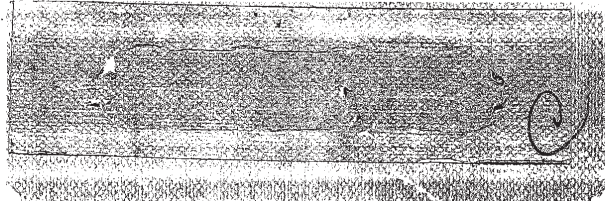


1c518 U.S. PTO
09/283160
04/01/99

707	Class	ISSUE CLASSIFICATION
2	S. subclass	



PATENT NUMBER
6415280
6415280

U.S. UTILITY PATENT APPLICATION

O.I.P.E. SCANNED <i>[Signature]</i> Q.A. <i>[Signature]</i>	PATENT DATE JUL 02 2002
--	----------------------------

SECTOR	CLASS 707	SUBCLASS 2	ART UNIT 2/77	EXAMINER <i>[Signature]</i>
--------	---------------------	----------------------	-------------------------	--------------------------------

FILED WITH: DISK (CRF) FICHE
(Attached in pocket on right inside flap)

PREPARED AND APPROVED FOR ISSUE

ISSUING CLASSIFICATION						
ORIGINAL		CROSS REFERENCE(S)				
CLASS	SUBCLASS	CLASS	SUBCLASS (ONE SUBCLASS PER BLOCK)			
707	2	707	3	10	101	
INTERNATIONAL CLASSIFICATION		709	203	219	229	
G06F	17 / 30					

Continued on Issue Slip Inside File Jacket

<input checked="" type="checkbox"/> TERMINAL DISCLAIMER	DRAWINGS			CLAIMS ALLOWED	
	Sheets Drwg. 24	Figs. Drwg. 36	Print Fig. 1A, 1B	Total Claims 55	Print Claim for O.G. 1
<input type="checkbox"/> a) The term of this patent subsequent to _____ (date) has been disclaimed.	_____ (Assistant Examiner) (Date)			NOTICE OF ALLOWANCE MAILED 10-25-01	
<input checked="" type="checkbox"/> b) The term of this patent shall not extend beyond the expiration date of U.S. Patent. No. 5,778,791 .	JEAN R. HOMERE PRIMARY EXAMINER Jean R. Homere 10/24/01 (Primary Examiner) (Date)			ISSUE FEE Amount Due 640.00 Date Paid 11-21-01	
<input type="checkbox"/> c) The terminal _____ months of this patent have been disclaimed.	_____ (Legal Instruments Examiner) (Date)			ISSUE BATCH NUMBER 299	

WARNING:
The information disclosed herein may be restricted. Unauthorized disclosure may be prohibited by the United States Code Title 35, Sections 122, 181 and 368. Possession outside the U.S. Patent & Trademark Office is restricted to authorized employees and contractors only.

Form PTO-436A
(Rev. 6/98)

Issue Fee
(LABEL AREA)

PATENT APPLICATION



09283160

10:518 U.S. PTO
79/283160



04/01/99

INITIALS

(4)

CONTENTS

	Date received (Incl. C. of M.) or Date Mailed	Date received (Incl. C. of M.) or Date Mailed
1. Application _____ papers.		
2. <u>Re Amdt A</u>	<u>4/01/89</u>	
3. <u>Re Amdt B</u>	<u>4/11/01</u>	
4. <u>Petition Infringe</u>	<u>3-02-01</u>	
5. <u>Petition Directed</u>	<u>4-13-01</u>	
6. <u>IDS</u>	<u>4/18/01</u>	
7. <u>Rej (3 mos)</u>	<u>6/5/01</u>	
8. <u>Amtd C</u>	<u>8/22/01</u>	
9. <u>IDS</u>	<u>8/22/01</u>	
10. <u>Terminal Disclaimer</u>	<u>10-9-01</u>	
11. <u>Supp. Amtd D</u>	<u>10/10/01</u>	
12. <u>Drawing Corrections</u>	<u>10/10/01</u>	
13. <u>Allowance</u>	<u>10-25-01</u>	
14. <u>Communication</u>	<u>11/21/01</u>	
15. <u>Format Drawing</u>	<u>11/21/01</u>	
16. <u>Miss Office Action</u>	<u>1-25-02</u>	
17. <u>Drawings (3 sheets)</u>	<u>11/21/01</u>	
18. <u>Letter Re IDS</u>	<u>2/13/02</u>	
19. _____		
20. _____		
21. _____		
22. _____		
23. _____		
24. _____		
25. _____		
26. _____		
27. _____		
28. _____		
29. _____		
30. _____		
31. _____		
32. _____		
33. _____		
34. _____		
35. _____		
36. _____		
37. _____		
38. _____		
39. _____		
40. _____		
41. _____		
42. _____		
43. _____		
44. _____		
45. _____		
46. _____		
47. _____		
48. _____		
49. _____		
50. _____		
51. _____		
52. _____		
53. _____		
54. _____		
55. _____		
56. _____		
57. _____		
58. _____		
59. _____		
60. _____		
61. _____		
62. _____		
63. _____		
64. _____		
65. _____		
66. _____		
67. _____		
68. _____		
69. _____		
70. _____		
71. _____		
72. _____		
73. _____		
74. _____		
75. _____		
76. _____		
77. _____		
78. _____		
79. _____		
80. _____		
81. _____		
82. _____		

POSITION	INITIALS	ID NO.	DATE
FEE DETERMINATION	AH	72192	2/12/99
O.I.P.E. CLASSIFIER			
FORMALITY REVIEW	EA	59380	4-20-99

INDEX OF CLAIMS

✓ Rejected N Non-elected
 = Allowed I Interference
 - (Through numeral)... Canceled A Appeal
 + Restricted O Objected

Claim	Date
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	
15	
16	
17	
18	
19	
20	
21	
22	
23	
24	
25	
26	
27	
28	
29	
30	
31	
32	
33	
34	
35	
36	
37	
38	
39	
40	
41	
42	
43	
44	
45	
46	
47	
48	
49	
50	

Claim	Date
51	
52	
53	
54	
55	
56	
57	
58	
59	
60	
61	
62	
63	
64	
65	
66	
67	
68	
69	
70	
71	
72	
73	
74	
75	
76	
77	
78	
79	
80	
81	
82	
83	
84	
85	
86	
87	
88	
89	
90	
91	
92	
93	
94	
95	
96	
97	
98	
99	
100	

Claim	Date
99	
100	
101	
102	
103	
104	
105	
106	
107	
108	
109	
110	
111	
112	
113	
114	
115	
116	
117	
118	
119	
120	
121	
122	
123	
124	
125	
126	
127	
128	
129	
130	
131	
132	
133	
134	
135	
136	
137	
138	
139	
140	
141	
142	
143	
144	
145	
146	
147	
148	
149	
150	

If more than 150 claims or 10 actions
staple additional sheet here

RACK-1002
Page 3 of 351

(LEFT INSIDE)

PATENT APPLICATION SERIAL NO. _____

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

04/12/1999 AHAYES 00000010 09283160
01 FC:201 380.00 DP

*MDH
0-0-00*

PTO-1556
(5/87)

*U.S. GPO: 1998-433-214/80404

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
REQUEST FOR FILING
(RULE 53(b)(1))

For Design or Utility Applications

Rule 53(b)(1) PATENT APPLICATION:

- Continuation)
- Divisional) application under 37 CFR 1.53(b)(1)

of pending prior application of

Group Art Unit: 2776

Examiner: Homere, J.

Inventor(s): FARBER et al.

Parent Appln. No.:	08	960,079
	Series Code ↑	↑ Serial No.

Atty. Dkt. PM 252465

Parent Filed: October 24, 1997

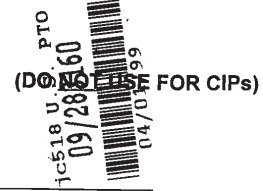
This Case Filed: April, 1999

Title: IDENTIFYING DATA IN A DATA PROCESSING SYSTEM

(Our Deposit Account No. 03-3975)

(Our Order No. 7018/252465)

C# / New M#



Asst. Commissioner of Patents
Washington, DC 20231

Date: April 1, 1999

(Parent Matter No. 243063)

Sir:

To effect the above-requested filing today:

- Attached** is a copy (**which must be filed**) of this application, including:

- Abstract
- Specification and claims (94 pages) (**must be attached**)
- Drawings (**must be attached if originally filed**): 24 sheet(s)/set: 1 set informal; Formal of size A4 11"

1A. **Always X one box, only:**

- Signed** declaration or oath as originally filed in prior application **attached**
- NO** declaration or fee is enclosed; therefore, this is a filing under Rule 53(f).

- This application is hereby filed by **less than all of the inventors** named in the prior application. Petition is hereby made requesting deletion as inventor(s) of the following who is/are **not** inventor(s) of the invention being claimed in this application:

- | | |
|----------|-----------|
| 1. _____ | 2. _____ |
| 3. _____ | 4. _____ |
| 5. _____ | 6. _____ |
| 7. _____ | 8. _____ |
| 9. _____ | 10. _____ |

- The entire disclosure of the prior application is considered as being part of the disclosure of the accompanying application and is hereby incorporated therein by reference thereto.

4. Priority is claimed under 35 U.S.C. 119/365 based on filing in _____ of _____ (country)
- | | | | | | |
|-----|------------------------|--------------------|-----|------------------------|--------------------|
| | <u>Application No.</u> | <u>Filing Date</u> | | <u>Application No.</u> | <u>Filing Date</u> |
| (1) | _____ | _____ | (4) | _____ | _____ |
| (2) | _____ | _____ | (5) | _____ | _____ |
| (3) | _____ | _____ | (6) | _____ | _____ |

- a. _____ (No.) Certified copy/copies attached.
 b. Certified copy/copies previously filed on _____ in _____ U.S. Application No. _____ / _____, filed on _____ series code ↑ serial no.

4. (a) Domestic priority is claimed from PCT/ _____ / _____, filed _____
 (b) Benefit is claimed of Provisional Application No. 60/_____, filed _____.

5. Prior application is assigned to kiNETech, Inc.
 by assignment recorded June 23, 1995 (Date) Reel 7593 Frame 0036.

6. Attached is the following number of Assignments (including original and all later successive ones by different assignors): 1 and respective **new** Cover Sheets. (Do **NOT** file old cover sheets.)
 (Assignments in parent **must be refiled** with new Cover Sheets in this continuing application if you want it/them recorded against the continuing application.)

Please return the recorded Assignment to the undersigned.

7. The power of attorney in the prior application is to Dale S. Lazar, Reg. No. 28,872
 (Name and Reg. No.)
 whose current address is as in item 8 below.

- a. Recognize as associate attorney Brian Siritzky, Reg. No. 37,497
 (Name, Reg. No. and Address)

8. **Address all future communications to Intellectual Property Group of Pillsbury Madison & Sutro LLP, Ninth Floor, East Tower 1100 New York Avenue, N.W., Washington, D.C. 20005-3918**

9. Amend the specification by inserting before the first line the sentence:--This is a continuation division of Application No. 08/960,079, filed October 24, 1997 series code ↑ serial no.
 which is a continuation of 08/425,160, filed April 11, 1995, now abandoned.

9. (a) Amend the specification by inserting before the first line: --This application claims the benefit of Provisional Application No. 60/_____, filed _____.

10. It has been recently determined that this new continuing application is entitled to small entity status. Hence:
 _____ (No.) Verified Statement(s) establishing "small entity" status under Rules 9 & 27 were/are:
 filed in above prior application (and hence applicable hereto)
 attached.

11. Petition to extend the life of the above prior application to at least the date hereof
 (one box) is being concurrently filed in that prior application (Use Form PAT-111).
 (must be) was previously filed in that prior application (Check length of prior extension).
 (X'd) is not necessary for copendency (Double check before X'ing this box).

- 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 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.
- 13. Attached is a Rule 103(a) Petition to Suspend Action.
- 14. **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.

FILING FEE

THE FOLLOWING FILING FEE IS BASED ON

-->>>>CLAIMS AS FILED AND CHANGED BY PRELIMINARY AMENDMENT IN ITEM 14<<<<<<

NOTE: If box 1A₂ is X'd, do not pay fees, but leave lines 15-22 and 27-32 blank.

				Large/Small Entity		Fee Code
15. Basic Filing Fee Design Application				\$310/\$155		106/26
16. Basic Filing Fee Not Design Application				\$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 <u>any proper</u> multiple dependent claim (ignore improper) is present,				\$260/\$130	+0	104/204
20. Subtotal =				\$380		
21. If "petition" box 13 above is X'd, add petition fee. \$130					+0	122
21A. If box 6 above is X'd, add Assignment recording fee \$ 40					+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.)
- 25. The following PRELIMINARY AMENDMENT is to be entered after assigning Appln. No.:

26.

**ADDITIONAL FEE CALCULATION FOR
PRELIMINARY AMENDMENT
PER BOXES 24/25**

	Claims remaining after amendment	Highest number previously paid for	Present Extra	Additional Fee	File Code
				<u>Large/Small Entity</u>	
27.	Total Effective Claims *	minus ** 20	= 0	x \$18/\$9 = \$ 0	(103/203)
28.	Independent Claims *	minus *** 3	= 0	x \$78/\$39 = + 0	(102/202)
29.	If amendment enters proper multiple dependent claim(s) into this application for the first time, add (per application) \$260/\$130			+ 0	(104/204)
30.	ADDITIONAL FEE			\$ 0	
31.	plus 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 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

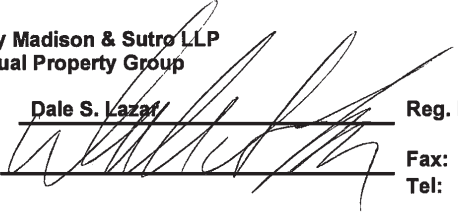
CHARGE STATEMENT: Upon the filing of a Declaration pursuant to Rule 60(b) or 60(d), the Commissioner is hereby authorized to charge any fee specifically authorized hereafter, or any missing or insufficient fee(s) filed, or asserted to be filed, or which should have been filed herewith or concerning any 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 duplicate 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.

**Pillsbury Madison & Sutro LLP
Intellectual Property Group**

1100 New York Avenue, N.W.
Ninth Floor, East Tower
Washington, D.C. 20005-3918
Tel: (202) 861-3000
DSL/BS:kim
Atty./Sec.

By Atty: Dale S. Lazar Reg. No. 28872

Sig:  Fax: (202) 822-0944
Tel: (202) 861-3527

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

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

This is a:

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

SPECIFICATION

CDC-100 3/95

0924260-0912260

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

5 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.

10 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.

15 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.

20
25
30 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
35 pathname can be determined only when the file system (context) is known. Similarly, the addresses in a

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.

5 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.

10 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.

15 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.

20 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

4

05100005100000

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).

5 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
10 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
15 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
20 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
25 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
30 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
35 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.

001010-0912260

5 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
10 data item.

15 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

20 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.

25 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 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
35 database, or the like that stores a plurality of data
items, by making possible or improving the design and

6

092220 092220

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;

5

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

the system can verify that data retrieved from another location is the desired 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;

15

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.

20

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.

25

30

BRIEF DESCRIPTION OF THE DRAWINGS

DI> 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 stored at any location in such a data processing system;

35

S

067040-0376260

The processes and mechanisms (services) provided in this embodiment are grouped into the following categories: primitive mechanisms, operating system mechanisms, remote mechanisms, background mechanisms, and extended mechanisms.

