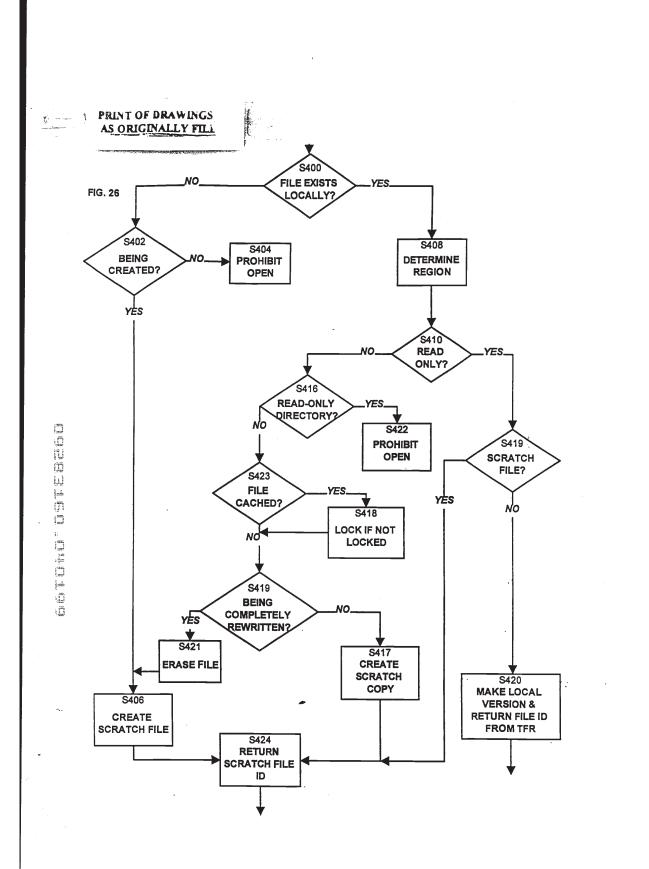




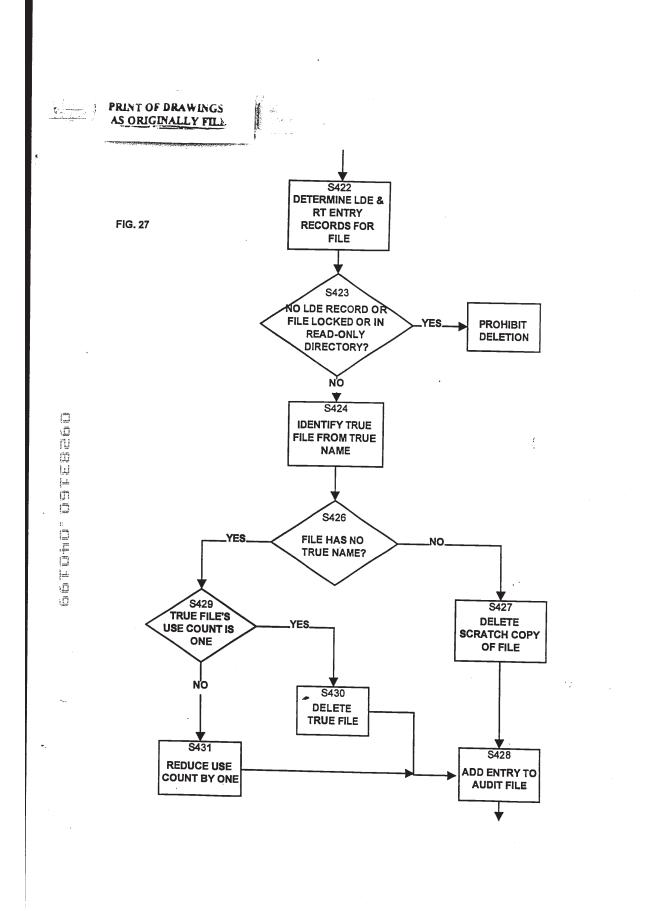




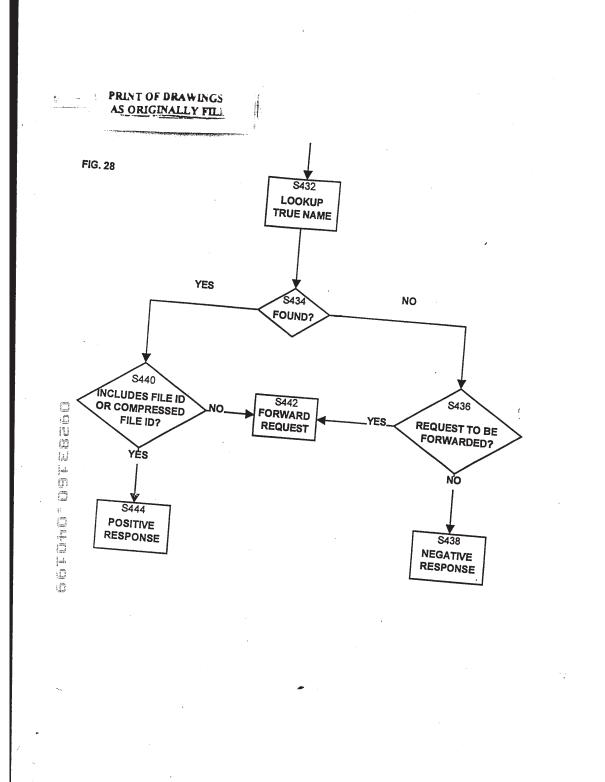




RACK-1002 Page 130 of 351



RACK-1002 Page 131 of 351



RACK-1002 Page 132 of 351

	DATENT A	001			TERMINATI			Application			
	PAIENIA				er 10, 1998			09/283	5 <u>/10</u>	ø	
	an a	CLA		S FILED - olumn 1)		mn 2)	SMALI TYPE		OR	OTHER	
FO	R			R FILED	NUMBER	EXTRA	RATE	FEE		RATE	FEE
BAS	SIC FEE							380.00	OR		760.0
тот	TAL CLAIMS		5	- minus 2	20= * ~	~	X\$ 9=		OR	X\$18=	
INDI	EPENDENT CL	AIMS	Ś	minus	3 = *		X39=		OR	X78=	
MUI	LTIPLE DEPEN	DENT	CLAIM PR	RESENT			+130=	1	OR	+260=	
* If 1	the difference	in colu	umn 1 is	less than ze	ero, enter "0" in a	column 2	TOTAL	. 380 -	OR	TOTAL	
				11/) - PART II			<u> </u>		OTHER	THAN
_		(Col	umn 1)		(Column 2)	(Column 3)	SMAL		OR	SMALL	
NTA		REM A	AIMS IAINING FTER NDMENT		HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA	RATE	ADDI- TIONAL FEE		RATE	ADD TIQN FEE
AMENDMENT	Total	*		Minus	**	=	X\$ 9=		OR	X\$18=	
<u>Z</u>	Independent	*		Minus	***	=	X39=	1	OR	X78=	
ΞI	•								.		
AM	FIRST PRESE	NTATI	ON OF M	ULTIPLE DEI	PENDENT CLAIN		. 100			+260=	
AMI	•	NTATI	ON OF M	ULTIPLE DEF	PENDENT CLAIN		+130=		OR	+260= TOTAL	
AMI	•	NTATI	ON OF M	ULTIPLE DEF			+130= TOTA ADDIT. FE	L			
AMI	•	(Co	ON OF MI	ULTIPLE DEF	(Column 2)	(Column 3)	TOTA	ъ.		TOTAL	
8	•	(Co C REM A	iumn 1)		(Column 2)		TOTA	L		TOTAL	TION
8	•	(Co C REM A	lumn 1) LAIMS MAINING FTER	ULTIPLE DEF	(Column 2) HIGHEST NUMBER PREVIOUSLY	(Column 3) PRESENT	TOTA ADDIT. FE	ADDI- TIONAL FEE		TOTAL ADDIT. FEE	TION
8	FIRST PRESE	(Co C REN A AME *	lumn 1) LAIMS MAINING IFTER NDMENT	Minus Minus	(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR **	(Column 3) PRESENT EXTRA = =	ADDIT. FE	ADDI- TIONAL FEE	OR	TOTAL ADDIT. FEE RATE	TION
	FIRST PRESE	(Co C REN A AME *	lumn 1) LAIMS MAINING IFTER NDMENT	Minus Minus	(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA = =	TOTA ADDIT. FE RATE X\$ 9=	ADDI- TIONAL FEE	OR	TOTAL ADDIT. FEE RATE X\$18=	TION
8	FIRST PRESE	(Co C REN A AME *	lumn 1) LAIMS MAINING IFTER NDMENT	Minus Minus	(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR **	(Column 3) PRESENT EXTRA = =	TOT/ ADDIT. FE RATE X\$ 9= X39= +130= TOT/		OR OR OR	TOTAL ADDIT. FEE RATE X\$18= X78= +260= TOTAL	
8	FIRST PRESE	(Co REM AME * *	lumn 1) Laims Maining Fter Ndment	Minus Minus	(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR ** *** PENDENT CLAIN	(Column 3) PRESENT EXTRA = =	TOT/ ADDIT. FE RATE X\$ 9= X39= +130=		OR OR OR	TOTAL ADDIT. FEE RATE X\$18= X78= +260=	
AMENDMENT B	FIRST PRESE	(Co REM A AME * * NTATI	lumn 1) LAIMS /AINING /FTER NDMENT ON OF M Jumn 1) LAIMS	Minus Minus	(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR *** PENDENT CLAIN (Column 2) HIGHEST	(Column 3) PRESENT EXTRA = = 1 (Column 3)	TOT/ ADDIT. FE RATE X\$ 9= X39= +130= TOT/		OR OR OR	TOTAL ADDIT. FEE RATE X\$18= X78= +260= TOTAL	
C AMENDMENT B	FIRST PRESE	(Co REM AME * * NTATI	lumn 1) LAIMS /AINING FTER NDMENT ON OF M	Minus Minus	(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR ** *** PENDENT CLAIN (Column 2)	(Column 3) PRESENT EXTRA = =	TOT/ ADDIT. FE RATE X\$ 9= X39= +130= TOT/		OR OR OR	TOTAL ADDIT. FEE RATE X\$18= X78= +260= TOTAL	
C AMENDMENT B	FIRST PRESE	(Co REM AME * * NTATI	lumn 1) LAIMS /AINING FTER NDMENT ON OF M Jumn 1) LAIMS VAINING VAINING VAINING	Minus Minus	(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR ** *** PENDENT CLAIN (Column 2) HIGHEST NUMBER PREVIOUSLY	(Column 3) PRESENT EXTRA = = 1 (Column 3) PRESENT	TOTA ADDIT. FE RATE X\$ 9= X39= +130= TOTA ADDIT. FE	ADDI- TIONAL FEE	OR OR OR	TOTAL ADDIT. FEE RATE X\$18= X78= +260= TOTAL ADDIT. FEE	
C AMENDMENT B	FIRST PRESE	(Co REM AME * * * * * (Co C REI A AME * * *	lumn 1) LAIMS /AINING FTER NDMENT ON OF M JUMN OF M JUMN 1) LAIMS VAINING FTER NDMENT	Minus Minus ULTIPLE DE Minus Minus	(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR ** *** PENDENT CLAIN (Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR **	(Column 3) PRESENT EXTRA = =	TOTA ADDIT. FE RATE X\$ 9= X39= +130= TOTA ADDIT. FE	ADDI- TIONAL FEE	OR OR OR OR OR	TOTAL ADDIT. FEE RATE X\$18= X78= +260= TOTAL ADDIT. FEE RATE	
AMENDMENT B	FIRST PRESE	(Co REM AME * * * * * (Co C REI A AME * * *	lumn 1) LAIMS /AINING FTER NDMENT ON OF M JUMN OF M JUMN 1) LAIMS VAINING FTER NDMENT	Minus Minus ULTIPLE DE Minus Minus	(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR ** *** PENDENT CLAIN (Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR **	(Column 3) PRESENT EXTRA = =	TOT/ ADDIT. FE RATE X\$ 9= +130= +130= TOT/ ADDIT. FE RATE X\$ 9= X39=		OR OR OR OR OR	TOTAL ADDIT. FEE RATE X\$18= X78= +260= TOTAL ADDIT. FEE RATE X\$18= X78=	
AMENDMENT C AMENDMENT B	FIRST PRESE	(Co REM AME * * NTATI CC REM AME * * AME	Iumn 1) LAIMS /AINING FTER NDMENT ON OF M	Minus Minus ULTIPLE DE Minus Minus IULTIPLE DE	(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR ** *** PENDENT CLAIN (Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR **	(Column 3) PRESENT EXTRA =	TOT/ ADDIT. FE RATE X\$ 9= X39= +130= TOT/ ADDIT. FE RATE X\$ 9=	ADDI- TIONAL FEE	OR OR OR OR OR	TOTAL ADDIT. FEE X\$18= X78= +260= TOTAL ADDIT. FEE RATE X\$18=	

FORM PTO-875 (Rev. 11/98)

- 8.

Patent and Trademark Office, U.S. DEPARTMENT OF COMMEF * U.S. GP0:1998-454-473/9030 RACK-1002

Page 133 of 351

	M PA	ent of Commerce T-1449) Frademark Office		#	1	1/2		Dkt. No.							
1		N DISCLOSUR	E ST/	ATEM	ENT			Applicant:	FARBER	et al.					
BY APPLI	ICAN	IT						_				F	9	60	
							•		.: R.53(B)		of 08/	960,07	9		
_				·····	1		ז ר		e: April 1, 1				20	V	
Date: April			Page	1	of	3		Examiner	Homere,	J	Grou	o Art U	nit?	926	
	1	OCUMENTS		<u> Albertanten</u>	in Rechter van de	<u>, in an in strady.</u>	<u>.</u>	<u></u>		an a station and	ii Addinate of the		1	<u> </u>	
Examiner's	X	Document		Date		Name					Class	Sub		Filing	
Initials*	\downarrow	Number		MM/Y				of First Inv	entor)			Clas	s	(if approp	p
	AR	4,888,681		12/19		BARNES	6	-	_						
	BR	4,972,367		11/19		BURKE						_			_
		5,050,212		09/19		DYSON									
	-	5,202,982		04/19	_	GRAMLI									
		5,208,858		05/19	_	VOLLER									
	1	5,301,316		05/19		HAMILTO	<u>NC</u>								
		5,343,527		08/19		MOORE						1			
	-	4,571,700	\rightarrow	02/19		EMRY, J									
	1	4,675,810		06/19		GRUNE	<u> </u>							+	
	-	5,050,074		09/19		MARCA						_			_
	<u> </u>	5,276,901		01/19		HOWELL						_		<u> </u>	
	1	5,384,565		01/19	~	CANNON						_			
	+	5,202,982 5,357,623		04/19		GRAMLI						-		-	_
	1.9.4 MAC 788	NT DOCUMENT	L. H. S.	10/19	<u> </u>	MEGOR				And the second state of the		nglish		Translat	
			nourum museuses I	1. San 1. San 1.											
		Document Number	Date MM/Y	YYY	Coun	try	Invent	tor Name	Class	SubC	A	bstract	T	Readily Availabl	le
		1	1	YYY	Coun	try	Invent	tor Name	Class	SubC	lass		No	Readily	le
	OR	1	1	YYY	Coun	try	Invent	tor Name	Class	SubC	lass	bstract	No	Readily Availabl	le
	PR	1	1	YYY 	Coun		Invent	tor Name	Class	SubC	lass	bstract	No	Readily Availabl	le
	PR QR	1	1	YYY 	Coun	try	Invent	tor Name		SubC	lass	bstract	No	Readily Availabl	le
	PR QR RR	1	1	· · · · ·	Coun			tor Name	Class	SubC	lass	bstract	No	Readily Availabl	le
	PR QR RR SR	1	1	· · · · · · · · · · · · · · · · · · ·	Coun			tor Name	Class	SubC	lass	bstract	No	Readily Availabl	le
	PR QR RR SR TR	1	1	· · · · · · · · · · · · · · · · · · ·	Coun			tor Name	Class	SubC	lass	bstract	No	Readily Availabl	le
	PR QR RR SR TR UR	Number	MM/Y								lass	bstract	No	Readily Availabl	le
OTHER (Inc	PR QR RR SR TR UR	Number	MM/Y	Title, P	enodic	al Name,	Date,	Pertinent	Pages, etc		E	bstract	No 	Readily Availabl	le
OTHER (Inc	PR QR RR SR TR UR JUR	Number	MM/Y	l'itle, P Collision	eriodic	al Name,	Date,	Pertinent	Pages, etc) 292-30	lass A E	bstract	No	Readily Availabl	le
OTHER (Inc	PR QR RR SR TR UR UR UR WR	Number g in this order Au Bert dem Boer e Sakti Pramanik e 63-74	thor, 1 t al., C	Fitle, P Collision Multi-D	eriodic ns for irector	al Name, the compr y Hasing,	Date, ession 1993,	Pertinent function of Info. Sys.	Pages, etc of MD ₅ , pp , Vol. 18, N	292-30 10 1, p	lass A E	bstract	No	Readily Availabl	le
OTHER (Inc	PR QR RR SR TR UR UR UR WR	Number g in this order Au Bert dem Boer e Sakti Pramanik o	thor, 1 t al., C t al., I	Fitle, P Collision Multi-D	eriodic ns for irector Hashir	al Name, the compr y Hasing, ng with Di	Date, ession 1993, stribute	Pertinent function of Info. Sys.	Pages, etc of MD ₅ , pp , Vol. 18, N w Space:) 292-30 No 1, pl	lass A E	bstract	No	Readily Availabl	le
OTHER (Inc	PR QR RR SR TR UR UR VR VR XR YR	Number g in this order Au Bert dem Boer e Sakti Pramanik (63-74 Murlidhar Koush Organization wit 299-317 Witold Litwin et a No. HPL-93-21,	MM/Y	Fitle, P Collision Multi-D namic d Inser *-Linea 1993, p	eriodit ns for Hashin H	al Name, the compr ry Hasing, erformanc hing for D	Date, ession 1993, stribute ce, 199 istribut	Pertinent function of Info. Sys. ed Overflo 3, Info. Sy ed Files, H	Pages, etc of MD ₅ , pp , Vol. 18, N w Space: /s., Vol. 18	292-30 No 1, p A File , No. 5, ech. Re	A E C C C C C C C C C C C C C C C C C C	bstract		Readily Availabl	le
OTHER (Inc	PR QR RR SR TR UR UR UR VR WR VR VR ZR	Number g in this order Au Bert dem Boer e Sakti Pramanik e 63-74 Murlidhar Koush Organization wit 299-317 Witold Litwin et a No. HPL-93-21, Yuliang Zheng e	MM/Y	Fitle, P Collision Multi-D namic I d Inser *-Linea 1993, p IAVAL	eriodit ns for tirector Hashir tion P rr Hasi pp. 1-2 - A Or	al Name, the compr ry Hasing, erformand hing for D 2 ne-Way H	Date, ession 1993, stribute ce, 199 istribut	Pertinent function of Info. Sys. ed Overflo 3, Info. Sy ed Files, H	Pages, etc of MD ₅ , pp , Vol. 18, N w Space: /s., Vol. 18	292-30 No 1, p A File , No. 5, ech. Re	A E C C C C C C C C C C C C C C C C C C	bstract		Readily Availabl	le
OTHER (Inc	PR QR RR SR TR UR UR WR WR XR YR ZR AAR	Number g in this order Au Bert dem Boer e Sakti Pramanik of 63-74 Murlidhar Koush Organization wit 299-317 Witold Litwin et a No. HPL-93-21, Yuliang Zheng e of Output (Exten Chris Charnes a	MM/Y	Fitle, P Collision Multi-D d Inser *-Linea 1993, p IAVAL bstract	eriodit ns for Hashin Hashin nr Hasi pp. 1-2 - A Or), pp.	al Name, the compr ry Hasing, erformanc hing for D 2 ne-Way H 83-105	Date, ession 1993, stribute ce, 199 istribut	Pertinent function of Info. Sys. ed Overflo 3, Info. Sy ed Files, H Algorithm	Pages, etc of MD ₅ , pp , Vol. 18, N w Space: 's., Vol. 18 HP Labs Te with Varia	292-30 No 1, p A File , No. 5, ech. Re	A E E C C C C C C C C C C C C C C C C C	bstract		Readily Availabl	le
OTHER (Inc	PR QR RR SR TR UR UR WR WR XR YR ZR AAR	Number g in this order Au Bert dem Boer e Sakti Pramanik (63-74 Murlidhar Koush Organization wit 299-317 Witold Litwin et a No. HPL-93-21, Yuliang Zheng e of Output (Exten	MM/Y	Fitle, P Collision Multi-D d Inser *-Linea 1993, p IAVAL bstract	eriodit ns for Hashin Hashin nr Hasi pp. 1-2 - A Or), pp.	al Name, the compr ry Hasing, erformanc hing for D 2 ne-Way H 83-105	Date, ession 1993, stribute ce, 199 istribute ashing	Pertinent function of Info. Sys. ed Overflo 3, Info. Sy ed Files, H Algorithm	Pages, etc of MD ₆ , pp , Vol. 18, N w Space: rs., Vol. 18 IP Labs Te with Varia	292-30 No 1, p A File , No. 5, ech. Re	Alass A E D4 p. port agth	bstract		Readily Availabl	le

,

,

(PM&S FOR	oartmo M PA1	(modified) ent of Commerce T-1449) Trademark Office						lty. kt. No.				Clien			
			E et/	TEN				pplicant:	252465	-	1				_
BY APPLI			E 317				ſ	ррпсант.							
								ppln. No	.: R.53	(B)(1) D	iv. of 0	3/960,07	9		_
				·	-		1 H	iling Dat							
Date: April 1	Xjuoteritosi:	in the second state of the	Page	2	of	3	<u> </u>	xaminer	Homer	re, J.	Gro	up Art U	nit: 2	776	
	IT DO	OCUMENTS		li l			<u>kod</u> tini								
Examiner's		Document		Date		Name		-			Clas	s Sub Clas	~	Filing	
Initials*	~	Number		1	YYYY	(Family N		First Inv	entor)		_	Cias	5	(if approp	21
	<u> </u>	3,668,647		06/1		EVANGE					_				_
		4,215,402		07/1		MITCHEL					_				_
		4,290,105		09/1		CICHELL	.I				_				_
	_	4,376,299		03/1		RIVEST									_
		4,405,829		09/1		RIVEST						_			_
		4,412,285		10/1		NECHES				_	_				_
		4,414,624		04/1		SUMMEF									-
		4,464,713	\frown	04/1		BENHAS									_
		5,772,933	\rightarrow	03/1		MATICK	5								_
	_	4,642,793		02/1		MEADEN	1								-
		4,691,2999		0911		RIVEST					·				-
		4,725,945	,	02/1		KRONST	ADT							<u> </u>	-
		4,773,039		09/1	\rightarrow	ZAMORA									-
and program that it is a straight to be		NT DOCUMEN	rs									English		Transla	
		Document	Date		Coun	itrx	Invento	r Name	Clas	ss ISu	bClass	Abstract		Readily Availab	,
		Number	MM/Y	YYY										Availab	It
												Enclosed	No	Enclose	•
	OR														
	PR		1				\sum					L			_
	QR														
	RR							<u></u>				L			_
	SR										_	<u> </u>			
	TR		<u> </u>												
	UR		-												_
		g in this order A													
	VR	Witold Litwin et pp. 327-336	al., Lin	ear H	ashing	for Distrib	uted File	es, ACM	ISIGMO	DD, May	/ 1993,				_
	WR	Ming-Ling Lo et Joins, ACM SIC	al., Or MOD,	n Opti pp. 6	mal Pro 9-78, N	ocessor Al /lay 1993	location	to Supp	ort Pipe	elined H	ash				
	XR	Thomas A. Ber 69-81	son, Di	fferen	tial Cry	/ptanalysis	Mod 2 ³	² with Ap	oplicatio	ons to M	D5, pp.				
-	YR	William Perrizo Seventh Hawai													-
	ZR	Vijay Kumar, A Main Memory I	Concu	rrency	/ Contr	ol Mechan	ism Bas	sed on E	xtendib						-
	AAR	Birgit Pfitzman, Computer & Co	Sorting	g Out	Signat	ure Schem				^t Conf.		$\left \right\rangle$	1		-
		1			,			ate Con				<u> </u>	<u> </u>	1	_