Primitive mechanisms provide fundamental capabilities used to support other mechanisms. The following primitive mechanisms are described:

1. Calculate True Name;
2. Assimilate Data Item;
3. New True File;
4. Get True Name from Path;
5. Link path to True Name;
6. Realize True File from Location;
7. Locate Remote File;
8. Make True File Local;
9. Create Scratch File;
10. Freeze Directory;
11. Expand Frozen Directory;
12. Delete True File;
13. Process Audit File Entry;
14. Begin Grooming;
15. Select For Removal; and
16. End Grooming.

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, and generally transparent to, existing applications. The following operating system mechanisms are described:

1. Open File;
2. Close File;
3. Read File;
4. Write File;
5. Delete File or Directory;

12

6. Copy File or Directory;
7. Move File or Directory;
8. Get File Status; and
9. Get Files in Directory.

5 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. The following remote mechanisms are described:

- 10 1. Locate True File;
2. Reserve True File;
3. Request True File;
4. Retire True File;
- 15 5. Cancel Reservation;
6. Acquire True File;
7. Lock Cache;
8. Update Cache; and
9. Check Expiration Date.

20 Background mechanisms are intended to run occasionally and at a low priority. These provide automated management capabilities with respect to the present invention. The following background mechanisms are described:

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

30 Extended mechanisms run within application programs over the operating system. These mechanisms provide solutions to specific problems and applications. The following extended mechanisms are described:

- 35 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.

5 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
10 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
15 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
20 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
25 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
performed in the background.

Data Structures

30 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.

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 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

09202450 09202450

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

5 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.

10 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.

15 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.

20 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

25 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.

TC/160

30

16

667010 0976800

Field	Description
Type	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.
5 Time of last access	the last access time to this file. If this file is a directory, this is the last access time to any file in the directory.
Time of last modification	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.
10 Safe flag	indicates that this file (and, if this file 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 processor 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.

15 Each record of the True File registry 126 has 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
20 True Name	computed True Name or identity of the file.

TOPIC

17

06F010" 09F0260

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.
5 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.
10 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.

15 A region table 128, specified by a directory pathname, records storage policies which allow files in 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 (with reference to FIGURE 5):

15

True

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 file 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 processors 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.

5

667040-09FEB60

10

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

19

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:

09FEB60

7020

Field	Description
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
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 availability	measurement of the bandwidth, cost, and reliability of the connection to this source of True Files. The availability is used to select from among several possible sources.
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):

09FEB60

7020

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

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.

5 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

10 FIGURE 8:

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.

15

20 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.

667490 0912250

TC 210

TC 211

667010-0972260

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

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;

22

SECRET

changes to B are virtually guaranteed to produce a different value of MD(B).

4. It must be computationally difficult to find a different value B' such that MD(B)=MD(B').
5. 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 authentication 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.

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

24

097E260

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

26

considered desirable features in the preferred embodiment.

2. Assimilate Data Item

5 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
reference to FIGURE 11. The purpose of this mechanism is
to add a given data item to the True File registry 126.
10 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.

15 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
20 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
25 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
30 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.

27

SECRET 031260

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.

5 3. New True File

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:

15 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).

20 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 4. Get True Name from Path

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.

30 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
35 pathname exists. Next, determine whether the local

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 (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. 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.)

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

30

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

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. Locate Remote File

^{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 follows (with reference to FIGURE 16).

The client processor 102 uses the source table 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 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 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 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

31

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
5 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
10 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.

15 9. Create Scratch File

^{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
20 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
25 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
30 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
35 mechanism (Step S318). Then create a new, empty scratch file and store its scratch file ID in the local directory

33

extensions table entry record 138 (Step S320). This mechanism is then complete.

5 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.

10 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.

15 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
20 mechanism's task is complete.

25 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.

30 10. Freeze Directory
Do 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
35 the pathname of a directory to freeze. This mechanism is described with reference to FIGURE 19.

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

5 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 10 S369), and record the newly computed True Name for the file in the audit file record entry (Step S370).

15 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).

20 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).

25 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 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 35 zero and empty the list of files selected for deletion

37

001010-0372260

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

15. Select For Removal

5 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
10 corresponding to the True File name in the local
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
15 grooming delete count of the True File registry entry
record 140 is equal to the use count of the True File
registry entry record 140, and if there are no
entries in the dependency list of the True File registry
entry record 140, then add the size of the file indicated
20 by the True File ID and or compressed file ID to the
total amount of space freed during grooming (Step S394).

16. End Grooming

This grooming mechanism ends the grooming phase
and removes all files selected for removal. With
25 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

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

The following operating system mechanisms are
described:

1. Open File;

007010-097000

with the longest region path which is a prefix of the file pathname (Step S408). This record identifies the region of the specified file.

5 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
10 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).

15 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).

20 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
25 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
30 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. Close File

35 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

4/c

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 6. Copy File or Directory

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.

10 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.

15 (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
20 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
25 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
30 mechanisms for groups of files.

7. Move File or Directory

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
35 actual transfer of data, and is performed as follows:

(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.

5 8. Get File Status

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.

9. Get Files in Directory

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

44

to expand the existing True File into directory entries in the local directory extensions table.

5 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

10 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.

15 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:

- 25
1. Locate True File;
 2. Reserve True File;
 3. Request True File;
 4. Retire True File;
 5. Cancel Reservation;
 6. Acquire True File;
 7. Lock Cache;
 - 30 8. Update Cache; and
 9. Check Expiration Date.

1. Locate True File

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

(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. Reserve True File

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 record 140 associated with the given True File. If no entry exists, 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.

3. Request True File

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.

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. Acquire True File

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.

(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.

(F) Send a positive reply.

49

001010-0912260

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. Update Cache

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. Check Expiration Date

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

52

2. Groom Region

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 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.

(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 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

recorded at the remote processor identified by the source.

Extended Mechanisms

5 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.

10 The following extended mechanisms are described:

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

20 1. 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 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.

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

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

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.

2. Inventory Removable, Read-only Files

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.

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.

001010-0918260

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 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 10 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.

15 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.

20 (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 25 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 30 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 35 mechanism.

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

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.

5

3. Synchronize Directories

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.

10

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.

15

20

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.

25

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.

30

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:

35

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. Retire Directory

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.

(ii) Determine an alternate source for the True File. If the source 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.

(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 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. Verify True File

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 problems.

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

61

CONFIDENTIAL

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.

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:

(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.

(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.

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

62

09122500

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

5 (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
10 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:

15 (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.

20 (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

25 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.

30 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
35 the data items and only on the data in the data items.

64

067070-0978260

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
5 for that data item can be determined.

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 fault-tolerant system). Multiple copies of data items are
10 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
15 already exists in the system, as indicated by an entry in 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
20 system from a floppy disk, if, based on the True Name of 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
25 name other than its current name, then, using the Link 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
30 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
35 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

45

667070-0122250

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 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 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.

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

66

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

5 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.

15 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.

20 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
25 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
30 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

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.

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

68

001010" 09 FEB 60

put on notice that another location is relying on the presence of the True File at their location.

5 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
10 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
15 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.

20 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
25 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
30 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
35 determined either from the Safe Flag or the source IDs, or by checking which other processors in the system have

69

SECRET

5 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.)

10 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.
15 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.

20 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
25 Synchronize Directories extended mechanism on the needed directories.

30 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
35 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

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 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 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 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, 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 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 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

72

00101010 09122660

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.

The Groom Cache background mechanism flushes the cache, removing the least-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 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

73

0928280 040409

5 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.

10 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.

15 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.

20 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).

25 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.

30 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.

35 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

75

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.

5 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
10 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

15 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
20 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
25 location-independent access to records, maintain archives and histories of records, and synchronize with distant or disconnected systems or databases.

30 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.)

007010 097820

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
5 the key of the record to its contents by using the True Name as a pointer to the contents.

The mechanisms described in the preferred embodiment, or similar mechanisms, would be employed in such a system. These mechanisms could include, for
10 example, the mechanisms for calculating true names, assimilating, locating, realizing, deleting, copying, and moving True Files, for mirroring True Files, for maintaining a cache of True Files, for grooming True
15 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
20 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.

667010 03E260

277

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.

5 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:

15 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;

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

25 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 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

0944260

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 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

20 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.

SECRET

5 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:

10 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
15 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
20 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. An apparatus as in claim 3, wherein at least some of said data items are compound data items,
25 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
30 the compound data item.

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

00000-050000

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
5 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 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
15 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
20 comprising:
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
25 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
30 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.

SECRET

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

5 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:

10 for each data item of said component data items,

15 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;

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

25 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:

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

(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

657010-0912200

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

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

5 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
10 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:

15 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 claim 43, further comprising the step of:

20 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
25 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 46. A method of eliminating a data item at a
given location in a data processing system when said data

05700-0512250

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

001070 0512200

5 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:

- 10 (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,
 - 15 (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
- 20 (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,
 - 25 (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.

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 B1

ADD C2

66FD4D 09TEEBDD

INS. C4

ABSTRACT OF THE DISCLOSURE

IDENTIFYING DATA IN A DATA PROCESSING SYSTEM

5 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.

001010-01E2E20

FOR UTILITY/DESIGN
CIP/PCT NATIONAL/PLANT
ORIGINAL/SUBSTITUTE/SUPPLEMENTAL
DECLARATIONS

RULE 63 (37 C.F.R. 1.63)
DECLARATION AND POWER OF ATTORNEY
FOR PATENT APPLICATION
IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

CUSHMAN
FORM

As a below named inventor, I hereby declare that my residence, post office address and citizenship are as stated below next to my name, and I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the INVENTION ENTITLED

IDENTIFYING DATA IN A DATA PROCESSING SYSTEM
the specification of which (CHECK applicable BOX(ES))
-> [] is attached hereto.
-> [x] was filed on April 11, 1995 as U.S. Application No. 0 8 / 425,160
BOX(ES) -> [] was filed as PCT International Application No. PCT/ / on
-> -> and (if applicable to U.S. or PCT application) was amended on

I hereby state that I have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment referred to above. I acknowledge the duty to disclose all information known to me to be material to patentability as defined in 37 C.F.R. 1.56. I hereby claim foreign priority benefits under 35 U.S.C. 119/365 of any foreign application(s) for patent or inventor's certificate listed below and have also identified below any foreign application for patent or inventor's certificate filed by me or my assignee disclosing the subject matter claimed in this application and having a filing date (1) before that of the application on which priority is claimed, or (2) if no priority claimed, before the filing date of this application:

PRIOR FOREIGN APPLICATION(S) Number	Country	Date first Laid- open or Published	Date Patented or Granted	Priority Claimed	
				Yes	No

I hereby claim the benefit under 35 U.S.C. 120/365 of all United States applications listed below and PCT international applications listed above or below and, if this is a continuation-in-part (CIP) application, insofar as the subject matter disclosed and claimed in this application is in addition to that disclosed in such prior applications, I acknowledge the duty to disclose all information known to me to be material to patentability as defined in 37 C.F.R. 1.56 which became available between the filing date of each such prior application and the national or PCT international filing date of this application:

PRIOR U.S. OR PCT APPLICATION(S) Application No. (series code/serial no.)	Date/MONTH/Year Filed	Status pending, abandoned, patented

I hereby 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 were 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 the application or any patent issued thereon.

And I hereby appoint Cushman Darby & Cushman, L.L.P., 1100 New York Avenue, N.W., Ninth Floor, East Tower Washington, D.C. 20005-3918, telephone number 861-3000 (to whom all communications are to be directed), and the below-named persons (of the same address) individually and collectively my attorneys to prosecute this application and to transact all business in the Patent and Trademark Office connected therewith and with the resulting patent. I hereby authorize them to act and rely on instructions from and communicate directly with the person/assignee/attorney/firm/ organization who/which first sends/sent this case to them and by whom/which I hereby declare that I have consented after full disclosure to be represented unless/until I instruct Cushman, Darby & Cushman in writing to the contrary.

Paul N. Kokulis	16773	Edward M. Prince	22429	Dale S. Lazar	28872	Michelle N. Lester	32331
Raymond F. Lippitt	17519	Donald B. Deaver	23048	Glenn J. Perry	28458	Jeffrey A. Simenauer	31933
G. Lloyd Knight	17698	David W. Brinkman	20817	Kendrew H. Colton	30368	Robert A. Molan	29834
Carl G. Love	18781	George M. Sirilla	18221	Chris Comuntzis	31097	G. Paul Edgell	24238
Edgar H. Martin	20534	Donald J. Bird	25323	Wallace G. Walter	27843	Lynn E. Eceleston	35861
William K. West, Jr.	22057	W. Warren Taltavull	25647	Lawrence Harbin	27644	Frederick S. Frei	27105
Kevin E. Joyce	20508	Reiter W. Gowdey	25872	Paul E. White, Jr.	32011	David A. Jakopin	32995
						Mark G. Paulson	30793

1. INVENTOR'S SIGNATURE: David A. Farber Date 6/7/95

Inventor's Name (typed) David A. FARBER U.S.A.

Residence (City) Ojai Middle Initial A. Family Name FARBER Country of Citizenship U.S.A.

Post Office Address (Include Zip Code) 2300 Carillo Rd, Ojai, CA 93023

2. INVENTOR'S SIGNATURE: Ronald D. Lachman Date 6/13/95

Inventor's Name (typed) Ronald D. LACHMAN U.S.A.

Residence (City) Northbrook Middle Initial D. Family Name LACHMAN Country of Citizenship U.S.A.

Post Office Address (Include Zip Code) 3140 Whisperwoods Court, Northbrook, Il. 60062

3. INVENTOR'S SIGNATURE: _____ Date _____

Inventor's Name (typed) _____

Residence (City) _____ Middle Initial _____ Family Name _____ Country of Citizenship _____

Post Office Address (Include Zip Code) _____

FOR 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 (Atty. Dkt.
Appl. No.: 0 8 /425,160 or Patent No.: (213987 /
Filed: April 11, 1995 or Issued: _____ M# / Client Ref.
Title: IDENTIFYING DATA IN A DATA PROCESSING SYSTEM

VERIFIED STATEMENT (DECLARATION) CLAIMING SMALL ENTITY
STATUS (37 CFR 1.9(d) and 1.27(c)) - **SMALL BUSINESS CONCERN**

I hereby declare that I am

- 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 KINETECH INC.

ADDRESS OF CONCERN 202N Carillo Rd., Olat, California 93023 DE 6/7/95
3140 Whisperwoods Ct., Northbrook Illinois 60062 (27) c/l

I hereby declare that the above identified small business concern qualifies as a small business concern as defined in 13 CFR 121.12, and reproduced in 37 CFR 1.9(d), for purposes of paying reduced fees under Section 41(a) and (b) of Title 35, United States Code, in that the number of employees of the concern, including those of its affiliates, does not exceed 500 persons. For purposes of this statement, (1) the number of employees of the business concern is the average over the previous fiscal year of the concern of the persons employed on a full-time, part-time or temporary basis during each of the pay periods of the fiscal year, and (2) concerns are affiliates of each other when either, directly or indirectly, one concern controls or has the power to control the other, or a third party or parties controls or has the power to control both.