RACK-1002PAT-1449 12/98 Page 135 of 351

To: U.S. Do (PM&S FOR	partm M PA	(modified) hent of Commerce (T-1449) Frademark Office	•				Dkt. No.	M# 252465			Client			1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
INFORMA BY APPLI		N DISCLOSUI	RE ST/	TEME	T		Applicant:	FARBER	et al.					-
	0711						Appln. No.:	R.53(B)(1) Div.	of 08/9	960,07	9		-
							Filing Date							
Date: April		NERVICES CONTRACTOR OF STREET	Page	3 0	f 3		Examiner:	Homere, .	l <u>.</u>	Group	o Art U	nit: 2	2776	
	1	OCUMENTS							Anglin - and - and Constant					
Examiner's		Document		Date	Name					Class	Sub	_	Filing	
Initials* t	<u> </u>	Number		MM/YY			of First Inve	ntor)			Clas	s	(if appro	F
		4,887,235		12/1989										_
		4,888,681		12/1989										_
	1	X		12/1984							_			_
	DR	4,972,367		11/1990										_
	+	4,922,414		05/1990										_
	+	5,057,887		10/1991			-				2			_
	+	5,007,658		12/1991		ERT								_
	+	5,025,421	<u> </u>	06/1991										_
		5,129,081	\rightarrow	07/1992										_
		5,129,082	\rightarrow	07/1992										-
		5,144,667		09/1992										-
_		5,179,680		01/1993										_
	1	5,301,286		04/1992									-	-
FOREION	1997 J. Sailt 24	Laph True de la construcción de la	TO	04/130					SAN SKINI TINS	-			Transla	_
FUREIGNE		NT DOCUMEN	Date	10	ountry	line la ca	ntor Name	Class	SubCl	Δ	nglish b stract		Readily	1
		Number	MM/Y		ountry		ntor marrie	Class	Jouber	lass			Availab	le
										E	nclosed	No	Enclose	e
	OR				$\overline{}$			+						-
	PR							+				+		
	QR							1				┢	1	
	RR		+			+		+						
	- Sudin				indiant Mar				1			1-	1	-
OTHER (Ind		[0]]] (a) [4] [5] [0] [4] [4] [4] [4] [4]	NUMBER	Tie Per		e Date	Pertinent P	ages etc	Ŋ	Augusta and a				
OTHER (Ind														
OTHER (Inc		Zhiyu Tian et a pp. 3-13								3				
OTHER (Inc	SR	Zhiyu Tian et a	I., A Ne	w Hashir	ng Function	: Statis	tical Behavio	our and Alg	gorithm	3				_
OTHER (Inc	SR TR	Zhiyu Tian et a pp. 3-13 G. L. Friedman Produced from	I., A Ne , Digital an Ima	w Hashir Camera ge File, I	ng Function with Appar	: Statis	tical Behavio Authenticatio	our and Alg	gorithm	3				
OTHER (Inc	SR TR	Zhiyu Tian et a pp. 3-13 G. L. Friedman Produced from 08/159,980, No	I., A Ne , Digital an Ima ovembei	w Hashir Camera ge File, I 24, 199	ng Function with Appar NASA CASI 3	: Statis atu for a E NO. N	tical Behavio Authenticatic PO-19108-1	our and Alg on of Imag -CU, Seri	gorithm es al No.	3 				_
OTHER (Inc	SR TR	Zhiyu Tian et a pp. 3-13 G. L. Friedman Produced from 08/159,980, No H. Goodman, A	I., A Ne , Digital an Ima ovember	w Hashir Camera ge File, I 24, 199 ect-Orie	ng Function with Appar NASA CASI 3 nted Techn	: Statis ratu for A E NO. N iques, a	tical Behavio Authenticatio PO-19108-1 nd Concurre	our and Alg on of Imag -CU, Serie	gorithm es al No. aching					
OTHER (Inc	SR TR	Zhiyu Tian et a pp. 3-13 G. L. Friedman Produced from 08/159,980, No	I., A Ne , Digital an Ima ovember da, Obj s and Fi	w Hashir Camera ge File, I 24, 199 ect-Orie	ng Function with Appar NASA CASI 3 nted Techn	: Statis ratu for A E NO. N iques, a	tical Behavio Authenticatio PO-19108-1 nd Concurre	our and Alg on of Imag -CU, Serie	gorithm es al No. aching					_
OTHER (Inc	SR TR UR	Zhiyu Tian et a pp. 3-13 G. L. Friedman Produced from 08/159,980, No H. Goodman, A Data Structures A275 385 - 94-	I., A Ne an Ima ovember da, Ob s and Fi 04277	w Hashir Camera ge File, I - 24, 199 ect-Orie le Mana	ng Function with Appar NASA CASI 3 nted Techn gement RE	: Statis ratu for A E NO. N iques, a PORT E	Authenticatic PO-19108-1 nd Concurre DOCUMENT	our and Alg on of Imag -CU, Seri oncy in Tea ATION PA	gorithm es al No. aching AGE AD					
OTHER (Inc	SR TR UR	Zhiyu Tian et a pp. 3-13 G. L. Friedman Produced from 08/159,980, No H. Goodman, A Data Structures A275 385 - 94- Advances in Ci Application of C	I., A Ne an Ima ovember da, Ob s and Fi 04277 ryptolog	w Hashir Camera ge File, I - 24, 199 lect-Orie le Mana y-EURO	ng Function with Appar NASA CASI 3 nted Techn gement RE CRYPT'93,	: Statis ratu for A E NO. N iques, a PORT E Worksl	tical Behavio Authenticatic PO-19108-1 Ind Concurre DOCUMENT,	on of Imag -CU, Seri -CU, Seri ATION PA heory and	gorithm es al No. aching AGE AD					_
OTHER (Inc	SR TR UR VR	Zhiyu Tian et a pp. 3-13 G. L. Friedman Produced from 08/159,980, No H. Goodman, A Data Structures A275 385 - 94- Advances in Ci Application of C Proceedings	I., A Ne an Imagovernber Ada, Obj s and Fi 04277 ryptolog Cryptogr	w Hashir Camera ge File, I 24, 199 ect-Orie le Mana y-EURO aphic Te	ng Function with Appar NASA CASI 3 nted Techn gement RE CRYPT'93, echniques L	: Statis ratu for , E NO. N iques, a PORT [Works] ofthus,	Authenticatic PO-19108-1 Ind Concurre DOCUMENT, hop on the Ti Norway, May	on of Imag -CU, Seri- incy in Tea ATION PA heory and y 23-27, 1	es al No. aching GE AD)_				
OTHER (Inc	SR TR UR VR	Zhiyu Tian et a pp. 3-13 G. L. Friedman Produced from 08/159,980, No H. Goodman, A Data Structures A275 385 - 94- Advances in Ci Application of C Proceedings Proceedings of	I., A Net an Ima ovember Ada, Ob s and Fi 04277 ryptolog Cryptogr the 199	w Hashir Camera ge File, I 24, 199 iect-Orie le Manag y-EURO aphic Te 33 ACM	ig Function with Appar NASA CASI 3 nted Techn gement RE CRYPT'93, echniques L SIGMOD In	: Statis ratu for , E NO. N iques, a PORT [Works] ofthus,	Authenticatic PO-19108-1 Ind Concurre DOCUMENT, hop on the Ti Norway, May	on of Imag -CU, Seri- incy in Tea ATION PA heory and y 23-27, 1	es al No. aching GE AD)_				
OTHER (Inc	SR TR UR VR	Zhiyu Tian et a pp. 3-13 G. L. Friedman Produced from 08/159,980, No H. Goodman, A Data Structures A275 385 - 94- Advances in Ci Application of C Proceedings Proceedings of of Data, Vol. 22 Advances in Ci	I., A Ner , Digital an Ima ovember Ada, Ob s and Fi 04277 ryptolog Cryptogr the 198 2, Issue ryptolog	w Hashir Camera ge File, I 24, 199 ect-Orie le Manage y-EURO aphic Te 93 ACM 2, June y-AUSC	ig Function with Appar NASA CASI inted Techn gement RE CRYPT'93, echniques L SIGMOD In 1993 RYPT '92 -	: Statis atu for A E NO. N iques, a PORT E Worksh ofthus, ternatio	Authenticatic Authenticatic PO-19108-1 nd Concurre DOCUMENT, nop on the T Norway, May nal Conferer op on the Th	no of Imag -CU, Seri -CU, Seri ATION PA heory and y 23-27, 1 nce on Ma neory and	es al No. aching GE AD 993 nagem	ent				_
OTHER (Inc	SR TR UR VR	Zhiyu Tian et a pp. 3-13 G. L. Friedman Produced from 08/159,980, No H. Goodman, A Data Structures A275 385 - 94- Advances in Ci Application of C Proceedings Proceedings of of Data, Vol. 22	I., A Ner , Digital an Ima ovember Ada, Ob s and Fi 04277 ryptolog Cryptogr the 198 2, Issue ryptolog ic Techn	w Hashir Camera ge File, I 24, 199 ect-Orie le Manage y-EURO aphic Te 93 ACM 2, June y-AUSC	ig Function with Appar NASA CASI inted Techn gement RE CRYPT'93, echniques L SIGMOD In 1993 RYPT '92 -	: Statis atu for A E NO. N iques, a PORT E Worksh ofthus, ternatio	Authenticatic Authenticatic PO-19108-1 nd Concurre DOCUMENT, nop on the T Norway, May nal Conferer op on the Th	no of Imag -CU, Seri -CU, Seri ATION PA heory and y 23-27, 1 nce on Ma neory and	es al No. aching GE AD 993 nagem	ent				
OTHER (Inc	SR TR UR VR	Zhiyu Tian et a pp. 3-13 G. L. Friedman Produced from 08/159,980, No H. Goodman, A Data Structures A275 385 - 94- Advances in Ci Application of C Proceedings Proceedings of of Data, Vol. 22 Advances in Ci of Cryptograph	I., A Ner , Digital an Ima ovember Ada, Ob s and Fi 04277 ryptolog Cryptogr the 198 2, Issue ryptolog ic Techn	w Hashir Camera ge File, I 24, 199 ect-Orie le Manage y-EURO aphic Te 93 ACM 2, June y-AUSC	ig Function with Appar NASA CASI inted Techn gement RE CRYPT'93, echniques L SIGMOD In 1993 RYPT '92 -	: Statis atu for A E NO. N iques, a PORT E Worksh ofthus, ternatio	Authenticatic Authenticatic PO-19108-1 nd Concurre DOCUMENT, nop on the T Norway, May nal Conferer op on the Th	n of Imag -CU, Seri Nov in Tea ATION PA heory and y 23-27, 1 nce on Ma heory and lia, Decen	es al No. aching GE AD 993 nagem	ent				

	IN THE UNITED STATES PATEI REQUEST F (RULE 5)	OR FILING
II/99 Inven Parer This C	t Appln. No.: 08 960,079 Series Code û û Serial No. t Filed: October 24, 1997 Case Filed: April , 1999	(DO NOT USE FOR CIPs)
Title: Asst. Wash Sir: 1. 1A. (1) (2)	IDENTIFYING DATA IN A DATA PROCESSING SYS Commissioner of Patents ington, DC 20231 To effect the above-requested filing today: Attached is a copy (which must be filed) of this appl Abstract Abstract Specification and claims (94 pages) (must be attached if originally filed): Always X one box, only: Signed declaration or oath as originally filed in pr NO declaration or fee is enclosed; therefore, this	Date: April 1, 1999 (Parent Matter No. 243063 (Parent Matter No. 243063) ilication, including: tached) : : 24 sheet(s)/set: 1 set informal; □ Formal of size □ A4 □ 11"
2.	hereby made requesting deletion as inventor(s invention being claimed in this application: 1	of the inventors named in the prior application. Petition is) of the following who is/are not inventor(s) of the 2. 4. 6. 8. 10. ered as being part of the disclosure of the accompanying rence thereto.

.

PAT-108 1298 RACK-1002 Page 137 of 351

A

	Second and an and a second second							
	4.	Priority is claimed under 35 L	USC 119/365 base					Page 2 of 4
		Application No.	Filing Date	a on may		ountry) lo.	Of Filing (- 14
		(1) (2) (3)		(4) (5)			<u>Filing [</u>	
r I		a 🗌 👘 👘		(6)				
	r	U.S. Application N	s previously filed on	- <u>-</u>	, filed on	i	in _:	
	4.	c. Certified copy/copies	S filed during Internation	ional stars		1	·	
		(a) Domestic priority is claimed (b) Benefit is claimed of Provi	d from PCT/ /isional Application N	_/, lo. 60/	filed , filed	d t	<u>.</u>	
	5.	Prior application is assigned to	to <u>kiNETech, Inc.</u>					
	6.	by assignment recorded <u>June 23,</u>	(Date)		7593		0036.	
		Attached is the following numb <u>different</u> assignors): <u>1</u> a	her of Assignmente (including c Cover She	original and all I ets. (Do <u>NOT</u> f	ater succ ile old co	essive ones: ver sheets.)	; by
		(Assignments in parent <u>must t</u> want it/them recorded against	he refiled with now (0		•		
2 4 34		Please return the recorded Assignmen	and continuing applic	callon.)				
	ա 447. ՄԴ	The power of attorney in the p			<u>azar, Reg. No.</u>	<u>28.872</u>		
		(Name and Reg. No.) whose current address is as in item 8						
		a. 🛛 Recognize as associate attorn		eg. No. 37,	,497			
) – É	(Name, Reg. No. and Address)						
	₩ 8 .	Address all future communications of Pillsbury Madison & Sutro LLP, N Washington, D.C. 20005-3918	to Intellectual Prop linth Floor, East To	erty Grou wer 1100	⊧p New York A∨e	nue, N.V	V.,	
	9	Amend the specification by inserti	ting before the first lir	ne the sen	tence:-This is	 a		
A1	/	which is a continuation of 08/425,160, f	pplication No. 08/96	60,079,	filed	<u>Octobe</u>	er 24. 1997,	Patent No
¢	9.	(a) Amend the specification by ins Provisional Application No. 60/_	serting before the fire	rot line: Th		1		5,978,7 10/24/1
	10.	 It has been recently determined th Hence: 	, med					Im
3 3 3		Nence: (No.) Verified Statement(s) e ⊠ filed in above prior applicatio ⊠ attached.	establishing "small o			9 & 27 we	ere/are:	
	11. (<u>one</u> box) (<u>must</u> be) (X'd)		prior application (Use	e Form PA	AT-111).			
			7				DAT	A
				,			PAT-1	108 12/98

.