I hereby declare that rights under contract or law have been conveyed to and remain with the small business concern identified above with regard to the invention entitled: IDENTIFYING DATA IN A DATA PROCESSING SYSTEM

by inventors(s) David A. FARBER and Ronald D. LACHMAN
described in

-> [] the Specification filed herewith,
 -> [X] Application No. 0 8 /425,160, filed April 11, 1995
 -> [] Patent No. _____, issued _____

If the rights held by the above identified small business concern are not exclusive, each small entity individual, concern or organization having rights to the invention is listed in (A) and (B) below and no rights to the invention are held by any person, other than the inventor, who could not qualify under 37 CFR 1.9(c) as an independent inventor if that person had made the invention, or by any concern which would not qualify as a small business concern under 37 CFR 1.9(d) or a nonprofit organization under 37 CFR 1.9(e).

(A) FULL NAME of assignee/licensee/grantee/conveyee* _____

ADDRESS _____

X proper box: [] INDIVIDUAL [] SMALL BUSINESS CONCERN [] NONPROFIT ORGANIZATION

(B) FULL NAME of assignee/licensee/grantee/conveyee* _____

ADDRESS _____

X proper box: [] INDIVIDUAL [] SMALL BUSINESS CONCERN [] NONPROFIT ORGANIZATION

*NOTE: Separate verified statement is required from each person, concern or organization named in (A) and (B) above having rights to the invention, averring to his/her/its status as a small entity. (37 CFR 1.27)

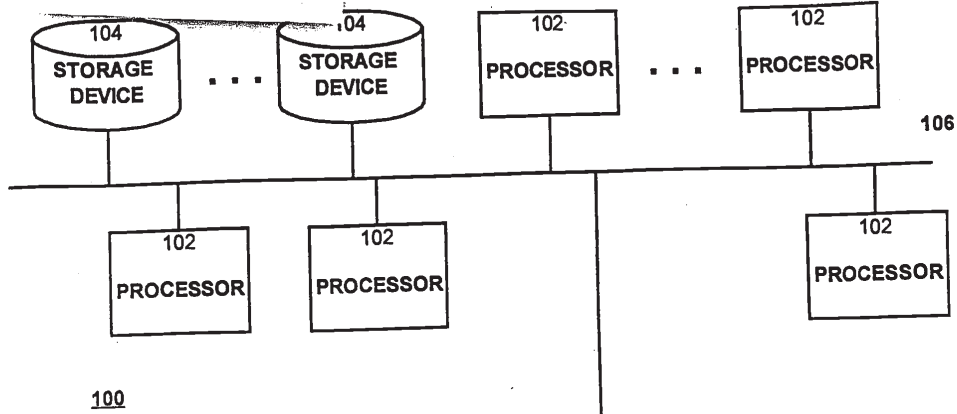
I acknowledge the duty to file, in this case, notification of any change in status resulting in loss of entitlement to small entity status prior to paying, or at the time of paying, the earliest of the issue fee or any maintenance fee due after the date on which status as a small entity is no longer appropriate. (37 CFR 1.28(b))

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were 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 the application, any patent issuing thereon, or any patent to which this verified statement is directed.

NAME OF PERSON SIGNING: Ronald Lachman
TITLE OF PERSON OTHER THAN OWNER: _____
ADDRESS OF PERSON SIGNING: 3140 WHISPERWOODS CT
NORTHBROOK IL 60062

SIGNATURE: [Signature] DATE: 6-14-95

PRINT OF DRAWINGS
AS ORIGINALLY FILE



100

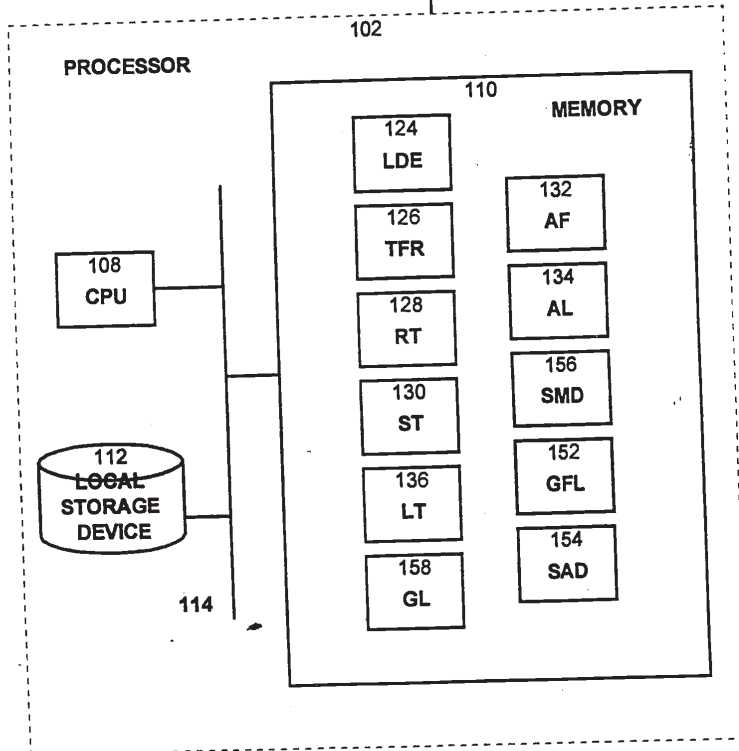
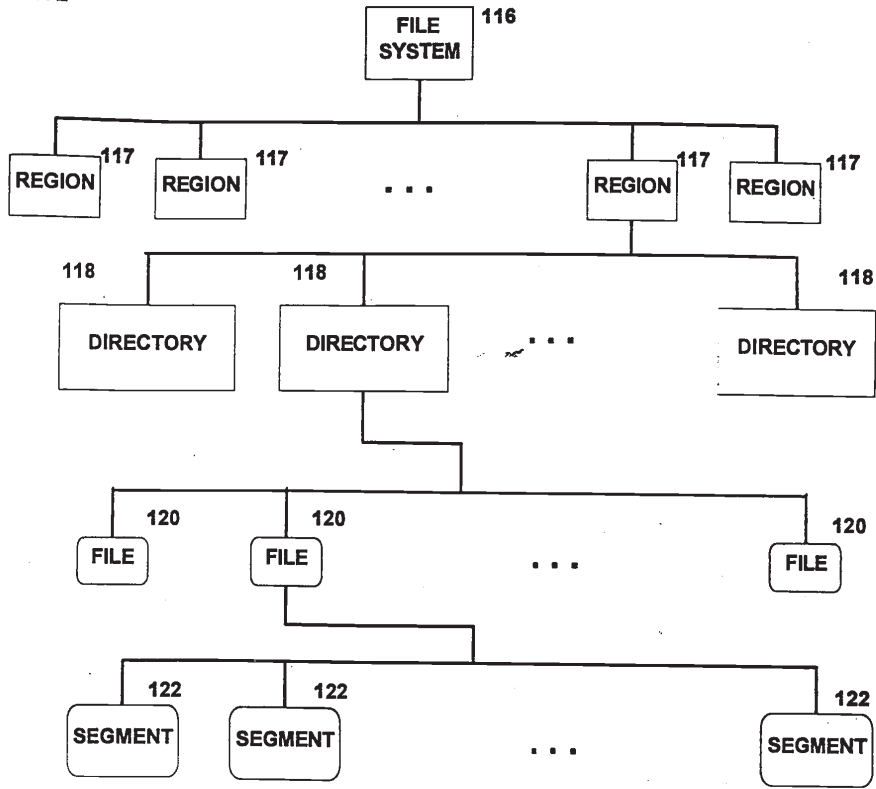


FIG. 1

5070707057000

PRINT OF DRAWINGS
AS ORIGINALLY FILE

FIG. 2



097000-097000

**PRINT OF DRAWINGS
AS ORIGINALLY FILED**

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

140

True Name
File ID
Compressed File ID
Source IDs
Dependent processors
Use count
Time of last access
Expiration
Grooming delete count

FIG. 4

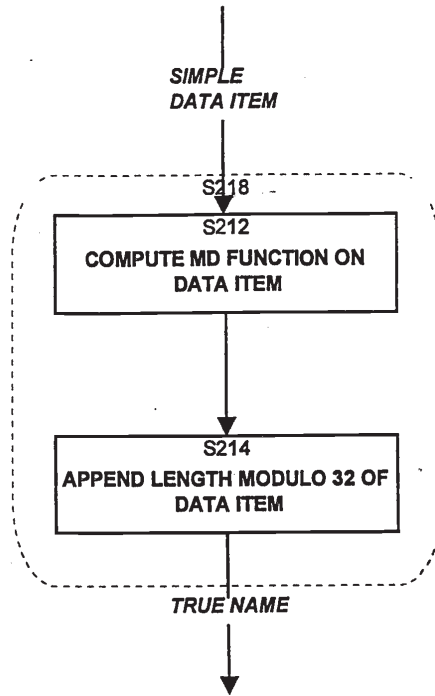
142

Region ID
Region file system
Region pathname
Region status
Mirror processor(s)
Mirror duplication count
Policy

FIG. 5

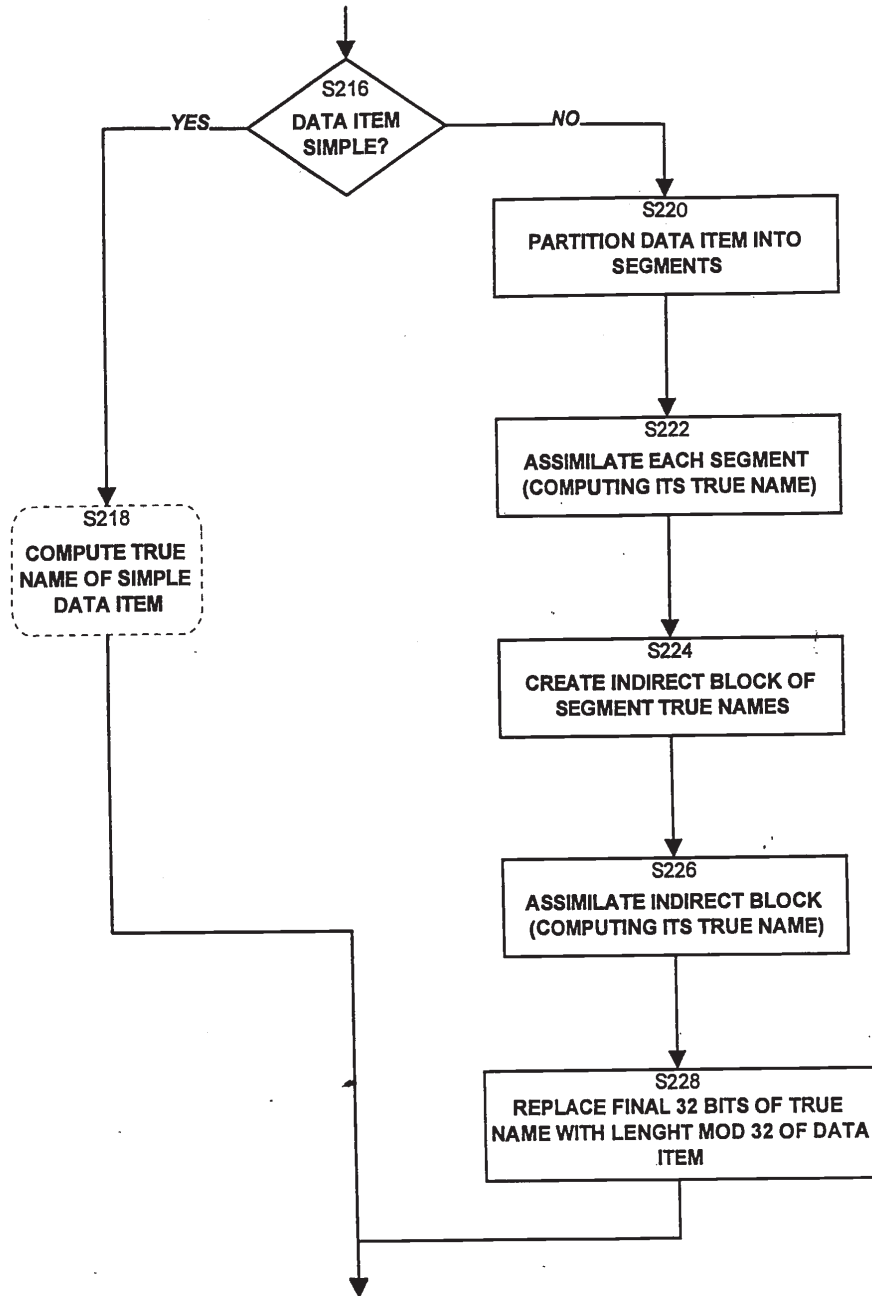
557070 03E2E60

FIG. 10(a)



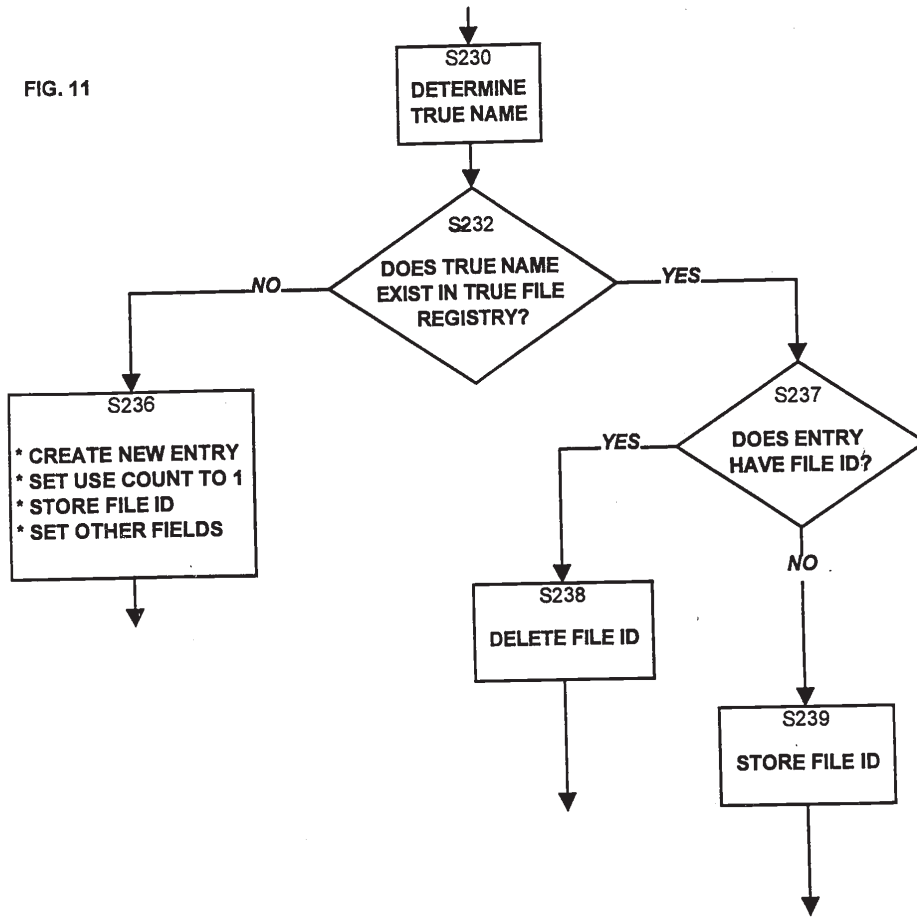
001010 091000

FIG. 10(b)