PAT-108 12/98 RACK-1002 Page 138 of 351

Page 3 of 4

- 12. INFORMATION DISCLOSURE STATEMENT: Attached is Form PTO-1449 listing all of the documents cited by Applicant and the PTO in the parent application(s) relied upon under 35 USC 120 and referenced in item 9 above. Per Rule 98(d) copies of those documents are <u>not required</u> now. Please consider those documents and <u>advise</u> that they have been considered in <u>this new</u> application as by returning a copy of the enclosed Form PTO-1449 with the Examiner's initials in the left column per MPEP 609.
- 13. Attached is a Rule 103(a) Petition to Suspend Action.
- 14. X PRELIMINARY AMENDMENT to be entered before fee calculation: (Do not make amendments here except for correction of improper multiple dependencies or cancellation of whole claims or multiple dependencies for purpose of reducing the filing fee per MPEP §§ 506 and 607; do not cancel all claims).

Please cancel claims 1-45 and 50-53 without prejudice. The remaining claims correspond to non-elected Groups III & IV from the Examiner's Restriction Requirement of June 4, 1996.

	->->-> <u>CLAIMS AS</u>					<u> </u>	
NOTE:	If box 1A <u>2</u> is X'd, do but leave lines 15-2		ank.			·	
			<u>ar 111.</u>				
					Large/Small	1	Fee
ų.					Entity		Cod
15. Bas	ic Filing Fee		Desi	gn Application	\$310/\$155		106/26
	ic Filing Fee			gn Application	\$760/\$380	+380	101/20
	al Effective Claims	5	minus 20 =	0	x \$18/\$9	+0	103/203
P	pendent Claims	3	minus 3 =	0	x \$78/\$39	+0	102/20
	<u>iy proper</u> multiple depe	endent claim (ig	nore improper)	is present,	\$260/\$130	+0	104/20-
20.					Subtotal =	\$380	
	etition" box 13 above is			\$130		+0	122
	box 6 above is X'd, add	Assignment re	ecording fee	\$40		+40	581
, ²							
22.			τοτ	AL FILING FE	E ATTACHED =	\$420	
					(carr	y forward to	Item 31)
23. [ATTACHED:						

PAT-108 12/98 RACK-1002 Page 139 of 351

4

Page 4 of 4

ADDITIONAL FEE CALCULATION FOR PRELIMINARY AMENDMENT <u>PER BOXES 24/25</u>

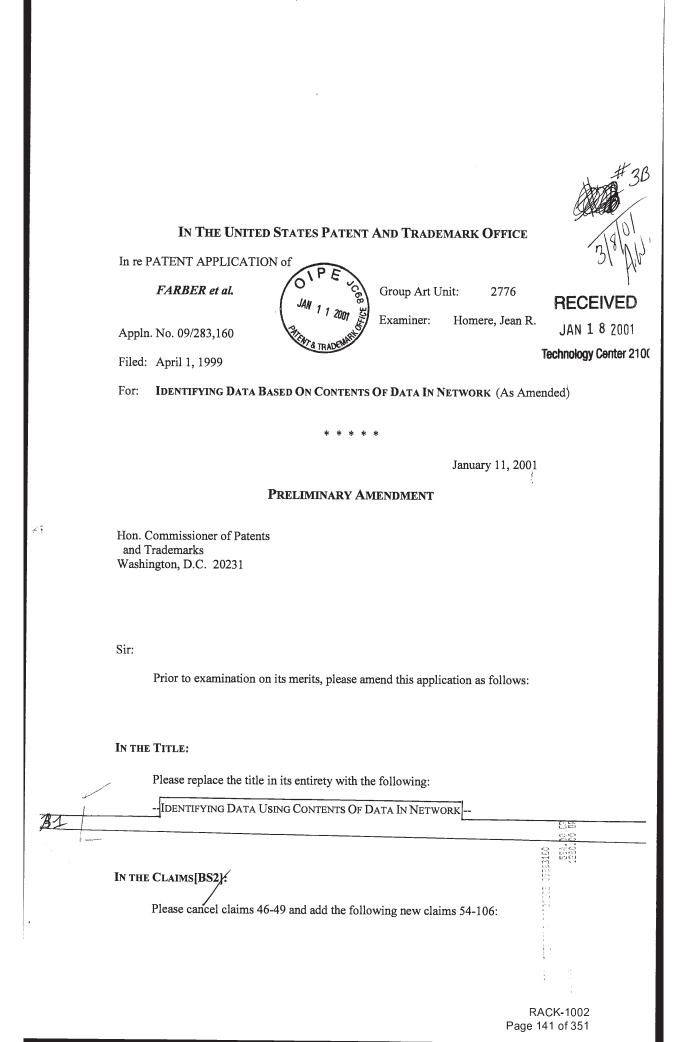
	re af	aims maining ter nendmei	r F	Highest number previous paid for	ly	Present Extra						Additional Fee			
								L	arge/Smal	<u>l Entity</u>			File Code		
27.	Total Effective Cla	aims _*	m	inus **	20	. =	0	x	\$18/\$9	=	\$_	0	(103/203)		
28.	Independent Clai	ims <u>*</u>	m	inus ***	3	_ =	0	x	\$78/\$39	=	+_	0	(102/202)		
29.	If amendment ent <u>first time</u> , add (pe										+_	0	(104/204)		
30.								A		L FEE	\$ <u>_</u>	0			
31.					<u>plus</u> F	EE fro	m item :	22 (on page 3		+_	420			
☐ 32.							TOTAL	FE	E ATTACH	IED	\$_	420			
Ш ДЗЗ.	*If the entry in this space is	s less than the	entry in the ne	ext space, the	"Present I	Extra" res	ult is "0"								
ايدا ⊒≟ 34.	**If the "Highest number pr	eviously paid	for" (see item 1	7 above) is le	ess than 20), write "2	0" in this spa	ace							
⊡ 35 .	If the "Highest number prev	viously paid fo	r" (see item 18	above) is less	s than 3, w	rite "3" in	this space								
filed, of the second se	RGE STATEMENT: rized to charge any or which should hav 16-18 (<u>missing o</u> nent under Rule 20, se a <u>duplicate</u> copy CHARGE STATEMI d.	fee speci e been fil <u>r_insuffici</u> , or credit of this sh	fically auth ed herewit ent fee or any overp eet is attac	norized h th or cond <u>nly</u>) now payment, ched.	ereafte cerning or her to our	r, or a any p eafter Accou	ny missi aper file relative ınt/Orde	ng dh to rN	or insufficie ereafter, ar this applic os. shown	ent fee(s) f nd which r ation and in the hea	filed, nay I the ading	or assert be require resulting hereof f	ed to be ed under g Official or which		
			Pillsbury Intellectu				-P	/							
	New York Avenue, Floor, East Tower		By Atty:	Dale	s. Lazz	\$ <u>}</u>		<u> </u>	hi-	Reg. No	2	8872			
Wash Tel: (2 DSL/E	ington, D.C. 2000 202) 861-3000 3S:kim ./Sec.		Sig:	///	M	1/	AJ.	1/		Fax: Tel:	•	02) 822-0 02) 861-3			
, ,	NOTE No. 1: NOTE No. 2:														

RACK-1002 Page 140 of 351

A

26.

27



Application of ber et al, No. 09/283,160

SUBCI

ji Ti

--54. (New) In a system in which a set of data items are distributed across a network of servers, at least some of the data items being cached versions of data items from a source server, a content delivery method comprising:

determining a data identifier for a particular data item, the data identifier being determined using a given function of the data comprising the particular data item; and

responsive to a request for the particular data item, the request including at least the data identifier of the particular data item, providing the particular data item from a given one of the servers of the network of servers.

55. (New) In a system in which a set of data items are distributed across a network of servers, some of the data items being cached from a source server, a content delivery method comprising:

determining a data identifier for a particular data item, the data identifier being determined using a given function of the data comprising the particular data item; and

responsive to a request for the particular data item, the request including at least the data identifier of the particular data item, causing a copy of the particular data item to be provided from a given one of the servers of the network of servers.

56. (New) A content delivery method, comprising:

distributing a set of data items across a network of servers

determining a data identifier for a particular data item, the data identifier being determined using a given function of the data comprising the particular data item; and

> RACK-1002 Page 142 of 351

Application of ber et al, No. 09/283,160

e g

in response to a request for the particular data item, the request including at least the data identifier of the particular data item, providing the particular data item from a given one of the servers of the network of servers.

57. (New) A method as in claim 56 further comprising: determining whether the data identifier corresponds to a data identifier of any data item present on the given server.

58. (New) A method as in claim 57 further comprising:
based on said determining, if the data identifier does not correspond to a data item present on the given server, locating the particular data item from another server.

59. (New) A method as in claim 58 further comprising:obtaining, on the given server, a local copy of the particular data item,from the other server.

60. (New) A method as in claim 56 wherein at least some of the data items distributed across the network of servers are cached versions of data items from another server.

61. (New) A method as in claim 56 further comprising:

resolving the request for the particular data item based on a measure of availability of at least one of the servers.

3

Application of l ber et al, No. 09/283,160

62. (New) A method as in claim 61 wherein the measure of availability comprises one or more of:

(a) a measurement of bandwidth to the server;

(b) a measurement of a cost of a connection to the server, and

(c) \setminus a measurement of a reliability of a connection to the server.

63. (New) A method as in claim 56 wherein the data item is a compound data item made up of various component data items, the method further comprising:

for each component data item of at least some of the component data

items:

(a) determining a data identifier for the component data item,
 the data identifier determined using the given function of the data
 comprising the component data item; and

(b) providing the component data item from a given one of the servers of the network of servers.

64. (New) A content delivery method, comprising:

distributing a set of data items across a network of servers;

for a particular data item having a particular name specifying a location in the network at which the data item may be located, determining another name for the particular data item, the other name including a data identifier determined using a given function of the data comprising the particular data item; and

Application of l ber et al, No. 09/283,160

in response to a request for the particular data item, the request including the other name of the particular data item, providing the particular data item from a given one of the servers of the network of servers.

65. (New) A method as in claim 64 wherein at least some of the data items are cached versions of data items from another server.

66. (New) \setminus A method as in claim 64 further comprising:

resolving the request for the particular data item based on a measure of availability of at least one of the servers.

67. (New) A method as in claim 66 wherein the measure of availability comprises one or more of:

(a) a measurement of bandwidth to the server;

(b) a measurement of a cost of a connection to the server, and

(c) a measurement of a reliability of a connection to the server.

68. (New) A method as in claim 64 wherein the particular data item is a compound data item comprising various component data items, the method further comprising:

for at least one component data item:

(a) determining a data identifier for the component data item,
 the data identifier determined using a given function of the data
 comprising the component data item; and

Application of i ber et al, No. 09/283,160

(b) providing the component data item from a given one of the servers of the network of servers.

69. (New) A content delivery method, comprising: distributing a set of data items across a network of servers, at least some of the data items being cached versions of data items from another server; determining a data identifier for a particular data item, the data identifier determined using a given function of the data comprising the particular data item; and

in response to a request for the particular data item, the request including at least the data identifier of the particular data item, providing the particular data item from a given one of the servers of the network of servers.

70. (New) A content delivery method, comprising:

causing a set of data items to be distributed across a network of servers, at least some of the data items being cached versions of data items from another server;

determining a data identifier for a particular data item, the data identifier determined using a given function of the data comprising the particular data item; and

in response to a request for the particular data item, the request including at least the data identifier of the particular data item, causing the particular data item to be provided from a given one of the servers of the network of servers.

71. (New) A content delivery method, comprising:

6

RACK-1002 Page 146 of 351

Application of ber et al, No. 09/283,160

distributing a set of data items across a network of servers, the network of servers being organized into a set of regions;

determining a data identifier for a particular data item, the data identifier determined using a given function of the data comprising the data item;

in response to a client request for the particular data item, the request including at least the data identifier of the particular data item, providing the client with the particular data item from a given one of the servers of the network of servers within the region.