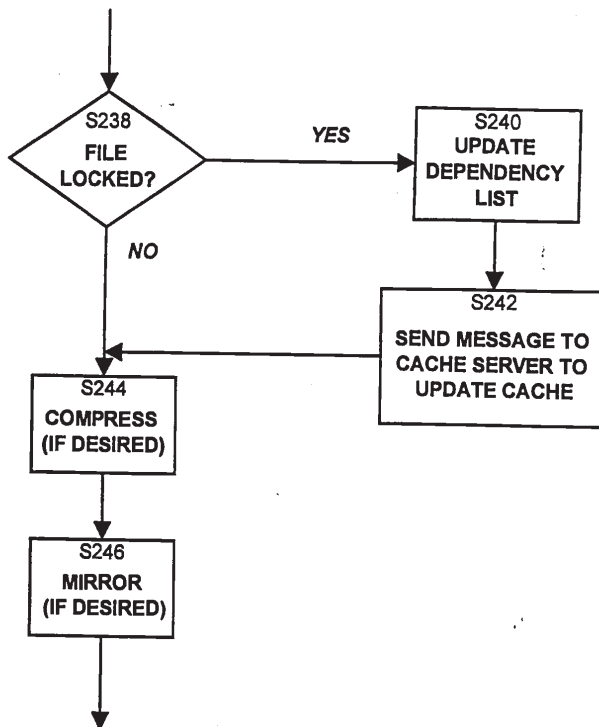
051040-091E0260

FIG. 11



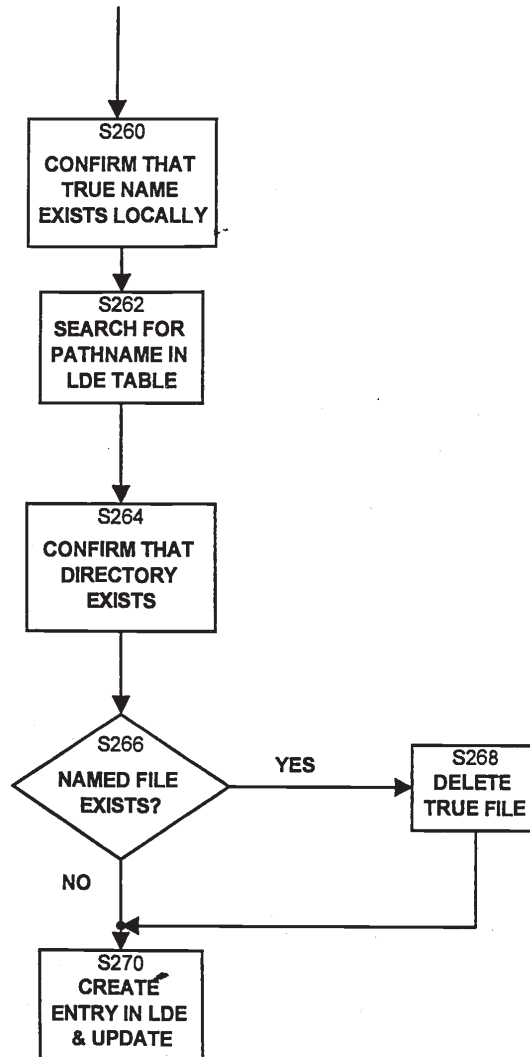
00100-091000

FIG. 12



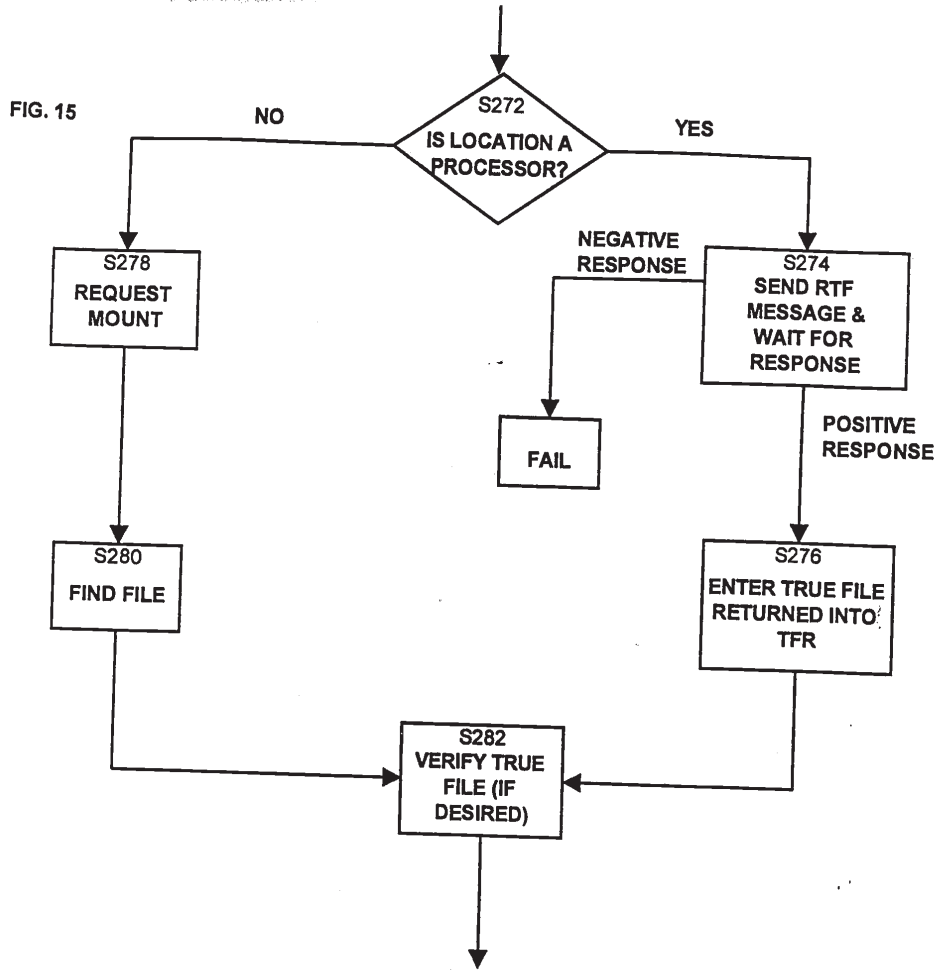
09122660

FIG. 14



66F010-09T0000

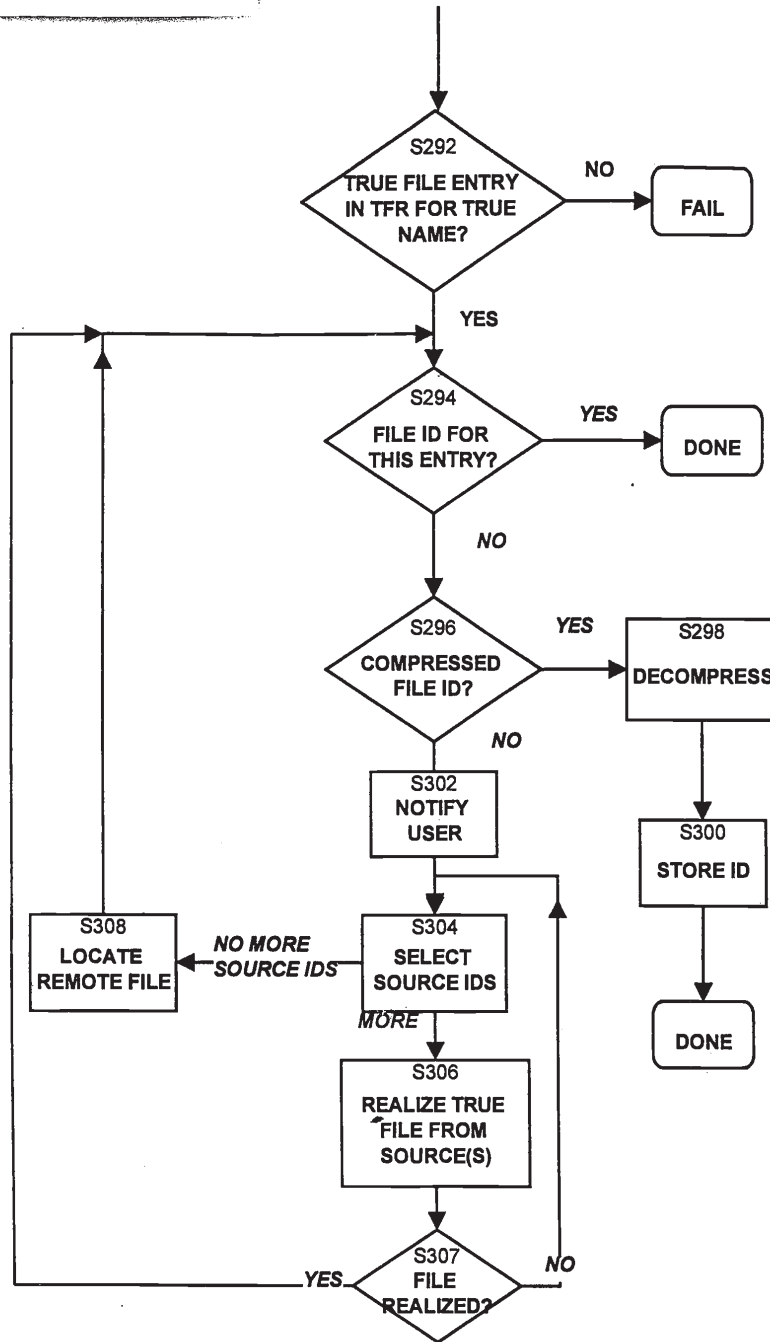
FIG. 15



067010-0976260

PRINT OF DRAWINGS
AS ORIGINALLY FILED

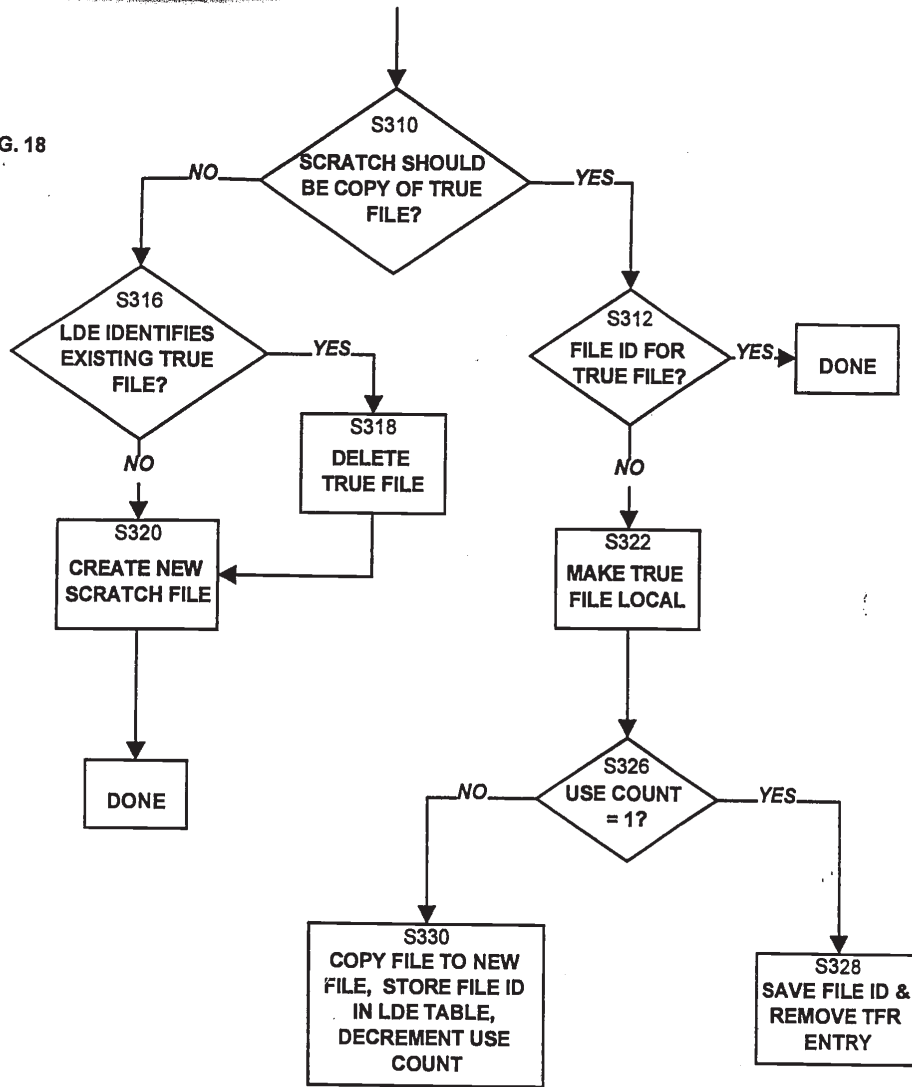
Fig. 17



657010 0912260

PRINT OF DRAWINGS
AS ORIGINALLY FILED

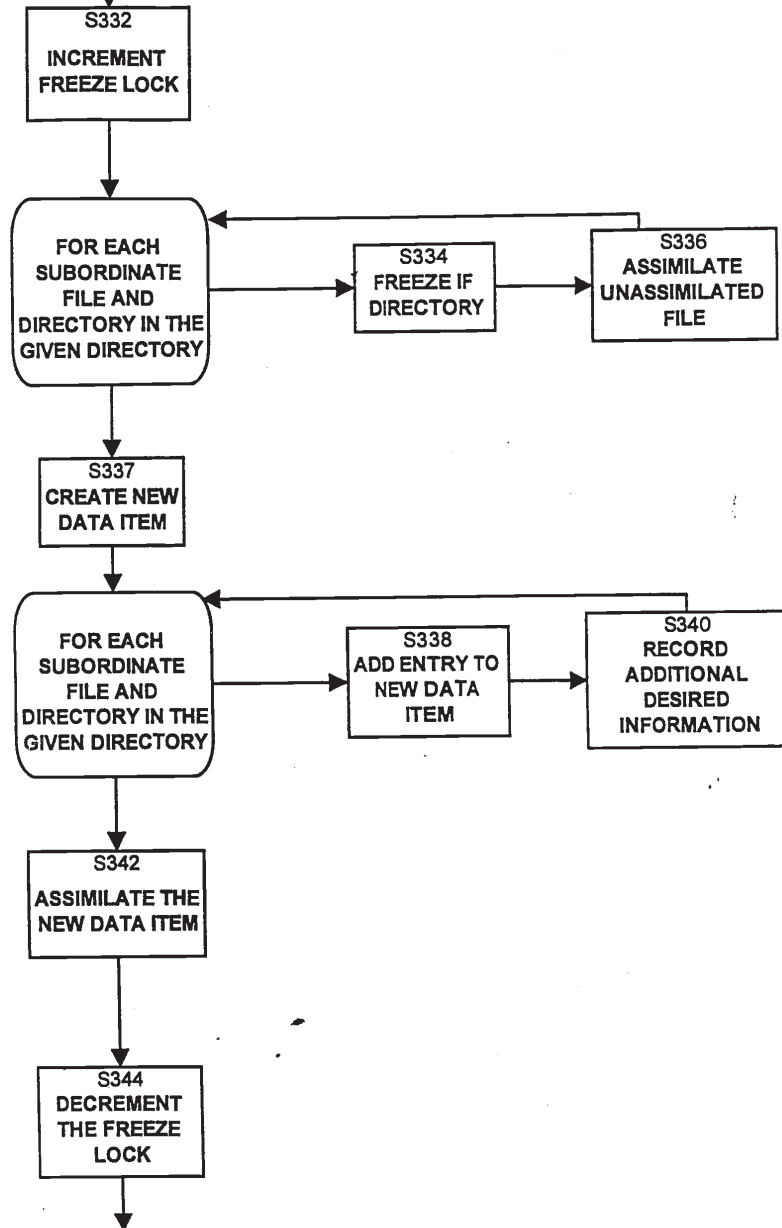
FIG. 18



000000-0976260

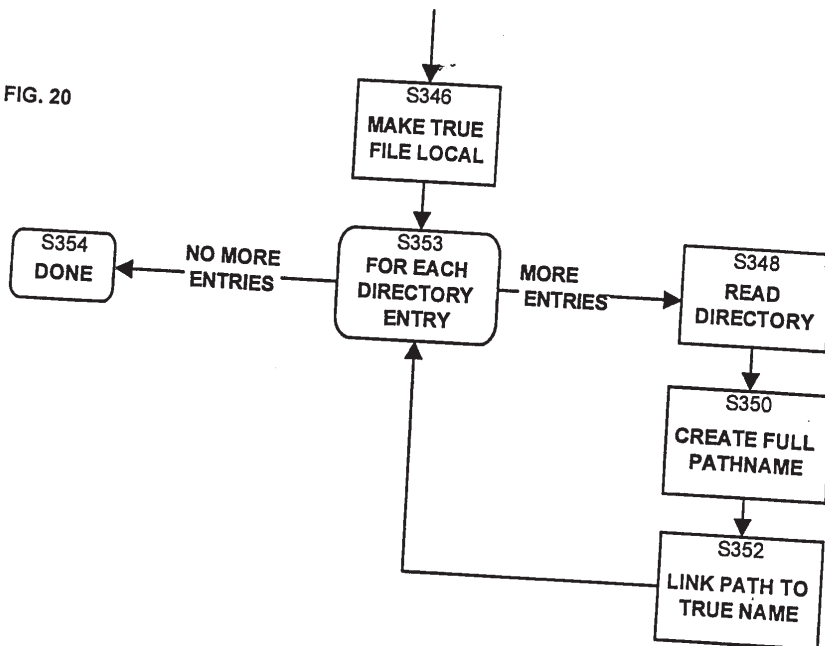
**PRINT OF DRAWINGS
AS ORIGINALLY FILE**