72. (New) A method as in claim 54 wherein the given function is a message digest function or a hash function.

73. (New) A method as in claim 72 wherein the given function is selected from the functions: MD4, MD5, and SHA.

74. (New) A method as in claim 4 wherein the given function randomly distributes its outputs.

75. (New) A method as in claim 54 wherein, for a particular data item, the given function produces a substantially unique value based on the data comprising the data item.

76. (New) A method as in claim 54 wherein a data item may comprise a file, a portion of a file, a page in memory, a digital message, a digital image, a video signal or an audio signal.

Application of I Jer et al, No. 09/283,160

. 19. 19. 77. (New) In a system in which a set of data items are distributed across a network of servers, at least some of the data items being cached versions of data items from a source server, a content delivery method comprising:

Responsive to a request for a particular data item, the request including at least a data identifier of the particular data item, wherein the data identifier is determined by applying a message digest function MD5 to the data comprising the particular data item, providing the particular data item from a given one of the servers of the network of servers,

wherein a data item may be a file, a portion of a file, a page in memory, a digital message, a digital image, a video signal or an audio signal.

78. (New) A content delivery method, comprising:

distributing a set of data items across a network of servers, at least some of the data items being cached versions of data items from another server;

determining a data identifier for a particular data item, the data identifier determined using a given function of the data comprising the particular data item, wherein the given function randomly distributes its outputs; and

in response to a request for the particular data item, the request including at least the data identifier of the particular data item, providing the particular data item from a given one of the servers of the network of servers.

79. (New) A method as in claim 78 further comprising: maintaining accounting information relating to the data items; and

> RACK-1002 Page 148 of 351

Application of [.]ber et al, No. 09/283,160

using the accounting information as a basis for a value-based accounting system in which charges are based on an identity of the data items.

80. (New) A method as in claim 79 wherein the maintaining of accounting information includes at least some of:

(a) \tracking which data items have been stored on a system; and

(b) tracking which data items have been transmitted from a server.

81. (New) A method as in claim 79 further comprising: ensuring that a data item is not used by an unauthorized party.

82. (New) A content delivery method, comprising:
distributing a set of data items across a network of servers;
determining a hash of a particular data item; and
in response to a request for the particular data item, the request including
at least the hash of the particular data item, providing the particular data item from
a given one of the servers of the network of servers.

83. (New) A method as in claim 82 further comprising:resolving the request for the particular data item based on a measure ofavailability of at least one of the servers.

84. (New) A method as in claim 83 wherein the measure of availability for a server comprises one or more of:

(a) a measurement of bandwidth to the server;

Application of ber et al, No. 09/283,160

(b) a measurement of a cost of a connection to the server, and(c) a measurement of reliability of a connection to the server.

85. (New) A content delivery method, comprising:
distributing a set of data items across a network of servers;
for a particular data item having a particular data identifier specifying a
location in the network at which the particular data item may be located,
determining another data identifier for the particular data item, the other data
identifier including a data identifier determined using a hash of the particular data

in response to a request for the particular data item, the request including the other data identifier of the particular data item, providing the particular data item from a given one of the servers of the network of servers.

86. (New) A content delivery method, comprising:

distributing a set of data items across a network of servers, at least some of the data items being cached versions of data items from another server;

determining a data identifier for a particular data item, the data identifier including a hash of the particular data item; and

in response to a request for the particular data item, the request including at least the data identifier of the particular data item, providing the particular data item from a given one of the servers of the network of servers.

Application of l ber et al, No. 09/283,160

87. (New) A method of delivering a data item in a network comprising a plurality of processors, some of the processors being servers and some of the processors being clients, the method comprising:

storing the data item is on a first server in the network and storing copies of the data item on a set of servers in the network distinct from the first server; and

responsive to a client request for the data item, the request including a hash of the data item, causing the data item to be provided to the client.

88. (New) A method as in claim 87 wherein the data item has a contextual name comprising a pathname including a processor name and a file name, the method further comprising:

associating the contextual name of the data item with the hash of the data item.

89. (New) A method of delivering a data item in a network comprising a plurality of processors, some of the processors being servers and some of the processors being clients, the method comprising:

storing the data item is on a first server and storing copies of the data item on a set of servers distinct from the first server; and

responsive to a client request for the data item, the request including a value determined as a given function of the data in the data item, providing the data item to the client.

RACK-1002 Page 151 of 351

Application of ber et al, No. 09/283,160

90. (New) A method as in claim 89 wherein the data item has a contextual name comprising a pathname including a processor name and a file name, the method further comprising:

associating the contextual name of the data item with the value determined as the given function of the data in the data item.

91. (New) \setminus A method as in claim 54

混漏

wherein certain processors in the network communicate with each other using a TCP/IP communication protocol.

92. (New) A method of delivering a data item in a network comprising a plurality of processors, some of the processors being servers and some of the processors being clients, wherein some processors in the network communicate with each other using a TCP/IP communication protocol, wherein a key is required to identify a data item on the network and wherein ordinarily the key is a name or address for the data item, the method comprising:

storing some data items on a first server in the network and storing copies of some of the data items on a set of cache servers distinct from the first server;

determining a different cache key from the ordinarily used cache key, the different cache key being a function of the data it represents; and

responsive to a client request for the data item, the request including the different cache key for the data item, providing the data item to the client.

93. (New) A method as in claim 92 wherein the function is a message digest function or a hash function.

RACK-1002 Page 152 of 351

Application of 1 per et al, No. 09/283,160

94. (New) A method as in claim 93 wherein the function is selected from the functions: MD4, MD5, and SHA.

95. (New) A method as in claim 92 wherein the function randomly distributes its outputs.

96. (New) A framework operative in a computer network in which users of client processors connect to a content server, the framework comprising: a set of content servers, distinct from the content provider server, for hosting at least some of the data items that are normally hosted by the content provider server;

a mechanism constructed and adapted to determine an identifier for a data item as a given function of a data item in the network;

wherein, in response to requests for a data item, generated by one of the client machines the request including an identifier based on the given function of the particular data item, the particular data item is served from one of the content servers.

97. (New) A framework as in claim 96 wherein the given function is a message digest function or a hash function.

98. (New) A framework as in claim 97 wherein the given function is selected from the functions: MD4, MD5, and SHA.

Application of ^{ber} et al, No. 09/283,160

99. (New) A framework as in claim 96 wherein the given function randomly distributes its outputs.

100. (New) A framework as in claim 96 wherein processors in the network communicate with each other using a TCP/IP communication protocol.

101. (New) A framework as in claim 96 wherein the data item has a contextual name, the framework further comprising:

a mechanism constructed and adapted to associate the contextual name of the data item with the identifier for the data item.

102. (New) A framework as in claim 101 wherein the contextual name of the data item comprises a pathname including a processor name and a file name.

103. (New) In a network comprising a plurality of processors, some of the processors functioning as servers and some of the processors functioning as clients, wherein some processors in the network communicate with each other using a TCP/IP communication protocol, wherein a key is required to identify a data item on the network and wherein ordinarily the key is a name or address for the data item, a method of delivering a data item:

storing some data items on a first server in the network and storing copies of some of the data items from the first server on a set of cache servers distinct from the first server;

Application of L _____ Jer et al, No. 09/283,160

مى بوكلى كتم for a particular data item, determining a different cache key from the ordinarily used cache key for the data item, the different cache key being determined using a message function MD5 of the data comprising the particular data item; and

responsive to a client request for the particular data item, the request including the different cache key for the data item, causing the particular data item to be provided to the client,

wherein the data item may be a file, a portion of a file, a page in memory, a digital message, a digital image, a video signal or an audio signal.

104. (New) A framework operative in a computer network in which users of client processors connect to a content server, wherein processors in the network communicate with each other using a TCP/IP communication protocol, the framework comprising:

a mechanism constructed and adapted to determine a given function of a data item in the network, the given function being a message digest function or a hash function;

a set of content servers, distinct from the content provider server, for hosting at least some of the data items that are normally hosted by the content provider server;

wherein, in response to requests for a data item, generated by one of the client machines the request including an identifier based on the given function of the particular data item, the particular data item is served from one of the content servers.

Application of ber et al, No. 09/283,160

105. (New) A framework as in claim 96 wherein a data item may be a file, a portion of a file, a page in memory, a digital message, a digital image, a video signal or an audio signal.

106. (New) A content delivery method in a network in which at least some processors in the network communicate with each other using a TCP/IP communication protocol, the method comprising:

for a particular data item having a particular name specifying a location in the network at which the data item may be located, determining another name for the particular data item, the other name including a data identifier determined using message digest function MD5 of the data comprising the particular data item; and

in response to a request for the particular data item, the request including the other name of the particular data item, causing the particular data item to be provided from a given one of the servers of the network of servers,

wherein the data item may be a file, a portion of a file, a page in memory, a digital message, a digital image, a video signal or an audio signal

> RACK-1002 Page 156 of 351

Application of Farber e , No. 09/283,160

REMARKS

By this Amendment, the title has been replaced, all pending claims (claims 46-49) have been cancelled without prejudice or disclaimer of their subject matter, and new claims 54-106 have been added. Claims 1-45 and 50-53 were cancelled by preliminary amendment with the filing of this application. Claims 54-106 are pending in this application. Applicants submit that this application is in condition for allowance and an early action allowing the claims is earnestly solicited.

Should the Examiner believe that a personal interview would expedite the prosecution of this application, the Examiner is requested to contact the undersigned at the telephone number provided.

Respectfully submitted,

PILLSBURY, MADISON & SUTRO, LLP

1

By

Brian Siritzky Reg. No. 37497 Tel. No.: (202) 861-3702 Fax No.: (202) 822-0944

1100 New York Avenue, N.W. Ninth Floor Washington, D.C. 20005-3918 (202) 861-3000

30098456v8

			ĸ					
	IN THE	UNITED STAT	TES PATEN	IT AN	D TRADEMAR		PATENT APP	LICATION
Appln. No.: 09	es Code ↑ 999 er of Patents	283,160 Serial No. ↑	OIP I JAN 11	2001	Group Art U. Examiner: Atty. Dkt. F Appln. Title:	Homere, Je	Cilent Ref ATA IN A DA YSTEM	
Sir:			ATENT & TRA	DEMART	Deter	January 11, 200°		ology Cent
	PLY/AMENDME				Date:	•		
This is a reply/ame which is incorpora signature thereto.	endment/letter in the ted hereinto by ref	erence and the	signature be	low is	treated as the sig	with attachment of nature to the attach	ment in abse	ence of a
		FEE REQUIR	REMENTS F	<u>OR CL</u>	<u>AIMS AS AMENI</u>	<u>DED</u>		
1. Small Entity claim A. 🖾 <u>NOT</u> made B. 🖾 Withdrawn C. 🗌 made herewith D. 🗌 made previously	For B & C See <u>Required</u> Separate Paper (Pat-256)	Claims remaining after amendment	Highest nur previously pa	nber aid for	Present Extra	Large/Small Entity	Additional Fee	Fee Code Lg/Sm
2. Total Effective	Claims	53	**minus	20	33	x \$18/\$9 =	+ \$594	103/203
3. Independent Cl	aims	19	***minus	3	16	x \$80/\$40 =	+ \$1280	102/202
4. If amendment e time (leave blank i	enters <u>proper</u> multip if this is a <u>reissue</u> a	ole dependent c application)	laim(s) into t	his ap	olication for <u>first</u> add	+ \$270/\$135 =	·. +\$	104/204
	eby made to exten date this response ached (Usab		ue (h the (2 (3 mo.OA 4		\$110/\$55 = \$390/\$195 = \$890/\$445 = \$1390/\$695= \$1890/\$945=	+\$		115/215 116/216 117/217 118/218 128/228
7. Enter any previ	ous extension fee	paid since abov	re <u>original</u> du	e date	and subtract	- \$0		100
8.		- 11 0.1- 00(4)	official fac			on Fee Attached + \$110/\$55	+\$ +\$0	148/248
10. If IDS attached	claimer attached, d requires Official I) Request	Fee under Rule	97 (c),		add	+ \$180 + \$180	+ \$0	126 126
11. After-Final Re	quest Fee per rule	s 129(a) and 17	'(r)			+ \$710/355	+ \$0	146/246
	nal inventions for e					x \$710/355 ea	+ \$0	149/249
13. Request for Contract 14. Petition fee for	ontinued Examinat					+ \$710/355	+ \$0 + \$0	1179/1279
15.						EE ENCLOSED =	\$1874	<u> </u>
16. *If the entry in this 17. **If the "Highest t	s space is less than entr number previously paid number previously pai	for" in this space i	is less than 20, v	vrite "20 vrite "3" Our	s "0". " in this space.	No. 03-3975)	3	
filed, or which should have hereafter relative to this and duplicate copy of this she	ve been filed herewith or application and the result	concerning any paper ing Official Document	r filed hereafter, a under Rule 20, o	ind which or credit a	may be required under ny overpayment, to our	iny missing or insufficient Rules 16-18 (<u>missing or in</u> Accounting/Order Nos. shi is Query: Is app	fee(s) filed, or ass <u>isufficiencies only</u> own above, for wh Deal deadline) now or lich purpose a now? If
mou.			Winthrop LI			so, file Notice	e of Appeals	separately.
1100 New York A Ninth Floor	venue, NW		Brian Siritz		,	Reg. No.	37497	
Washington, DC Tel: (202) 861-300 Atty/Sec: BS/ans		Sig:	Dr.	inf 1	<u></u>	Fax: Tel:	(202) 822- (202) 861 -	
, eee, borailo	NOTE: File this	cover sheet in	duplicate w	ith PT	O receipt (PAT-1	03A) and attachm	nents	
30135752_1.DOC							P.	AT-120 12/00
1.000							1.	

RACK-1002 Page 158 of 351

Inventor(s): FARBER et al. Appln. No.: 09	283,160	ND TRADEMAR Group Art Unit Examiner: Atty. Dkt. PM	2771 Homere, J 252465 M#	PATEN APPLICATION ean R. Client Ref
Series Code ↑ Filed: April 1, 1999 Pat. No.:	Serial No. 1			DATA IN A DATA
Issued:	SHEAT& TRAN	EMAN		JAN 1 8 200
Hon. Commissioner of Patents Washington, D.C. 20231				Technology Center
Sir:		Date: Ja	anuary 11, 20	
NOTIFICAT	ION OF CHANGE IN SM	ALL/LARGE EN	TITY STAT	<u>US</u>
Please note the follo	wing change to/from Small Ent	ity Status:		ł.
	IOW : Applicant hereby claims S	Small Entity Status		
	<u>VN NOW</u>: Applicant hereby with		Status	
1100 New York Avenue, NW ^{Ninth Floor} Washington, DC 20005-3918 Tel: (202) 861-3000 Atty/Sec: BS/ans	Respectfully Submitted Pillsbury Winthrop LLP Intellectual Property Grou By Atty: Brian Sintzky Sig:	p M S	_ Reg. No. _ Fax: _ Tel:	

١

•

			w. m			
IN THE	UNITED STA	TES PATENT		K OFFICE	PATENT API	PLICATION
Inventor(s): FARBER et al. Appln. No.: 09	283,160		Group Art U Examiner: Atty. Dkt.	Homere, Je	ean R. 🗲	271
Series Code ↑ Filed: April 1, 1999	Serial No. 个	54 - 45	Appin. Title:	M# IDENTIFYING D	Client Ref	
Hon. Commissioner of Patents Washington, D.C. 20231		JAN 1 1 2001		PROCESSING		ÉCEIVI
Sir:	6	TE TRADEMART				AN 182
<u>REPLY/AMENDME</u>	NT/LETTER		Date:	January 11, 200	1 Techn	ology Cent
This is a reply/amendment/letter in the which is incorporated hereinto by refusion and the signature thereto.	erence and the	signature below	nd includes the here is treated as the sig CLAIMS AS AMEN	nature to the attac	f same date a hment in abso	and subject ence of a
1. Small Entity claim						
A. ⊠ <u>NOT</u> made B. ⊠ Withdrawn C. □ made herewith D. □ made previously For B & C See <u>Required</u> <u>Separate</u> Paper (Pat-256)	Claims remaining after amendment	Highest number previously paid fo	Preser : Extra r	Large/Smail Entity	Additional Fee	Fee Code Lg/Sm
2. Total Effective Claims	53	**minus 20	33	x \$18/\$9 =	+ \$594	103/203
3. Independent Claims	19	***minus 3	16	x \$80/\$40 =	+ \$1280	102/202
 If amendment enters proper multip time (leave <u>blank</u> if this is a reissue a Optimized due Date; 				+ \$270/\$135 =	+\$	104/204
 <u>Original due Date:</u> Petition is hereby made to exten 	d the original d	UNONE	o) \$110/\$55 =			115/215
ate to cover the date this response requisite fee is attached (Usab	is filed for which $\log \frac{1}{2}$	h the (2 mo (3 mo no.OA 4 mo no.OA 5 mo	s) \$890/\$445 = s) \$1390/\$695=	+\$		116/216 117/217 118/218 128/228
7. Enter any previous extension fee	paid since abov	e <u>original</u> due da		- \$0		
8.				on Fee Attached	+\$	148/248
9. If Terminal Disclaimer attached, 10. If IDS attached requires Official F				+ \$110/\$55 + \$180	+ \$0	148/248
or if Rule 97(d) Request				+ \$180	+ \$0	126
11. After-Final Request Fee per rule:	s 129(a) and 17	(r)		+ \$710/355	+ \$0	146/246
12. No. of additional inventions for ex				x \$710/355 ea	+ \$0	149/249 1179/1279
13. Request for Continued Examinat				+ \$710/355	+ \$0 + \$0	11/9/12/9
14. Petition fee for 15.				EE ENCLOSED =	\$1874	
 *If the entry in this space is less than entry 7. **If the "Highest number previously paid 18. ***If the "Highest number previously paid 	for" in this space i	s less than 20, write is less than 3, write C	Ilt is "0". "20" in this space.	No. 03-3975) 104 000000	3	
<u>CHARGE STATEMENT</u> : The Commissioner is he filed, or which should have been filed herewith or hereafter relative to this application and the resulti <u>duplicate</u> copy of this sheet is attached.	concerning any paper	filed hereafter, and wh	nich may be required under	Rules 16-18 (missing or in	fee(s) filed, or ass sufficiencies only) now or
This CHARGE STATEMENT <u>does not authorize</u> filed.			sue fee transmittal sheet	is Query: Is app so, file Notice		
		Winthrop LLP al Property Gro	αu			
1100 New York Avenue, NW	By Atty:	Brian Sintzky	(Reg. No.	37497	
		194. 1		_	(000) 000	
Ninth Floor Washington, DC 20005-3918	Sig:	VIII		Fax:	(202) 822-	.0944

30135752_1.DOC

.

ë j

.

PAT-120 12/00

RACK-1002 Page 160 of 351

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Group Art Unit:

Examiner:

In re PATENT APPLICATION of

FARBER et al.

Appln. No. 09/283,160

Filed: April 1, 1999

For: IDENTIFYING DATA BASED ON CONTENTS OF DATA IN NETWORK (As Amended

MAR 0 2 200

March 2, 2001

2776

Homere, Jean H

#4

PETITION TO MAKE SPECIAL BECAUSE OF ACTUAL INFRINGEMENT

Hon. Commissioner of Patents and Trademarks Washington, D.C. 20231

Sir:

27

Applicant hereby petitions to make the above-identified patent application special because of actual infringement, pursuant to Section 708.02(II) of the Manual of Patent Examining Procedure ("MPEP").

Accompanying this petition is a Declaration of Facts in Support of Petition to Make Special Because of Actual Infringement.

Also accompanying this petition is a check in the amount of \$130 as specified in 37 CFR 1.17(i). Please charge any additional fees or credit any overpayment to Deposit Account No. 03-3975, Order No. 018404/0000003.

1

Respectfully submitted,

PILLSBUR¥ WINTHROP LLP

By

Brian Siritzky) Reg. No. 37497 Tel. No.: (202) 861-3702 Fax No.: (202) 822-0944

1100 New York Avenue, N.W. Ninth Floor Washington, D.C. 20005-3918 (202) 861-3000 30152402v1

03/05/2001 CCHAU1 00000024 09283160

01 FC:122

130.00 OP

RACK-1002 Page 161 of 351

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re PATENT APPLICATION of

FARBER et al.

Appln. No. 09/283,160

Filed: April 1, 1999



Group Art Unit: 2776

Examiner: Homere, Jean R.

For: IDENTIFYING DATA BASED ON CONTENTS OF DATA IN NETWORK (As Amended)

March 2, 2001

COUNTY POINT

DECLARATION OF FACTS IN SUPPORT OF PETITION TO MAKE SPECIAL BECAUSE OF ACTUAL INFRINGEMENT

Hon. Commissioner of Patents and Trademarks Washington, D.C. 20231

Sir:

I, Brian Siritzky, of the law firm of PILLSBURY WINTHROP LLP, 1100 New York Avenue, N.W., Washington, D.C. 20005-3918, Reg. No. 37,497, telephone number (202) 861-3702, am the attorney for the applicants in the above-identified case and make the following declarations:

- 1. There is an actual infringement of this invention by a service now publicly available.
- Various promotional materials for the service and technical papers (including a White Paper) describing the service are publicly available. Features of the infringing service are provided in the promotional materials and technical papers.
- 3. An analysis of various Internet web sites confirms that actual use of the system and the actual infringement of this invention.
- 4. The White Paper and promotional materials describe a system operating on the Internet.

Declaration of Facts in Support of Petition to Make Special Because of Actual Infringement

ا البغير

5.	One feature of this service provides a system in which a set of data items are
	distributed across a network of servers, at least some of the data items being cached
	versions of data items from a source server. Specifically, the White Paper states
	'Currently, more than 1475 servers are deployed at more than 55 network
	providers around the world. This installation includes global providers with data
	centers in Europe, the Pacific Rim, Australia, Asia, Africa and South America, putting
	servers in 24 different countries."
6.	Another feature describes how the system operates to serve data items to users.

"First, the user's browser sends a request for a web page to the site. In response, the web site returns the appropriate HTML code as usual, the only difference being that the enclosed embedded object URLs have been modified to point to the . . . network. As a result, the browser next requests and obtains the media-rich embedded objects from an optimally located. . . . server, instead of from the home site."

- 7. Another feature describes determining, for a particular data item, a data identifier. Specifically, the White Paper describes modifying a URL to include a so-called "object data" field. "The object data field is used to guarantee object freshness ..., this field will contain either the object's expiration time, or a string that uniquely identifies a particular version of the object.... This field may be set up to be the cryptographic hash (MD5) of the object contents"
- 8. Another feature describes providing the data item from a given one of the servers in the system based on a request that includes the identifier (i.e., in this case, the object data filed). Specifically, the White Paper teaches that "... when the object is modified, its object data field changes, ... its ARL (embedded in the container HTML page) changes as well. Any user that requests the page after an update receives a page that points to the new version of the object"

2

Declaration of Facts in Support of Petition to Make Special Because of Actual Infringement

9.	I have accessed various Internet Web sites that use the alleged infringing service, and
	these sites include modified URLs that appear to include the object data field
	described in the service provider's White Paper. I have examined the modified URLs
	that include that object data field and I have determined that they point a user to a
	server in a network of servers.
10	. I have compared the alleged infringing service with the claims of this application.
11	. In my opinion, at least claims 54-58, 60, 64, 65, 69-78, 82, 85-87, 89, 91-100 and
	103-106 on file in this application are unquestionably infringed.
12	. For example, claim 54 of this application has two steps: (1) determining a data
	identifier for a particular data item, the data identifier being determined using a given
	function of the data comprising the particular data item; and (2) responsive to a
	request for the particular data item, the request including at least the data identifier of
	the particular data item, providing the particular data item from a given one of the

servers of the network of servers.

اتنو

- The steps of claim 54 appear to correspond to the method of accessing data items using modified URLs with the so-called "object data" field, discussed in paragraph 7 above.
- 14. I have made similar assessments of infringement for the other independent claims.
- 15. In addition, claims 73-75 require that the given function to be applied to the data items be "a message digest function or a hash function" (claim 72), be "selected from the functions: MD4, MD5, and SHA" (claims 73), "randomly distributes its outputs" (claim 74); and "produces a substantially unique value based on the data comprising the data item."
- 16. As discussed in paragraph 7 above, in the alleged infringing service the value in the "object data" field may be computed using an MD5 hash.

Declaration of Facts in Support of Petition to Make Special Because of Actual Infringement

- 17. Similar arguments apply to claims 77 and 93-95.
- 18. Some of the claims (e.g., claims 91, 92, 100, 103-105) specify that the network is one "wherein some processors in the network communicate with each other using a TCP/IP communication protocol." The Internet operates based on TCP/IP protocol, and the alleged infringing service operates on the Internet..
- 19. I have a good knowledge of the pertinent prior art and I believe that each reference most closely related to the subject matter of encompassed by the claims is already of record in this case or has already been provided to the Examiner in an Information Disclosure Statement.

I declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true, and further that these statements are made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of this application and any patent issuing thereon.