FIG. 19



0925150 0925150

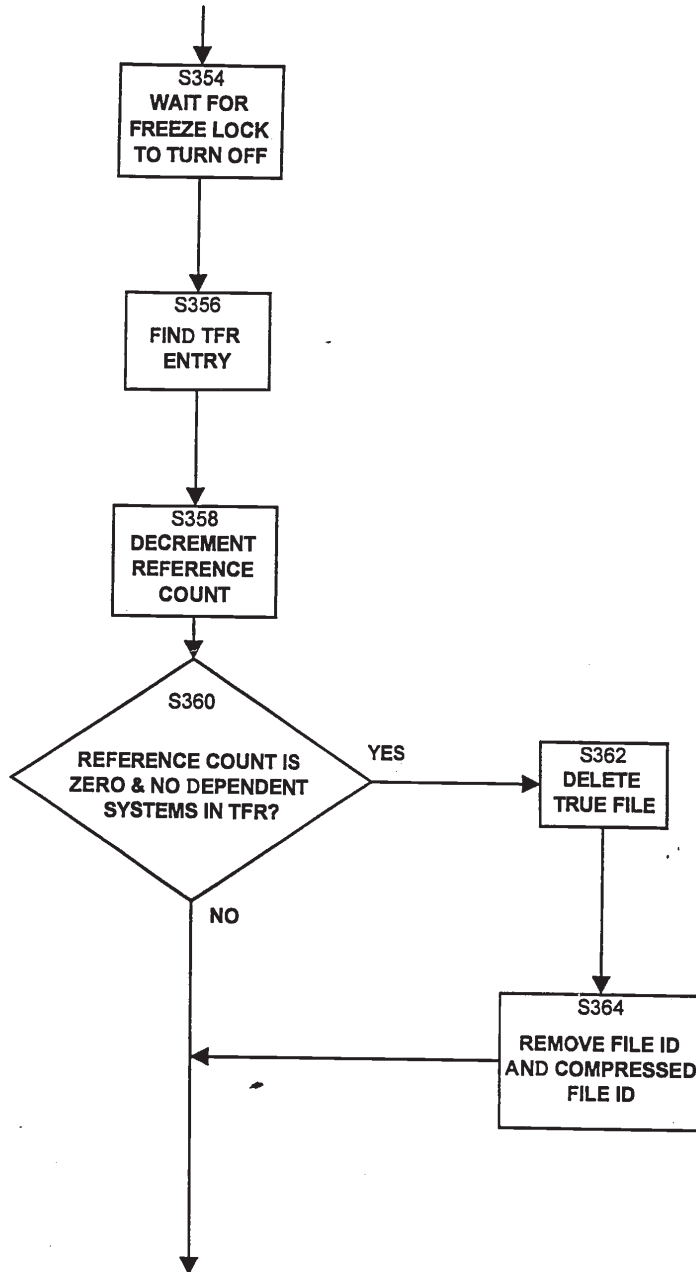
FIG. 20



067010-097000

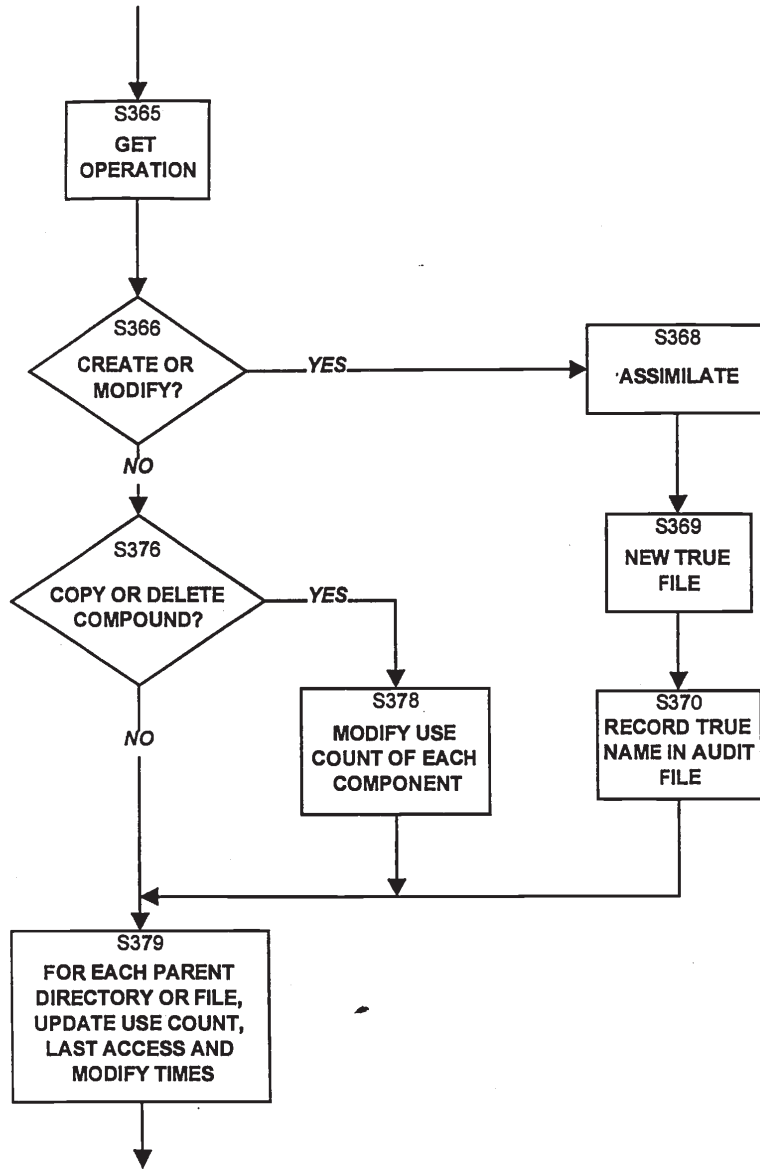
FIG. 21

667010-0976260



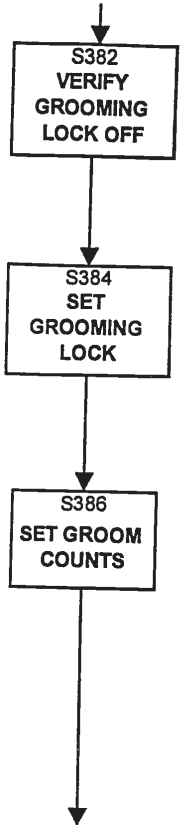
PRINT OF DRAWINGS
AS ORIGINALLY FIL

FIG. 22



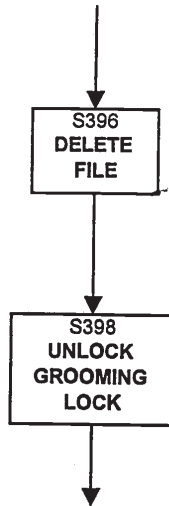
06F040-037E2E00

FIG. 23



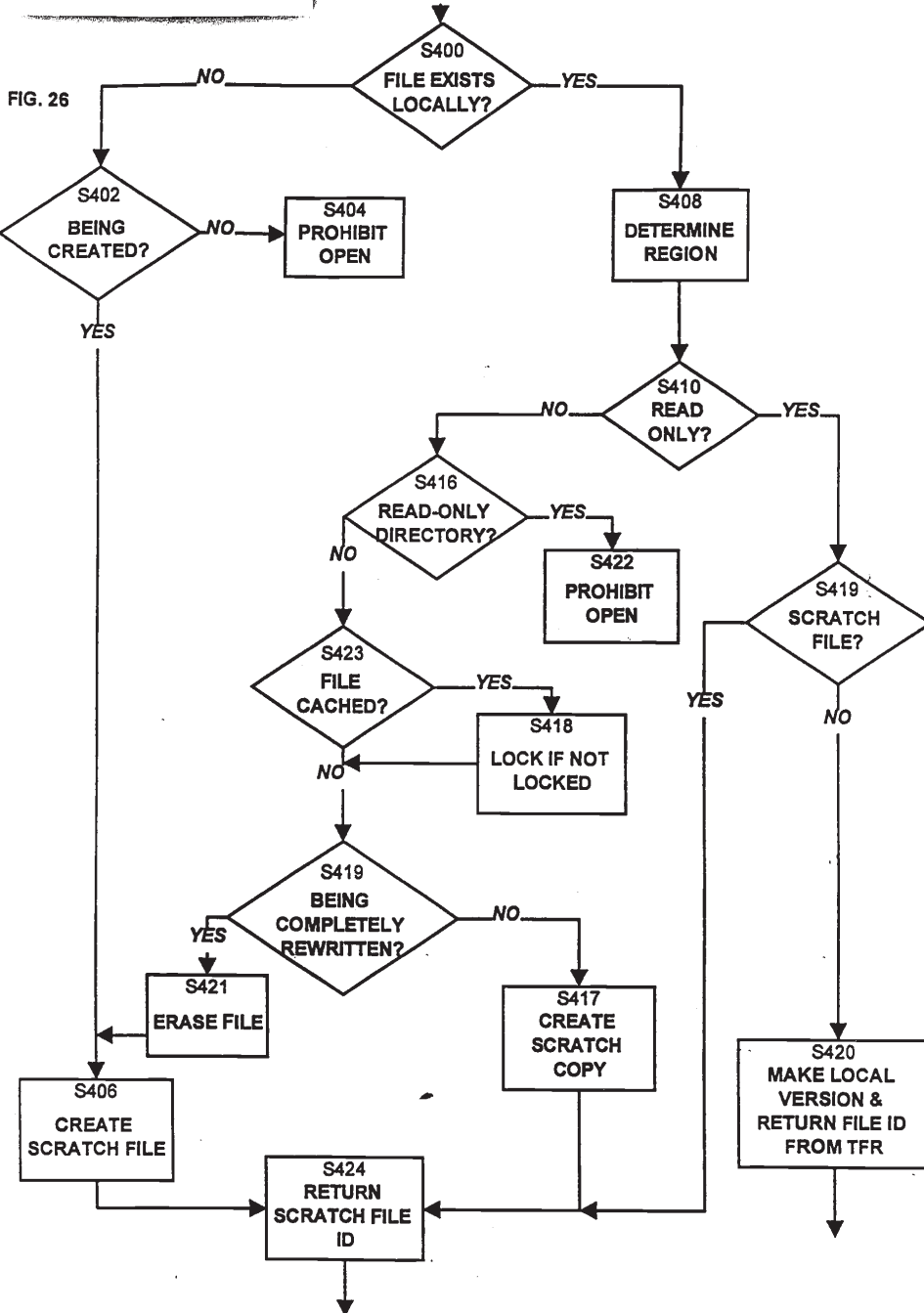
667070 09T E26260

FIG. 25



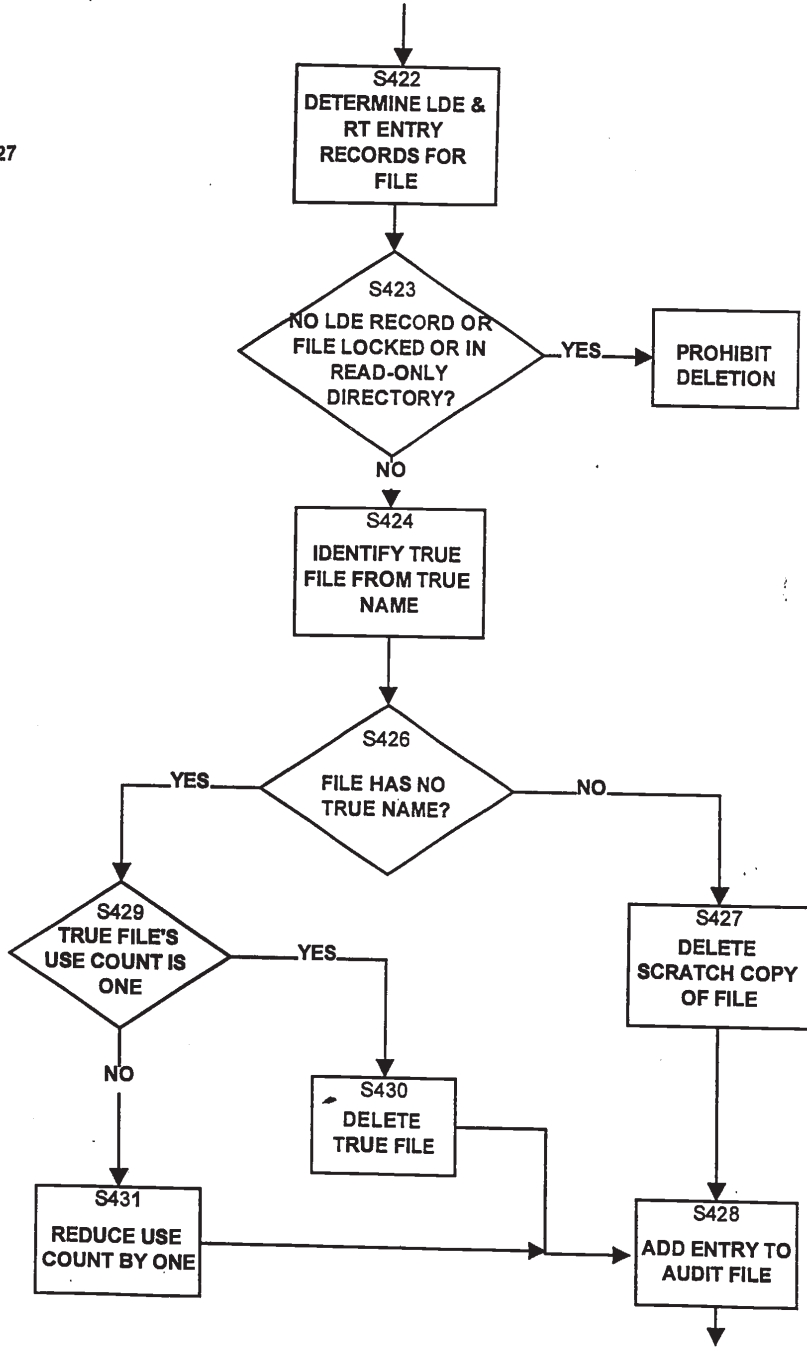
607010 09 FEB 60

PRINT OF DRAWINGS
AS ORIGINALLY FILE



6670707 097E6260

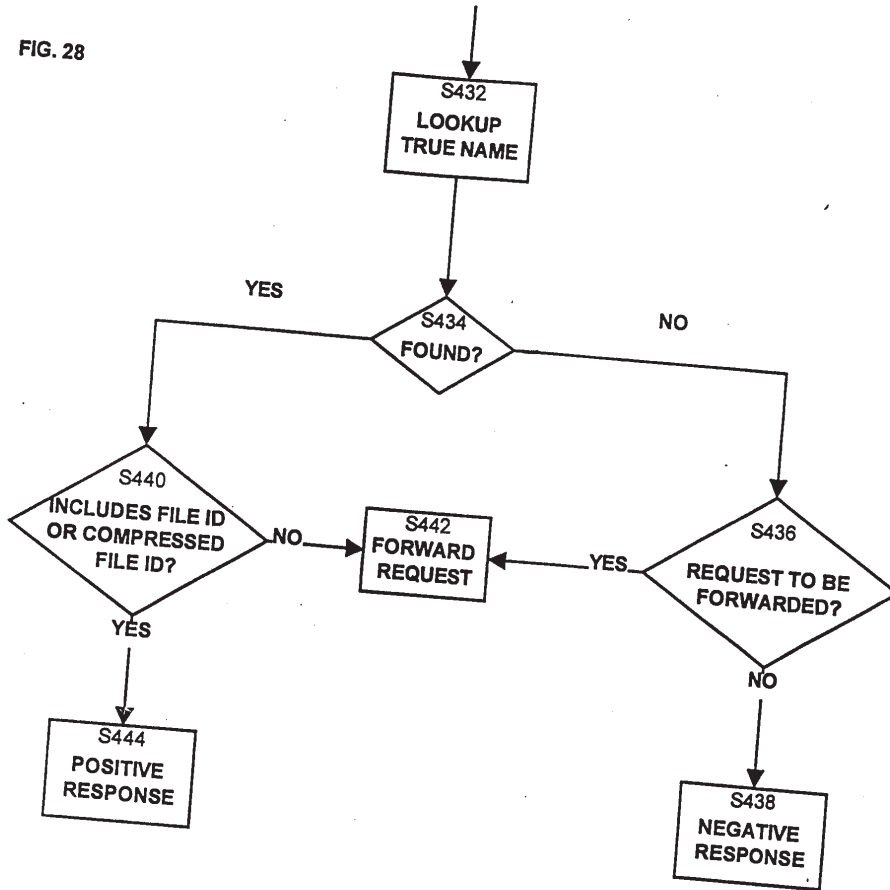
FIG. 27



66T040-09E8260

PRINT OF DRAWINGS
AS ORIGINALLY FIL

FIG. 28



000000-0916260

041-45 + 50-53

58-40

PATENT APPLICATION FEE DETERMINATION RECORD
Effective November 10, 1998

Application or Docket Number

09/28/93/1100 -

CLAIMS AS FILED - PART I

FOR	(Column 1) NUMBER FILED	(Column 2) NUMBER EXTRA
BASIC FEE		
TOTAL CLAIMS	5 minus 20 = *	Z
INDEPENDENT CLAIMS	3 minus 3 = *	
MULTIPLE DEPENDENT CLAIM PRESENT		

* If the difference in column 1 is less than zero, enter "0" in column 2

CLAIMS AS AMENDED - PART II

AMENDMENT A	(Column 1)	(Column 2)	(Column 3)
	CLAIMS REMAINING AFTER AMENDMENT	HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA
Total	*	Minus **	=
Independent	*	Minus ***	=
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM			

AMENDMENT B	(Column 1)	(Column 2)	(Column 3)
	CLAIMS REMAINING AFTER AMENDMENT	HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA
Total	*	Minus **	=
Independent	*	Minus ***	=
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM			

AMENDMENT C	(Column 1)	(Column 2)	(Column 3)
	CLAIMS REMAINING AFTER AMENDMENT	HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA
Total	*	Minus **	=
Independent	*	Minus ***	=
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM			

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.
 ** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20."
 *** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3."
 The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.

SMALL ENTITY TYPE <input type="checkbox"/>		OR	OTHER THAN SMALL ENTITY	
RATE	FEE		RATE	FEE
	380.00	OR		760.00
X\$ 9=		OR	X\$18=	
X39=		OR	X78=	
+130=		OR	+260=	
TOTAL	380	OR	TOTAL	

SMALL ENTITY		OR	OTHER THAN SMALL ENTITY	
RATE	ADDITIONAL FEE		RATE	ADDITIONAL FEE
X\$ 9=		OR	X\$18=	
X39=		OR	X78=	
+130=		OR	+260=	
TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

RATE	ADDITIONAL FEE		RATE	ADDITIONAL FEE
X\$ 9=		OR	X\$18=	
X39=		OR	X78=	
+130=		OR	+260=	
TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

RATE	ADDITIONAL FEE		RATE	ADDITIONAL FEE
X\$ 9=		OR	X\$18=	
X39=		OR	X78=	
+130=		OR	+260=	
TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

FORM PTO-1449 (modified)
 To: U.S. Department of Commerce
 (PM&S FORM PAT-1449)
 Patent and Trademark Office

T1/2

Atty. Dkt. No.	M#	Client Ref.
	252465	

**INFORMATION DISCLOSURE STATEMENT
 BY APPLICANT**

Applicant: FARBER et al.

Appl. No.: R.53(B)(1) Div. of 08/960,079

Filing Date: April 1, 1999

Examiner: Homere, J. Group Art Unit: 2776



Date: April 1, 1999 Page 1 of 3

U.S. PATENT DOCUMENTS

Examiner's Initials*	Document Number	Date MM/YYYY	Name (Family Name of First Inventor)	Class	Sub Class	Filing Date (if appropriate)
	AR 4,888,681	12/1989	BARNES			
	BR 4,972,367	11/1990	BURKE			
	CR 5,050,212	09/1991	DYSON			
	DR 5,202,982	04/1993	GRAMLICH			
	ER 5,208,858	05/1993	VOLLERT			
	FR 5,301,316	05/1994	HAMILTON			
	GR 5,343,527	08/1994	MOORE			
	HR 4,571,700	02/1996	EMRY, JR.			
	IR 4,675,810	06/1987	GRUNER			
	JR 5,050,074	09/1991	MARCA			
	KR 5,276,901	01/1994	HOWELL			
	LR 5,384,565	01/1995	CANNON			
	MR 5,202,982	04/1993	GRAMLICH			
	NR 5,357,623	10/1994	MEGORY-COHEN			

FOREIGN PATENT DOCUMENTS

	Document Number	Date MM/YYYY	Country	Inventor Name	Class	SubClass	English Abstract		Translation Readily Available	
							Enclosed	No	Enclose	No
	OR									
	PR									
	QR									
	RR									
	SR									
	TR									
	UR									

OTHER (Including in this order Author, Title, Periodical Name, Date, Pertinent Pages, etc.)

VR	Bert dem Boer et al., Collisions for the compression function of MD ₅ , pp. 292-304								
WR	Sakti Pramanik et al., Multi-Directory Hasing, 1993, Info. Sys., Vol. 18, No. 1, pp. 63-74								
XR	Murlidhar Koushik, Dynamic Hashing with Distributed Overflow Space: A File Organization with Good Insertion Performance, 1993, Info. Sys., Vol. 18, No. 5, pp. 299-317								
YR	Witold Litwin et al., LH*-Linear Hashing for Distributed Files, HP Labs Tech. Report No. HPL-93-21, June 1993, pp. 1-22								
ZR	Yuliang Zheng et al., HAVAL - A One-Way Hashing Algorithm with Variable Length of Output (Extended Abstract), pp. 83-105								
AAR	Chris Charnes and Josef Pieprzky, Linear Nonequivalence versus Nonlinearity, Pieprzky, pp. 156-164								

Examiner _____ Date Considered: _____

*EXAMINER: Initial if citation considered, whether or not citation is in conformance with MPEP § 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to Applicant.

FORM PTO-1449 (modified)
 To: U.S. Department of Commerce
 (PM&S FORM PAT-1449)
 Patent and Trademark Office

Atty. Dkt. No.	M#	Client Ref.
	252465	

**INFORMATION DISCLOSURE STATEMENT
 BY APPLICANT**

Applicant: FARBER et al.

Appl. No.: R.53(B)(1) Div. of 08/960,079

Filing Date: April 1, 1999

Examiner: Homere, J. Group Art Unit: 2776

Date: April 1, 1999 Page 2 of 3

U.S. PATENT DOCUMENTS

Examiner's Initials*	Document Number	Date MM/YYYY	Name (Family Name of First Inventor)	Class	Sub Class	Filing Date (if appropriate)
	AR 3,668,647	06/1992	EVANGELISTI			
	BR 4,215,402	07/1980	MITCHELL			
	CR 4,290,105	09/1981	CICHELLI			
	DR 4,376,299	03/1983	RIVEST			
	ER 4,405,829	09/1983	RIVEST			
	FR 4,412,285	10/1983	NECHES			
	GR 4,414,624	11/1983	SUMMER, JR.			
	HR 4,441,155	04/1984	FLETCHER			
	IR 4,464,713	08/1984	BENHASE			
	JR 5,772,933	03/1986	MATICK			
	KR 4,642,793	02/1987	MEADEN			
	LR 4,691,2999	09/1987	RIVEST			
	MR 4,725,945	02/1988	KRONSTADT			
	NR 4,773,039	09/1988	ZAMORA			

FOREIGN PATENT DOCUMENTS

	Document Number	Date MM/YYYY	Country	Inventor Name	Class	SubClass	English Abstract		Translation Readily Available	
							Enclosed	No	Enclose	No
	OR									
	PR									
	QR									
	RR									
	SR									
	TR									
	UR									

OTHER (Including in this order Author, Title, Periodical Name, Date, Pertinent Pages, etc.)

VR	Witold Litwin et al., Linear Hashing for Distributed Files, ACM SIGMOD, May 1993, pp. 327-336								
WR	Ming-Ling Lo et al., On Optimal Processor Allocation to Support Pipelined Hash Joins, ACM SIGMOD, pp. 69-78, May 1993								
XR	Thomas A. Berson, Differential Cryptanalysis Mod 2 ³² with Applications to MD5, pp. 69-81								
YR	William Perrizo et al., Distributed Join Processing Performance Evaluation, Twenty-Seventh Hawaii International Conference on System Sciences, Vol. II, pp. 236-244								
ZR	Vijay Kumar, A Concurrency Control Mechanism Based on Extendible Hashing for Main Memory Database Systems, ACM, Vol. 3, 1989, pp. 109-113								
AAR	Birgit Pfizman, Sorting Out Signature Schemes, November 1993, 1 st Conf. Computer & Comm. Security '93, p. 74-85								

Examiner _____ Date Considered: _____

*EXAMINER: Initial if citation considered, whether or not citation is in conformance with MPEP § 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to Applicant.

FORM PTO-1449 (modified)
 To: U.S. Department of Commerce
 (PM&S FORM PAT-1449)
 Patent and Trademark Office

Atty. Dkt. No.	M#	Client Ref.
	252465	

**INFORMATION DISCLOSURE STATEMENT
 BY APPLICANT**

Applicant: FARBER et al.

Appl. No.: R.53(B)(1) Div. of 08/960,079

Filing Date: April 1, 1999

Examiner: Homere, J. Group Art Unit: 2776

Date: April 1, 1999 Page 3 of 3

U.S. PATENT DOCUMENTS

Examiner's Initials*	Document Number	Date MM/YYYY	Name (Family Name of First Inventor)	Class	Sub Class	Filing Date (if appropriate)
AR	4,887,235	12/1989	HOLLOWAY			
BR	4,888,681	12/1989	BARNES			
CR	4,490,782	12/1984	DIXON			
DR	4,972,367	11/1990	BURKE			
ER	4,922,414	05/1990	HOLLOWAY			
FR	5,057,837	10/1991	COLWELL			
GR	5,007,658	12/1991	BENDERT			
HR	5,025,421	06/1991	CHO			
IR	5,129,081	07/1992	KOBAYASHI			
JR	5,129,082	07/1992	TIRFING			
KR	5,144,667	09/1992	POGUE, JR.			
LR	5,179,680	01/1993	COLWELL			
MR	5,301,286	04/1994	RAJANI			
NR	5,404,508	04/1995	KONRAD			

FOREIGN PATENT DOCUMENTS

	Document Number	Date MM/YYYY	Country	Inventor Name	Class	SubClass	English Abstract		Translation Readily Available	
							Enclosed	No	Enclose	No
OR										
PR										
QR										
RR										

OTHER (Including in this order Author, Title, Periodical Name, Date, Pertinent Pages, etc.)

SR	Zhiyu Tian et al., A New Hashing Function: Statistical Behaviour and Algorithm, pp. 3-13									
TR	G. L. Friedman, Digital Camera with Apparatu for Authentication of Images Produced from an Image File, NASA CASE NO. NPO-19108-1-CU, Serial No. 08/159,980, November 24, 1993									
UR	H. Goodman, Ada, Object-Oriented Techniques, and Concurrency in Teaching Data Structures and File Management REPORT DOCUMENTATION PAGE AD-A275 385 - 94-04277									
VR	Advances in Cryptology-EUROCRYPT'93, Workshop on the Theory and Application of Cryptographic Techniques Lofthus, Norway, May 23-27, 1993 Proceedings									
WR	Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, Vol. 22, Issue 2, June 1993									
XR	Advances in Cryptology-AUSCRYPT '92 - Workshop on the Theory and Application of Cryptographic Techniques Gold Coast, Queensland, Australia, December 13-16, 1992 Proceedings									

Examiner: _____ Date Considered: _____

*EXAMINER: Initial if citation considered, whether or not citation is in conformance with MPEP § 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to Applicant.

4. Priority is claimed under 35 U.S.C. 119/365 based on filing in _____ of _____ (country)
- | | | | | | | | |
|-----------|-----------------|-------|-------------|-----------|-----------------|-------|-------------|
| (1) _____ | Application No. | _____ | Filing Date | (4) _____ | Application No. | _____ | Filing Date |
| (2) _____ | | | | (5) _____ | | | |
| (3) _____ | | | | (6) _____ | | | |

- a. _____ (No.) Certified copy/copies attached.
 b. Certified copy/copies previously filed on _____ in _____ U.S. Application No. _____ / _____, filed on _____ series code ↑ serial no.

4. c. Certified copy/copies filed during international stage of PCT/
 (a) Domestic priority is claimed from PCT/ _____ / _____, filed _____
 (b) Benefit is claimed of Provisional Application No. 60/_____, filed _____

5. Prior application is assigned to kiNETech, Inc.
 by assignment recorded June 23, 1995 (Date) Reel 7593 Frame 0036

6. Attached is the following number of Assignments (including original and all later successive ones by different assignors): 1 and respective new Cover Sheets. (Do **NOT** file old cover sheets.)
 (Assignments in parent **must be refiled** with new Cover Sheets in this continuing application if you want it/them recorded against the continuing application.)