Dated: March 2, 2001

By Brian Siritz

30152415vl

IN THE	E UNITE STA	TES PATEN	r and '	TRADEMARK	(C JE)	PATENT APF	LICATI
				Group Art Un	it771		
Inventor(s): FARBER et al.				Examiner:	Homere, Je	ean R.	
Appin. No.: 09	283,160			Atty. Dkt.	P 252465		
Series Code 1	Serial No. 个	OIPE		•	M#	Client Ref	
Filed: April 1, 1999		0	2	Appln. Title:	IDENTIFYING D	ATA BASED	<u>ON</u>
Hon, Commissioner of Patents			A		CONTENTS OF	DATA IN NE	TWORK
Washington, D.C. 20231	0	MAR 0 2 2001			(As Amended)		
Huomington, Bron Leter	PATER		Se l			No P	1
Sir:		PRADEMA PU	/			ΰų .	~ <
REPLY/AMENDM	ENT/LETTER	A A DEBA		Date:	March 2, 2001		E.
This is a reply/amendment/letter in which is incorporated hereinto by resignature thereto.	eference and the	signature belo	ow is trea	cludes the here eated as the sign MS AS AMENE	nature to the attacl	f same date a	nd subj
	FEE REQUI	KEIMEN 13 FU	<u>R GLAII</u>	IVIS AS AIVIENL			
1. Small Entity claim A, □ NOT made For B & C	Claims	Highest numb	ar l	Present Extra	Large/Small Entity	Additional	Fee Co
B Withdrawn See Required	remaining after	previously paid				Fee	110
C. made herewith Separate Paper (Pat-256)	amendment						Lg/S
D. 🖾 made previously							
O Total Effective Claims	5	**minus	20 0	<u> </u>	x \$18/\$9 =	+:\$0	103/2
2. Total Effective Claims	3		3 0		x \$80/\$40 =	+ \$0	102/2
3. Independent Claims 4. If amendment enters proper mult				·	χψου/ψ+υ		<u> </u>
time (leave <u>blank</u> if this is a <u>reissue</u>	application)		is applic	add add	+ \$270/\$135 =	+ \$0	104/20
5. Original due Date:	application			udu [
6. Petition is hereby made to exte	and the original d		mo) \$	\$110/\$55 =	And a state of the		115/2
date to cover the date this response	e is filed for whic	the (2 n		390/\$195 =	+ \$		116/2
requisite fee is attached				890/\$445 =	·	Ť	117/2
(Lles	able <u>only</u> for $\leq 2i$			1390/\$695=			118/2
(Usable o	only for 30 day/1	mo.OA 5 n		\$1890/\$945=			128/2
				al auchána at	- \$0		uri destatori
7. Enter any previous extension fee	paid since abov	ve <u>originai</u> que	uale an	Extensi	on Fee Attached	+\$	
8. 0. If Terminal Disolaimer attached		official foo			+ \$110/\$55	+ \$0	148/24
9. If <u>Terminal Disclaimer</u> attached 10. If IDS attached requires Official	I, aug Rule 20(u)			odd	+ \$180		126
or if Rule 97(d) Request					+ \$180	+ \$0	126
11. After-Final Request Fee per rul		7/r)		auu	+ \$710/355	+ \$0	146/24
12. No. of additional inventions for					x \$710/355 ea	+ \$0	149/24
13. Request for Continued Examina	examination per	Rule 129(b)			+ \$710/355	+ \$0	1179/1
44 5 111 1 1					+ \$110/333	+ \$130	
					E ENCLOSED -	\$130	<u> </u>
15.16. *If the entry in this space is less than en	try in next enang the	"Drecent Evtre" -	recult ic "'		E ENCLOSED =	\$150	
17. **If the "Highest number previously pa 18. ***If the "Highest number previously pa	id for" in this space i	is less than 20, wri	ite "20" in	n this space.			
	-		Our De	eposit Account	No. 03-3975)		
			(Our O	Order No. 701			
CHARGE STATEMENT: The Commissioner is I	hereby authorized to cl	hame any fee specif	fically autho	orized hereafter or a	C# M	•	ortod to be
filed, or which should have been filed herewith o	or concerning any pape	r filed hereafter, and	d which may	ay be required under F	Rules 16-18 (missing or li	nsufficiencies only	now or
hereafter relative to this application and the resu	Iting Official Document	t under Rule 20, or c	redit any o	overpayment, to our A	ccounting/Order Nos. sh	own above, for wh	ich purpos
	ze charge of the issue	<u>e fee</u> until/unless a	n issue fe	e transmittal sheet i	S Query: Is app	peal deadline	now? I
duplicate copy of this sheet is attached. This CHARGE STATEMENT does not authorized					so, file Notice		
duplicate copy of this sheet is attached.			,		,		
duplicate copy of this sheet is attached. This CHARGE STATEMENT does not authorized	Pillsburv	Winthrop LLF					
duplicate copy of this sheet is attached. This CHARGE STATEMENT does not authorized				/			
duplicate copy of this sheet is attached. This CHARGE STATEMENT does not author: filed. 1100 New York Avenue, NW		Winthrop LLF Jal Property G Brian Siritzky	froup	1	Reg. No.	37497	
duplicate copy of this sheet is attached. This CHARGE STATEMENT <u>does not authori</u> filed. 1100 New York Avenue, NW Ninth Floor	Intellectu By Atty:	ual Propert∦ Ģ	froup	Tehn			
duplicate copy of this sheet is attached. This CHARGE STATEMENT does not author: filed. 1100 New York Avenue, NW	Intellectu	ual Propert∦ Ģ	froup	Ety-	Reg. No. Fax: Tel:	37497 (202) 822- (202) 861-	

١

2

RACK-1000220 12/00 Page 166 of 351



UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner For Patents United States Patent and Trademark Office Washington, DC 2023 www.uppto.gov

APR 1 3 2001

Mailed

Technology Center 2100

Paper No. 5

ł

Brian Siritzky Pillsbury Winthrop LLP 1100 New York Avenue, N.W Ninth Floor Washington, D.C. 20005-3918

In re Application of: David A. Farber et al. Application No.: 09/283,160 Filed: April 1, 1999 For: IDENTIFYING DATA BASED ON CONTENTS OF DATA IN NETWORK

DECISION ON PETITION TO MAKE SPECIAL

This is a decision on the petition filed March 2, 2001 under 37 C.F.R. 102(d) and M.P.E.P. 708.02(II): Infringement, to make the above-identified application special.

A grantable petition under 37 C.F.R. $\S1.102(d)$, and M.P.E.P. \$708.02, Section II, must be accompanied by payment of the fee under 37 C.F.R. \$1.17(i) and a statement under 37 C.F.R. \$1.102 by the applicant or assignee or statements by an attorney/agent registered to practice before the Patent and Trademark Office that: (A) there is an infringing device or product actually on the market or method in use; (B) a rigid comparison of the alleged infringing device, product, or method with the claims of the application has been made, and that, in his or her opinion, some of the claims are unquestionably infringed; and (C) he or she has made or caused to be made a careful and thorough search of the prior art or has a good knowledge of the prior art. A fee under 37 C.F.R. for such a petition is required .

Applicant's submissions meet all the criteria set out above, accordingly, the Petition is **GRANTED**. The application file is being forwarded to the Examiner of Record for expedited examination.

hill

Robert A. Weinhardt Special Program Examiner Technology Center 2100 Computer Architecture, Software, & Electronic Commerce

IN THE UNITES STATES PATENT AND TRADEMARK OFFICE

In re PATENT APPLICATION OF

FARBER et al.

Group Art Unit: 2771

Examiner: Jean Homere

HECEWED AND

Filed: April 1, 1999

Appln. No.: 09/283,160

Title: IDENTIFYING DATA IN A DATA PROCESSING SYSTEM

* * * *

April 18, 2001

INFORMATION DISCLOSURE STATEMENT

Hon. Commissioner of Patents and Trademarks Washington, D.C. 20231

Sir:

Attached is Form PTO-1449 listing the enclosed documents.

Contingent Request Under Rule 97(c): Should a First Action on the merits have been issued on the same day or before this Information Disclosure Statement is filed, please accept this Information Disclosure Statement under Rule 97 (c) and charge the requisite Rule 17(p) fee to our Deposit Account No. 03-3975 under Order No. 007018/252465, and proceed to consider this Information Disclosure Statement.

Contingent Petition Under Rule 97(d): Should a Notice of Allowance have been issued on the same day as or before the filing date of this Information Disclosure Statement, please consider this a Petition under Rule 97(d) (ii), charge the petition fee to our Deposit Account No. 03-3975 under Order No. 007018/252465, and proceed to consider this Information Disclosure Statement under Rule 97(d).

This Information Disclosure Statement is intended to be in full compliance with the rules, but should the Examiner find any part of its required content to have been omitted, prompt notice to that effect is earnestly solicited, along with additional time under Rule 97(f) to enable Applicant to comply fully.

RACK-1002 Page 168 of 351

09/283,160 April 18, 2001 Page 2

.....

Consideration of the foregoing and enclosures plus the return of a copy of the herewith Form PTO-1449 with the Examiner's initials in the left column per MPEP 609 along with an early Action on the merits of this application are earnestly solicited. Applicant hereby expressly reserves the right to swear behind the effective date of the references and to question the relevance and materiality of the references cited, in whole, in part, or in combination, subsequent to filing this Information Disclosure Statement.

Respectfully submitted,

PILLSBURY MADISON & SUTRO LLP

By

Brian Siritzky Reg. No.: 37497 Tel. No.: (202) 861-3702 Fax No.; (202) 822-0944

BS/ans 1100 New York Avenue, N.W. Ninth Floor Washington, D.C. 20005-3918 (202) 861-3000

2

	epartr	9 (modified) ment of Commerce -1449)	•				_	Atty. Dkt. No.	M#				Client	Ref.		
		rademark Office														
NFORMA BY APPL		N BISELESSU	RE ST	ATEN	IENT			Applican	25246 t: FARI	BER et al.	10		9,0,	Ŕ	1 1 1 1 1 1 1	
		APR 1 8 2001						Appln. N	o.: 09/2	283,160		TANO	<u>~</u> ~	,	1/2	
	1.	AFR 1 4 2001	Ë)					Filing Da		· ·		- 6	De la	<u>کی</u>	$\frac{1}{2}$	>
Date: April	18.2	The st	Page	1	of	3]	Examine			Gro	מענ	Art	ăit: 2	5271	
OUD ACTIVITY IN THE OWNER	Gunnosecuou	DOCUMENTS					an in the state		in ooun					3	ini vetomist	÷.,
Examiner'		Document	S. BARRING	Date	<u></u>	Name	<u></u>	is a target particular.	<u>ti dan di</u>	Los Monterios	Clas		Sub	q	Filing	
3		Number		MM/	~~~~	(Family N	Jame	of First In	ventor)		Olas	.5	Clas	s	Date	
nitials*	1					() annay i	vanie v	or i a oc in	vontory						(if approp	oria
JRH	AR	5,638,443		06/19	97	STEFIK e	et al.				705		54		11/23/9	4
1		5,802,291		09/19		BALICK					709		202	_	3/30/95	
		5,542,087		07/19		NEIMAT			_		707		10		10/15/9	3
		5,287,499		02/19		NEMES					707		2		5/16/91	
		4,922,417		05/19		CHURM	et al.				707		1		10/24/8	6
	1	5,341,477		08/19		PITKIN e	-				709		226		8/6/93	
V	<u> </u>	5,907,704	_	05/19	99	GUDMUN		N et al.					,		4/3/95	
	HR													-		
	IR															
	JR															
	KR															
	LR			_												
	MR															
	NR															
OREIGN	PATI	ENT DOCUMEN	TS								43.00 Galledoudes		glish		Translat	tior
		Document	Date		Coun	try	Inven	tor Name	,			Ab	stract		Readily Availabl	е
		Number	MM/Y	YYY								End	closed	No	Enclose	
	OR	05162529	06/19	93	JP		HOR	OSHI				x				Ť,
		ng in this order A			1	ical Name			nt Page	s etc)				<u>+</u>	<u> </u>	ŕ
		Gwertzman, Jar Report HU TR 3 1994, 2 pgs.	nes, et	al. "T	he Ca	se for Geo	graph	ical Push	-Cachir	na." Techr	nical 38,	3				
	QR.	Grigni, Michelan SIAM Journal of	igelo, e Discre	et al. "T ete Ma	Fight B thema	ounds on tics, Vol. 4	Minim 4, No.	um Broa 2, May 1	dcasts 991, po	Networks. s. 207-22	" 2.					
	RR⊻	Devine, Robert. Hashing Algorith Foundations of I	"Desig 1m." In	n and Proce	Imple edings	mentation s of 4th Int	of DD ternati	H: A Dist onal Cont	ributed ference	Dynamic on						
	SR 🗸	Deering, Stephe Extended LANs, 1990, pgs. 85-1	en, et al ." ACM	l. "Mul	ticast I	Routing in	Datag	ram Inter	rnetwor	ks and	iy					
		Cormen, Thoma Massachusetts,	1994,	pgs. 2	19-243	3, 991-993	3.									
	UR 🗸	Naor, Moni, et a Proceedings of t	I. "The	Load,	Capa E Sym	city and Av	vailabi	lity of Qu dations o	orum Sy f Comp	ystems." I uter Scier	n nce,					
		November 1994	, pgs. 2	<u>214-2</u> 2												