Please return the recorded Assignment to the undersigned.

7. The power of attorney in the prior application is to Dale S. Lazar, Reg. No. 28,872
 (Name and Reg. No.)
 whose current address is as in item 8 below.

- a. Recognize as associate attorney Brian Sirtzky, Reg. No. 37,497
 (Name, Reg. No. and Address)

8. **Address all future communications to Intellectual Property Group of Pillsbury Madison & Sutro LLP, Ninth Floor, East Tower 1100 New York Avenue, N.W., Washington, D.C. 20005-3918**

9. Amend the specification by inserting before the first line the sentence: This is a
 continuation division of Application No. 08/960,079, filed October 24, 1997, now
Patent No. 5,978,7
 which is a continuation of 08/425,160, filed April 11, 1995, now abandoned.

9. (a) Amend the specification by inserting before the first line: --This application claims the benefit of Provisional Application No. 60/_____, filed _____ -- 10/24/1

10. It has been recently determined that this new continuing application is entitled to small entity status.
 Hence:
 (No.) Verified Statement(s) establishing "small entity" status under Rules 9 & 27 were/are:
 filed in above prior application (and hence applicable hereto)
 attached.

11. Petition to extend the life of the above prior application to at least the date hereof
 (one box) is being concurrently filed in that prior application (Use Form PAT-111).
 (must be) was previously filed in that prior application (Check length of prior extension).
 (X'd) is not necessary for copendency (Double check before X'ing this box).

667770 09122600

A1

79

A

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 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.
13. Attached is a Rule 103(a) Petition to Suspend Action.
14. **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.

FILING FEE

THE FOLLOWING FILING FEE IS BASED ON
 ->->->CLAIMS AS FILED AND CHANGED BY PRELIMINARY AMENDMENT IN ITEM 14<-<-<-<-

NOTE: If box 1A2 is X'd, do not pay fees, but leave lines 15-22 and 27-32 blank.

				Large/Small Entity		Fee Code
15. Basic Filing Fee Design Application				\$310/\$155		106/26
16. Basic Filing Fee Not Design Application				\$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 <u>any proper</u> multiple dependent claim (ignore improper) is present,				\$260/\$130	+0	104/204
20. Subtotal =				\$380		
21. If "petition" box 13 above is X'd, add petition fee. \$130					+0	122
21A. If box 6 above is X'd, add Assignment recording fee \$ 40					+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.)
25. The following PRELIMINARY AMENDMENT is to be entered after assigning Appln. No.:

26.

**ADDITIONAL FEE CALCULATION FOR
PRELIMINARY AMENDMENT
PER BOXES 24/25**

	Claims remaining after amendment	Highest number previously paid for	Present Extra	Additional Fee	
					Large/Small Entity File Code
27.	Total Effective Claims *	minus ** 20	= 0	x \$18/\$9 = \$ 0	(103/203)
28.	Independent Claims *	minus *** 3	= 0	x \$78/\$39 = + 0	(102/202)
29.	If amendment enters proper multiple dependent claim(s) into this application for the <u>first time</u> , add (per application) \$260/\$130				+ 0 (104/204)
30.	ADDITIONAL FEE				\$ 0
31.	plus FEE from item 22 on page 3				+ 420
32.	TOTAL FEE ATTACHED				\$ 420

RECEIVED

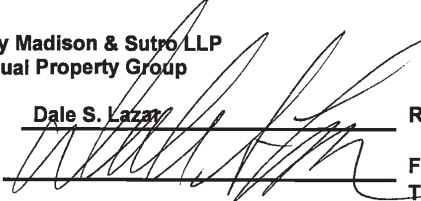
- 33. *If the entry in this space is less than the entry in the next space, the "Present Extra" result is "0"
- 34. **If the "Highest number previously paid for" (see item 17 above) is less than 20, write "20" in this space
- 35. If the "Highest number previously paid for" (see item 18 above) is less than 3, write "3" in this space

CHARGE STATEMENT: 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 duplicate 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.

**Pillsbury Madison & Sutro LLP
Intellectual Property Group**

1100 New York Avenue, N.W.
Ninth Floor, East Tower
Washington, D.C. 20005-3918
Tel: (202) 861-3000
DSL/BS:kim
Atty./Sec.

By Atty: Dale S. Lazar Reg. No. 28872
 Sig:  Fax: (202) 822-0944
Tel: (202) 861-3527

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

A

SUB C1

B1

--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

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

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.

B1

Application of I. Ober 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) 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:

B1

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 ~~iber 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) 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~~

B1

Application of ~~_____~~ 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:

B1

Application of [redacted] 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.

B1

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 54 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 ~~er et al, No. 09/283,160~~

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

B1

Application of [redacted] 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 of availability 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;

B1

Application of ~~_____~~ 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 item;

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.

B1

Application of Iyer 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.

B1

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.

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) A method as in claim 54 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.

B1

Application of Lerner 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.

B1

Application of [redacted] 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;

B1

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

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.

B1

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.--

B1

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

REMARKS

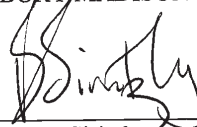
By this Amendment, the title has been replaced, all pending claims (claims 46-49) have been cancelled without prejudice or disclaimer of their subject matter, and new claims 54-106 have been added. Claims 1-45 and 50-53 were cancelled by preliminary amendment with the filing of this application. Claims 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

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

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE PATENT APPLICATION

Inventor(s): FARBER et al.
 Appln. No.: 09 | 283,160
 Series Code ↑ | Serial No. ↑
 Filed: April 1, 1999
 Hon. Commissioner of Patents
 Washington, D.C. 20231

Group Art U. 2771
 Examiner: Homere, Jean R.
 Atty. Dkt. PM 252465
 M# | Client Ref



Appln. Title: IDENTIFYING DATA IN A DATA PROCESSING SYSTEM **RECEIVED**

JAN 18 2001

Sir: **REPLY/AMENDMENT/LETTER**

Date: January 11, 2001 **Technology Center 21**

This is a reply/amendment/letter in the above-identified application and includes the herewith attachment of same date and subject which is incorporated hereinto by reference and the signature below is treated as the signature to the attachment in absence of a signature thereto.

FEE REQUIREMENTS FOR CLAIMS AS AMENDED

1. Small Entity claim	Claims remaining after amendment	Highest number previously paid for	Present Extra	Large/Small Entity	Additional Fee	Fee Code
A. <input checked="" type="checkbox"/> NOT made B. <input checked="" type="checkbox"/> Withdrawn C. <input type="checkbox"/> made herewith D. <input type="checkbox"/> made previously						
For B & C See Required Separate Paper (Pat-256)						
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
4. If amendment enters proper multiple dependent claim(s) into this application for first time (leave blank if this is a reissue application)				add	+ \$270/\$135 =	+ \$ 104/204
5. Original due Date:	<input type="checkbox"/> NONE					
6. Petition is hereby made to extend the original due date to cover the date this response is filed for which the requisite fee is attached	(1 mo)	\$110/\$55 =				115/215
	(2 mos)	\$390/\$195 =		+ \$		116/216
	(3 mos)	\$890/\$445 =				117/217
	(Usable only for ≤ 2mo.OA --- 4 mos)	\$1390/\$695=				118/218
	(Usable only for 30 day/1mo.OA --- 5 mos)	\$1890/\$945=				128/228
7. Enter any previous extension fee paid since above original due date and subtract				- \$0		
8. Extension Fee Attached					+ \$	
9. If Terminal Disclaimer attached, add Rule 20(d) official fee				+ \$110/\$55	+ \$0	148/248
10. If IDS attached requires Official Fee under Rule 97 (c),				add + \$180	+ \$0	126
or if Rule 97(d) Request				add + \$180		126
11. After-Final Request Fee per rules 129(a) and 17(r)				+ \$710/355	+ \$0	146/246
12. No. of additional inventions for examination per Rule 129(b).....				x \$710/355 ea	+ \$0	149/249
13. Request for Continued Examination (RCE)				+ \$710/355	+ \$0	1179/1279
14. Petition fee for					+ \$0	
15. TOTAL FEE ENCLOSED =						\$1874

16. *If the entry in this space is less than entry in next space, the "Present Extra" result is "0".
 17. **If the "Highest number previously paid for" in this space is less than 20, write "20" in this space.
 18. ***If the "Highest number previously paid for" in this space is less than 3, write "3" in this space.

Our Deposit Account No. 03-3975)
 (Our Order No. 18404 | 0000003
 C# | M#

CHARGE STATEMENT: 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 insufficiencies only) now or hereafter relative to this application and the resulting Official Document under Rule 20, or credit any overpayment, to our Accounting/Order Nos. shown above, for which purpose a duplicate copy of this sheet is attached.

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

Query: Is appeal deadline now? If so, file Notice of Appeals separately.

Pillsbury Winthrop LLP
Intellectual Property Group
 By Atty: Brian Britzky
 1100 New York Avenue, NW
 Ninth Floor
 Washington, DC 20005-3918
 Tel: (202) 861-3000
 Atty/Sec: BS/ans

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

NOTE: File this cover sheet in duplicate with PTO receipt (PAT-103A) and attachments

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

PATENT APPLICATION

Inventor(s): FARBER et al.
Appln. No.: 09 | 283,160
Series Code ↑ | Serial No. ↑
Filed: April 1, 1999
Pat. No.:

Group Art Unit 2771
Examiner: Homere, Jean R.
Atty. Dkt. PM 252465
In. Title: IDENTIFYING DATA IN A DATA PROCESSING SYSTEM



RECEIVED

JAN 18 2001

Technology Center 2100

Issued:
Hon. Commissioner of Patents
Washington, D.C. 20231

Sir:

Date: January 11, 2001

NOTIFICATION OF CHANGE IN SMALL/LARGE ENTITY STATUS

Please note the following change to/from Small Entity Status:

- CLAIMED NOW:** Applicant hereby claims Small Entity Status
- WITHDRAWN NOW:** Applicant hereby withdraws Small Entity 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 Group

By Atty: Brian Sritzky | Reg. No. 37497
Sig: [Signature] | Fax: (202) 822-0944
Tel: (202) 861-3702

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

PATENT APPLICATION

Inventor(s): FARBER et al.
 Appln. No.: 09 | 283,160
 Series Code ↑ | Serial No. ↑
 Filed: April 1, 1999
 Hon. Commissioner of Patents
 Washington, D.C. 20231



Group Art Un 2771
 Examiner: Homere, Jean R. *2771*
 Atty. Dkt. PM 252465
 M# | Client Ref

Appln. Title: IDENTIFYING DATA IN A DATA PROCESSING SYSTEM

RECEIVED

JAN 1 8 2001

Sir:

REPLY/AMENDMENT/LETTER

Date: January 11, 2001

Technology Center 210

This is a reply/amendment/letter in the above-identified application and includes the herewith attachment of same date and subject which is incorporated hereinto by reference and the signature below is treated as the signature to the attachment in absence of a signature thereto.

FEE REQUIREMENTS FOR CLAIMS AS AMENDED

1. Small Entity claim	Claims remaining after amendment	Highest number previously paid for		Present Extra	Large/Small Entity	Additional Fee	Fee Code Lg/Sm	
A. <input checked="" type="checkbox"/> NOT made B. <input checked="" type="checkbox"/> Withdrawn C. <input type="checkbox"/> made herewith D. <input type="checkbox"/> made previously For B & C See <u>Required Separate Paper</u> (Pat-256)	53	**minus	20	33	x \$18/\$9 =	+\$594	103/203	
	19	***minus	3	16	x \$80/\$40 =	+\$1280	102/202	
4. If amendment enters <u>proper</u> multiple dependent claim(s) into this application for <u>first</u> time (leave blank if this is a reissue application)					add	+\$270/\$135 =	+	104/204
5. <u>Original due Date:</u>			<input type="checkbox"/> NONE					
6. <u>Petition is hereby made</u> to extend the original due date to cover the date this response is filed for which the requisite fee is attached (Usable <u>only</u> for ≤ 2mo.OA --- 4 mos) (Usable <u>only</u> for 30 day/1mo.OA --- 5 mos)	(1 mo)	\$110/\$55 =					115/215	
	(2 mos)	\$390/\$195 =			+		116/216	
	(3 mos)	\$890/\$445 =					117/217	
		\$1390/\$695 =					118/218	
		\$1890/\$945 =					128/228	
7. Enter any previous extension fee paid since above <u>original due date</u> and <u>subtract</u>					- \$0			
8. <u>Extension Fee Attached</u>					+			
9. If <u>Terminal Disclaimer</u> attached, add Rule 20(d) official fee					+	\$110/\$55	148/248	
10. If IDS attached requires Official Fee under Rule 97 (c),					add	+\$180	126	
or if Rule 97(d) Request					add	+\$180	126	
11. After-Final Request Fee per rules 129(a) and 17(r)					+	\$710/355	146/246	
12. No. of additional inventions for examination per Rule 129(b).....					x	\$710/355 ea	149/249	
13. Request for Continued Examination (RCE)					+	\$710/355	1179/1279	
14. Petition fee for					+	\$0		
15. TOTAL FEE ENCLOSED =						\$1874		

16. *If the entry in this space is less than entry in next space, the "Present Extra" result is "0".
 17. **If the "Highest number previously paid for" in this space is less than 20, write "20" in this space.
 18. ***If the "Highest number previously paid for" in this space is less than 3, write "3" in this space.

Our Deposit Account No. 03-3975)
 (Our Order No. 18404 | 0000003
 C# | M#

CHARGE STATEMENT: 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 insufficiencies only) now or hereafter relative to this application and the resulting Official Document under Rule 20, or credit any overpayment, to our Accounting/Order Nos. shown above, for which purpose a duplicate copy of this sheet is attached.

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

Query: Is appeal deadline now? If so, file Notice of Appeals separately.

1100 New York Avenue, NW
 Ninth Floor
 Washington, DC 20005-3918
 Tel: (202) 861-3000
 Atty/Sec: BS/ans

Pillsbury Winthrop LLP
Intellectual Property Group

By Atty: Brian Sinizky

Sig: *[Signature]*

Reg. No. 37497

Fax: (202) 822-0944
 Tel: (202) 861-3702

NOTE: File this cover sheet in duplicate with PTO receipt (PAT-103A) and attachments

#4

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re PATENT APPLICATION of

FARBER et al.

Group Art Unit: 2776

Appln. No. 09/283,160

Examiner: Homere, Jean R

Filed: April 1, 1999



RECEIVED
MAR -7 2001
JTC 2100

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

* * * * *

March 2, 2001

PETITION TO MAKE SPECIAL BECAUSE OF ACTUAL INFRINGEMENT

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

Sir:

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.

Respectfully submitted,

PILLSBURY 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

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re PATENT APPLICATION of

FARBER et al.

Group Art Unit: 2776

Appln. No. 09/283,160



Examiner: Homere, Jean R.