. م

W FORM	oartm PAT-1	ent of Commerce						Atty. Dkt. No.	M# 25246	35		Client F			
NFORMA	TIO	N DISOLOEUR	EST	ATEME	INT			Applicant		BER et al.	lech	ALL A	K	Nr.	
BY APPLI	CAN	1	A					Appln. No	o.: 09/	283,160		eller and a second	5	$\overline{0}$	_
		APR 1 8 2001	U.S.					Filing Dat				Cen	<u>ر</u> ا	Ŋ ₁	
	40.0	A A A A A A A A A A A A A A A A A A A	Sage	2	of	3	ן ר	Examiner			Grou	ıp Art Ur	iii?2	771	_
Date: April		OCUMENTS				NK Shake	enine - a ege Mintal e a distric	ani na si	a canadar Silandar	Carlo Bartino Martino Martino (M	o de Mediana A Statistica de Característica de Característica de Característica de Característica de Característica de Carac			2	añ 15
		Document		Date	<u>di yan kana di</u>	Name	1 1000				Class			Filing	
		Number		MM/Y	YYY		Name	of First Inv	ventor)		Class	s 	Date (if appropr	ria
nitials*		INT DOCUMEN	тч									English		Translati Readily	iO
OREIGIN			Date		Cour	ntrv	Inver	ntor Name				Abstract		Available	e
		Number	MMA											ļ	т
		Number					1					Enclosed	No	Enclose	
		ing in this order /	Author	Title F	Perio	dical Nam	e. Dat	e. Pertine	nt Pag	es, etc.)					
JRH	AR V	Nisan, Noam. "P Proceedings of t	suedo he Tw	random enty-Se	Ger conc	nerators fo Annual A	or Space	ce-Bounde	ed Cor	mputation."	In	i i			
-1-		Computing, May Palmer, Mark, et	al "E	ido: A C	ache	e that Lear	ns to I	Fetch." In	Proce	edings of th	e				
		17th Internationa 255-264.	al Cont	erence	on V	ery Large	Data	Bases, 56	pterm		js. 				
	CR	Peleg, David, et Computation 123	3, 199	5, 210-2	223.										_
		Rabin, Michael. Fault Tolerance.	" Jour	nal of th	e AC	CM, Vol. 30	6, <u>No.</u>	2, April 19	<u>989, p</u>	<u>gs. 335-348</u>			-		
		Ravi, R., "Rapid In Proceedings o November 1994	of the : , pgs. :	35th IEE 202-213	EE S 3.	ymposium	on Fo	oundation	of Cor	mputer Scie	nce,				
	FR	Schmidt, Jeanet Independence." Algorithms, 1993	te, et a In Pro	al. "Che ceeding	rnoff js of	-Hoeffding the 4th AC	g Boun CS-SIA	ds for App M Sympo	plicatio osium	ons with Lim on Discrete	ited				
	GR	Tarjan, Robert E √ol. 22, No. 11,	Endre,	et al. "S	Storin	ng a Spars ogs. 606-6	e Tabl 11.	e." Comm	nunica	tions of the	ACM,				
	ΗŔ	Wegman, Mark, Set Equality.'' Jo 265-279.	et al	"New H	ash	Functions	and T	heir Use i iences Vo	n Auth 91. 22, -	entication a June 1981,	nd pgs.				
		Vitter, Jeffrey So Proceedings of November 1991	32nd , pgs.	IEEE Sy 121-13	ympo 0.	osium on F	Founda	ations of C	Compu	ter Science					
		Fredman, Micha Time." Journal o 1984, pgs. 538-	ael, et of the / 544.	al. "Stor Associa	ring a tion t	for Compu	iting M	lachinery,	Vol. 3	1, No. 3, Ju 	ly 	 			
		Yao, Andrew Ch Computing Mac	hinery	, Vol. 28	B, <u>N</u> c	5. 3 <u>, July 1</u>	<u>981, p</u>	<u>gs. 615-6</u>	28.				_		
*		Floyd, Sally, et a Application Leve	el Frar	nin <u>g." Ir</u>	<mark>۱ Pro</mark>	ceeding o	<u>f ACM</u>	SIGCOM	IM_'95	pgs. 342-3	56	ļ			
	MR	Feeley, Michael Cluster." In Pro	l, et al. ceedin	"Impler gs of th	ment e 15	ting Globa th ACM Sy	l Memo ympos	ory Manag ium on O	gemer peratir	it in a Work ig Systems	station				
		Principles, 1995	ō, pgs.	201-21	2.					red: 5/3	30/6	1			_

. 1 - 22

	epartn	nent of Commerc	e					Atty. Dkt. No.	M#			Client	Ref.	gelda Gelación Maria (277	He (s
PW FORM Patent		1449) rademark Office										A 4	~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	2007 5-2007 5-2100	
								Applicar	1	RBER et al.		NDO/OC		ک رخ	C
NFORM/ BY APPL		N DISCLOSE	REST	ATEN	IENT			Applicar	IC. FF	RDER et al.		9	Con	-007	
		/	A					Appin. N	lo.: 0	9/283,160				87270	
		APR 1 8 2						Filing Da	ate: A	pril 1, 1999				00	
Date: April	18, 2	200 1 20	Res.	3	of	3]	Examin	er: Je	an Homere	Grou	ip Art U	ni <u>t:</u> 2	771	
U.S. PATE		OCUMENTE	MAC		1000 1000	and and the second s Second second	inia - 1	en noble (elimination	978877789591 	an a		na en an	en e	n na sana na Ana ana ang taong tao	61
Examiner'		Document		Date		Name					Class	-		Filing	
s Initials*		Number		MM/\	////	(Family I	Name	of First I	nvent	or)		Clas	s	Date (if approp	_
FOREIGN	PAT	ENT DOCUME	NTS		<u></u>							English Abstract		Translat Readily	ion
		Document	Date	~~~	Coun	itry	Inver	ntor Nam	e			10011001		Available	е
		Number	MM/\	TTTT								Enclosed	No	Enclose	N
		ng in this order	and hands to had a set	ALC STREET				1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1	2 8 7					T	╪
JRH	BR	PATENT ABS Communicatio	TRACT	S OF . ol Met	JAPAN hod in	l, "Electro The Syst	nic Ma em." 0	ail Multipl 6/30/199	exing 193, J	System and P 05162529					
	CR												-		╀
	DR		_			-							-		╋
	ER FR		_										+		+
	GR		_	-	_			_							t
	HR				_				_			_	\uparrow	<u> </u>	T
	IR														Γ
	JR														
	KR													<u> </u>	
	LR											_	-	<u> </u>	+
	MR							1			<u> </u>				
Examiner	Je		Home					Date Co				01			
*EXAMINE		Initial if citation of ce and not consid										v line th	rough	citation	if
	- man		10100.11	0.000 0			101	a commu	nound	in to / oppriour	<u>. </u>				

23

,

thesa . Unite	UN UN	nited States P. ress: COMMISSI	Patent and Tradema	ark Office
NG DATE	FIRST NAMED I	NVENTOR	ATTORN	IEY DOCKET NO.
04/01/99	FARBER		<u>i</u> t for	M252465
	TM627660		EXAMINE	ER
PROPERTY G	RO ROUP	, L		PAPER NUMBER
< AVENUE NW EAST TOWER		L		PAPER NUMBER J
20005-391	8		DATE MAILED:	67 (0F /04
				06/05/01
			2	
				1- File Copy
	ING DATE 04/01/99 DISON & SUTI PROPERTY GI K AVENUE NW EAST TOWER 20005-391	ING DATE FIRST NAMED I 04/01/99 FARBER TM02/0605 DISON & SUTRO PROPERTY GROUP K AVENUE NW EAST TOWER 2 20005-3918	UNITED STATES United States P. Address: COMMISSI Washington 04/01/99 FARBER 04/01/99 FARBER 04/01/99 FARBER TM02/0605 DISON & SUTRO PROPERTY GROUP K AVENUE NW EAST TOWER 2 20005-3918	UNITED STATES DEPARTMENT O United States Patent and Tradema Address: COMMISSIONER OF PATENTS AND Washington, D.C. 20231 UNG DATE FIRST NAMED INVENTOR ATTORNE 04/01/99 FARBER D PN COMMISSIONER OF PATENTS AND Washington, D.C. 20231 ING DATE FIRST NAMED INVENTOR ATTORNE 04/01/99 FARBER D PN EXAMINE TM02/0605 DISON & SUTRO PROPERTY GROUP K AVENUE NW EAST TOWER 2177 D 20005-3918 C

Office Action Summary	Application No. 09/283,160	Applicant(s)	Farber	et al.
	Examiner Jean R. Ho	1	Art Unit 2177	
The MAILING DATE of this communication app	pears on the cover sheet	with the corres	pondence add	dress
Period for Reply A SHORTENED STATUTORY PERIOD FOR REPLY IS THE MAILING DATE OF THIS COMMUNICATION. - Extensions of time may be available under the provisions of 37 CF after SIX (6) MONTHS from the mailing date of this communica - If the period for reply specified above is less than thirty (30) days, be considered timely.	R 1.136 (a). In no event, how	ever, may a reply l	be timely filed	
 If NO period for reply is specified above, the maximum statutory p communication. Failure to reply within the set or extended period for reply will, by s Any reply received by the Office later than three months after the rearned patent term adjustment. See 37 CFR 1.704(b). 	tatute, cause the application to			6 1 2 2)
Status 1) ⊠ Responsive to communication(s) filed on <u>Apr 1</u>	8, 2001			
	action is non-final.			2
3) Since this application is in condition for allowand closed in accordance with the practice under E	e except for formal matte	ers, prosecutio 1; 453 O.G. 21	n as to the m 3.	erits is
Disposition of Claims				
4) 🕅 Claim(s) _54-106			is/are pen	ding in the applica
4a) Of the above, claim(s) <u>none</u>				
5) 🗍 Claim(s)				
6) 🔀 Claim(s) <u>54-106</u>				
7) 🗌 Claim(s)				
8) [] Claims				
Application Papers			estriction and	indi cicotion require
9) X The specification is objected to by the Examiner.				
10) 🕅 The drawing(s) filed on <i>Apr 1, 1999</i>	is/are objected to by the	Examiner		
11) The proposed drawing correction filed on			disapprove	ed and
12)		appioted b)		50.
Priority under 35 U.S.C. § 119				
13) \square Acknowledgement is made of a claim for foreign p	priority under 35 U.S.C. &	119(a)-(d)		
a) All b) Some* c) None of:	, ,	() (-).		
1. 🗌 Certified copies of the priority documents ha	ve been received.			
2. 🗌 Certified copies of the priority documents ha		ication No.		
3. Copies of the certified copies of the priority of application from the International Bure *See the attached detailed Office action for a list of the	documents have been red au (PCT Rule 17.2(a)).	ceived in this N		2
(4) Acknowledgement is made of a claim for domestic				
ttachment(s)				
5) X Notice of References Cited (PTO-892)	18) 🔄 Interview Summary (F	TO-413) Paper No(s)		
5) X Notice of Draftsperson's Patent Drawing Review (PTO-948)	19) 🗌 Notice of Informal Pat			
7) X Information Disclosure Statement(s) (PTO-1449) Paper No(s). 6	20) Other:			

F	эΤ	0	-32	6 (Rev.	9-	-00)

Part of Paper No. 7

RACK-1002 Page 174 of 351 Serial Number: 09/283,160

Art Unit: 2177

Page 2

DETAILED ACTION

Information Disclosure Statement

1. The information disclosure statement (IDS) filed on 04/18/01 complies with the provisions of MPEP § 609. The information referred to therein has been considered as to the merits. (see attached PTO-1449).

Drawings

2. This application has been filed with informal drawings which are acceptable for examination purposes only.

3. The drawings are objected to under 37 CFR 1.83(a). The drawings must show every feature of the invention specified in the claims. Therefore, the plurality of servers, including a source server, that allows a client to request and retrieve a cached data item through a hashed identifier must be shown or the feature(s) canceled from the claim(s). No new matter should be entered.

4. The drawings are objected to under 37 CFR 1.83(b) because they are incomplete. In particular, a flowchart was not provided to illustrate the claimed method steps. 37 CFR 1.83(b) reads as follows:

When the invention consists of an improvement on an old machine the drawing must when possible exhibit, in one or more views, the improved portion itself, disconnected from the old structure, and also in another view, so much only of the old structure as will suffice to show the connection of the invention therewith.

Correction is required.