Filed: April 1, 1999

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

* * * * *

March 2, 2001

RECEIVED
MAR - 7 2001
TOLSON

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 Sirtzky, 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.
2. 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***

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 field). 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”

***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.
13. 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 Siritzky

30152415v1

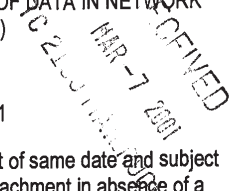
IN THE UNITED STATES PATENT AND TRADEMARK OFFICE PATENT APPLICATION

Inventor(s): FARBER et al.
 Appln. No.: 09 | 283,160
 Series Code ↑ | Serial No. ↑

Group Art Unit ..771
 Examiner: Homere, Jean R.
 Atty. Dkt. P 252465
 M# | Client Ref

Filed: April 1, 1999
 Hon. Commissioner of Patents
 Washington, D.C. 20231

Appln. Title: IDENTIFYING DATA BASED ON
 CONTENTS OF DATA IN NETWORK
 (As Amended)



Sir: **REPLY/AMENDMENT/LETTER**

Date: March 2, 2001

This is a reply/amendment/letter in the above-identified application and includes the herewith attachment of same date and subject which is incorporated hereinto by reference and the signature below is treated as the signature to the attachment in absence of a signature thereto.

FEE REQUIREMENTS FOR CLAIMS AS AMENDED

1. Small Entity claim
 A. NOT made
 B. Withdrawn
 C. made herewith
 D. made previously
 For B & C See Required Separate Paper (Pat-256)

	Claims remaining after amendment	Highest number previously paid for	Present Extra	Large/Small Entity	Additional Fee	Fee Code Lg/Sm	
2. Total Effective Claims	5	**minus 20	0	x \$18/\$9 =	+\$0	103/203	
3. Independent Claims	3	***minus 3	0	x \$80/\$40 =	+\$0	102/202	
4. If amendment enters <u>proper</u> multiple dependent claim(s) into this application for <u>first</u> time (leave blank if this is a reissue application)				add	+\$270/\$135 =	104/204	
5. <u>Original due Date:</u>	<input type="checkbox"/> NONE						
6. <u>Petition is hereby made</u> to extend the original due date to cover the date this response is filed for which the requisite fee is attached (1 mo) \$110/\$55 = (2 mos) \$390/\$195 = (3 mos) \$890/\$445 = (Usable <u>only</u> for ≤ 2mo.OA --- 4 mos) \$1390/\$695= (Usable <u>only</u> for 30 day/1mo.OA --- 5 mos) \$1890/\$945=				+		115/215 116/216 117/217 118/218 128/228	
7. Enter any previous extension fee paid since above <u>original due date</u> and <u>subtract</u>	- \$0						
8.	Extension Fee Attached					+\$	
9. If <u>Terminal Disclaimer</u> attached, add Rule 20(d) official fee					+\$110/\$55	+ \$0 148/248	
10. If IDS attached requires Official Fee under Rule 97 (c),					add + \$180	+ \$0 126	
or if Rule 97(d) Request					add + \$180	+ \$0 126	
11. After-Final Request Fee per rules 129(a) and 17(r)					+\$710/355	+ \$0 146/246	
12. No. of additional inventions for examination per Rule 129(b).....					x \$710/355 ea	+ \$0 149/249	
13. Request for Continued Examination (RCE)					+\$710/355	+ \$0 1179/1279	
14. Petition fee for					+\$130		
15.	TOTAL FEE ENCLOSED =					\$130	

16. *If the entry in this space is less than entry in next space, the "Present Extra" result is "0".
 17. **If the "Highest number previously paid for" in this space is less than 20, write "20" in this space.
 18. ***If the "Highest number previously paid for" in this space is less than 3, write "3" in this space.

Our Deposit Account No. 03-3975
 (Our Order No. 7018 | 252465
 C# | M#

CHARGE STATEMENT: 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 insufficiencies only) now or hereafter relative to this application and the resulting Official Document under Rule 20, or credit any overpayment, to our Accounting/Order Nos. shown above, for which purpose a duplicate copy of this sheet is attached.

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

Query: Is appeal deadline now? If so, file Notice of Appeals separately.

Pillsbury Winthrop LLP
 Intellectual Property Group
 By Atty: Brian Sirtzky
 Sig:
 1100 New York Avenue, NW
 Ninth Floor
 Washington, DC 20005-3918
 Tel: (202) 861-3000
 Atty/Sec: BS/ans

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

NOTE: File this cover sheet in duplicate with PTO receipt (PAT-103A) and attachments



UNITED STATES PATENT AND TRADEMARK OFFICE

COMMISSIONER FOR PATENTS
UNITED STATES PATENT AND TRADEMARK OFFICE
WASHINGTON, DC 20231
www.uspto.gov

Mailed

APR 13 2001

Technology Center 2100

Paper No. 5

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. §1.102(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.

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

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re PATENT APPLICATION OF

FARBER et al.

Group Art Unit: 2771

Appln. No.: 09/283,160

Examiner: Jean Homere

Filed: April 1, 1999

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.



#6
4/26/01
A.W.
RECEIVED
APR 18 2001
Technology Center 2100



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

FORM PTO-1449 (modified)
 To: U.S. Department of Commerce
 (PW FORM PAT-1449)
 Patent and Trademark Office

Atty. Dkt. No. M# Client Ref.
 252465

INFORMATION DISCLOSURE STATEMENT BY APPLICANT



Applicant: FARBER et al.
 Appl. No.: 09/283,160
 Filing Date: April 1, 1999
 Examiner: Jean Homere Group Art Unit: 2771

Date: April 18, 2001 Page 1 of 3

U.S. PATENT DOCUMENTS

Examiner's Initials*	Document Number	Date MM/YYYY	Name (Family Name of First Inventor)	Class	Sub Class	Filing Date (if appropriate)
JRH	AR 5,638,443	06/1997	STEFIK et al.	705	54	11/23/94
	BR 5,802,291	09/1998	BALICK et al.	709	202	3/30/95
	CR 5,542,087	07/1996	NEIMAT et al.	707	10	10/15/93
	DR 5,287,499	02/1994	NEMES	707	2	5/16/91
	ER 4,922,417	05/1990	CHURM et al.	707	1	10/24/86
	FR 5,341,477	08/1994	PITKIN et al.	709	226	8/6/93
	GR 5,907,704	05/1999	GUDMUNDSON et al.			4/3/95
	HR					
	IR					
	JR					
	KR					
	LR					
	MR					
	NR					

FOREIGN PATENT DOCUMENTS

Document Number	Date MM/YYYY	Country	Inventor Name	English Abstract		Translation Readily Available	
				Enclosed	No	Enclose	No
OR 05162529	06/1993	JP	HOROSHI	X			X

OTHER (Including in this order Author, Title, Periodical Name, Date, Pertinent Pages, etc.)

JRH	PR	Gwertzman, James, et al. "The Case for Geographical Push-Caching." Technical Report HU TR 34-94 (excerpt), Harvard University, DAS, Cambridge, MA 02138, 1994, 2 pgs.				
	QR	Grigni, Michelangelo, et al. "Tight Bounds on Minimum Broadcasts Networks." SIAM Journal of Discrete Mathematics, Vol. 4, No. 2, May 1991, pgs. 207-222.				
	RR	Devine, Robert. "Design and Implementation of DDH: A Distributed Dynamic Hashing Algorithm." In Proceedings of 4th International Conference on Foundations of Data Organizations and Algorithms, 1993, pgs. 101-114.				
	SR	Deering, Stephen, et al. "Multicast Routing in Datagram Internetworks and Extended LANs." ACM Transactions on Computer Systems, Vol. 8, No. 2, May 1990, pgs. 85-110.				
	TR	Cormen, Thomas H., et al. Introduction to Algorithms, The MIT Press, Cambridge, Massachusetts, 1994, pgs. 219-243, 991-993.				
	UR	Naor, Moni, et al. "The Load, Capacity and Availability of Quorum Systems." In Proceedings of the 35th IEEE Symposium on Foundations of Computer Science, November 1994, pgs. 214-225.				

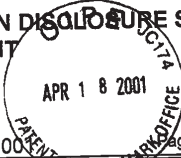
Examiner Jean R. Homere Date Considered: 5/30/01

*EXAMINER: Initial if citation considered, whether or not citation is in conformance with MPEP § 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to Applicant.

FORM PTO-1449 (modified)
 To: U.S. Department of Commerce
 (PW FORM PAT-1449)
 Patent and Trademark Office

Atty. Dkt. No.	M#	Client Ref.
	252465	
Applicant: FARBER et al.		RECEIVED APR 1 8 2001 Technology Center 200
Appl. No.: 09/283,160		
Filing Date: April 1, 1999		
Examiner: Jean Homere	Group Art Unit 22771	

INFORMATION DISCLOSURE STATEMENT BY APPLICANT



Date: April 18, 2001 Page 2 of 3

U.S. PATENT DOCUMENTS

Examiner's Initials*	Document Number	Date MM/YYYY	Name (Family Name of First Inventor)	Class	Sub Class	Filing Date (if appropriate)
----------------------	-----------------	--------------	--------------------------------------	-------	-----------	------------------------------

FOREIGN PATENT DOCUMENTS					English Abstract		Translation Readily Available	
	Document Number	Date MM/YYYY	Country	Inventor Name	Enclosed	No	Enclose	No

OTHER (Including in this order Author, Title, Periodical Name, Date, Pertinent Pages, etc.)

JRH	AR	Nisan, Noam. "Pseudorandom Generators for Space-Bounded Computation." In Proceedings of the Twenty-Second Annual ACM Symposium on Theory of Computing, May 1990, pgs. 204-212.						
	BR	Palmer, Mark, et al. "Fido: A Cache that Learns to Fetch." In Proceedings of the 17th International Conference on Very Large Data Bases, September 1991, pgs. 255-264.						
	CR	Peleg, David, et al. "The Availablitiy of Quorum Systems." Information and Computation 123, 1995, 210-223.						
	DR	Rabin, Michael. "Efficient Dispersal of Information for Security, Load Balancing, and Fault Tolerance." Journal of the ACM, Vol. 36, No. 2, April 1989, pgs. 335-348						
	ER	Ravi, R., "Rapid Rumor Ramification: Approximating the Minimum Broadcast Time." In Proceedings of the 35th IEEE Symposium on Foundation of Computer Science, November 1994, pgs. 202-213.						
	FR	Schmidt, Jeanette, et al. "Chernoff-Hoeffding Bounds for Applications with Limited Independence." In Proceedings of the 4th ACS-SIAM Symposium on Discrete Algorithms, 1993, pgs. 331-340.						
	GR	Tarjan, Robert Endre, et al. "Storing a Sparse Table." Communications of the ACM, Vol. 22, No. 11, November 1979, pgs. 606-611.						
	HR	Wegman, Mark, et al. "New Hash Functions and Their Use in Authentication and Set Equality." Journal of Computer and System Sciences Vol. 22, June 1981, pgs. 265-279.						
	IR	Vitter, Jeffrey Scott, et al. "Optimal Prefetching via Data Compression." In Proceedings of 32nd IEEE Symposium on Foundations of Computer Science, November 1991, pgs. 121-130.						
	JR	Fredman, Michael, et al. "Storing a Sparse Table with O(1) Worst Case Access Time." Journal of the Association for Computing Machinery, Vol. 31, No. 3, July 1984, pgs. 538-544.						
	KR	Yao, Andrew Chi-Chih. "Should Tables be Sorted?" Journal of the Association for Computing Machinery, Vol. 28, No. 3, July 1981, pgs. 615-628.						
	LR	Floyd, Sally, et al. "A reliable Multicast Framework for Light-Weight Sessions and Application Level Framing." In Proceeding of ACM SIGCOMM '95, pgs. 342-356.						
	MR	Feeley, Michael, et al. "Implementing Global Memory Management in a Workstation Cluster." In Proceedings of the 15th ACM Symposium on Operating Systems Principles, 1995, pgs. 201-212.						

Examiner Jean R. Homere Date Considered: 5/30/01

*EXAMINER: Initial if citation considered, whether or not citation is in conformance with MPEP § 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to Applicant.

FORM PTO-1449 (modified)
 To: U.S. Department of Commerce
 (PW FORM PAT-1449)
 Patent and Trademark Office

Atty.
 Dkt. No.

M#

Client Ref.

252465

Applicant: FARBER et al.

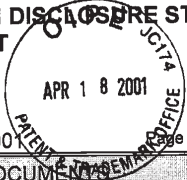
Appl. No.: 09/283,160

Filing Date: April 1, 1999

Examiner: Jean Homere

Group Art Unit: 2771

**INFORMATION DISCLOSURE STATEMENT
 BY APPLICANT**



Date: April 18, 2001

Page 3

of

3

U.S. PATENT DOCUMENTS

Examiner's Initials*	Document Number	Date MM/YYYY	Name (Family Name of First Inventor)	Class	Sub Class	Filing Date (if appropriate)
----------------------	-----------------	--------------	--------------------------------------	-------	-----------	------------------------------

FOREIGN PATENT DOCUMENTS

Document Number	Date MM/YYYY	Country	Inventor Name	English Abstract		Translation Readily Available	
				Enclosed	No	Enclose	No

OTHER (Including in this order Author, Title, Periodical Name, Date, Pertinent Pages, etc.)

JRH	AR	Carter, J. Lawrence, et al. "Universal Classes of Hash Functions." Journal of Computer and System Sciences, Vol. 18, No. 2, April 1979, pgs. 143-154.				
JRH	BR	PATENT ABSTRACTS OF JAPAN, "Electronic Mail Multiplexing System and Communication Control Method in The System." 06/30/19993, JP 05162529				
	CR					
	DR					
	ER					
	FR					
	GR					
	HR					
	IR					
	JR					
	KR					
	LR					
	MR					

Examiner *Jean R. Homere*

Date Considered: 5/30/01

*EXAMINER: Initial if citation considered, whether or not citation is in conformance with MPEP § 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to Applicant.

6



**UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office**

Address: COMMISSIONER OF PATENTS AND TRADEMARKS
Washington, D.C. 20231

DM

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.
09/283,160	04/01/99	FARBER	D PM252465

TM02/0605

PILLSBURY MADISON & SUTRO
INTELLECTUAL PROPERTY GROUP
1100 NEW YORK AVENUE NW
NINTH FLOOR EAST TOWER
WASHINGTON DC 20005-3918

EXAMINER

HOMERE, J

ART UNIT	PAPER NUMBER
----------	--------------

7

2177
DATE MAILED: 06/05/01


Please find below and/or attached an Office communication concerning this application or proceeding.

Commissioner of Patents and Trademarks

RE

Office Action Summary

Application No. 09/283,160	Applicant(s) Farber et al.
Examiner Jean R. Homere	Art Unit 2177



- The MAILING DATE of this communication appears on the cover sheet with the correspondence address -

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE three MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136 (a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) Responsive to communication(s) filed on Apr 18, 2001
- 2a) This action is FINAL.
- 2b) This action is non-final.
- 3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 35 C.D. 11; 453 O.G. 213.

Disposition of Claims

- 4) Claim(s) 54-106 is/are pending in the application.
- 4a) Of the above, claim(s) none is/are withdrawn from consideration.
- 5) Claim(s) _____ is/are allowed.
- 6) Claim(s) 54-106 is/are rejected.
- 7) Claim(s) _____ is/are objected to.
- 8) Claims _____ are subject to restriction and/or election requirements.

Application Papers

- 9) The specification is objected to by the Examiner.
- 10) The drawing(s) filed on Apr 1, 1999 is/are objected to by the Examiner.
- 11) The proposed drawing correction filed on _____ is: a) approved b) disapproved.
- 12) The oath or declaration is objected to by the Examiner.

Priority under 35 U.S.C. § 119

- 13) Acknowledgement is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d).
 - a) All b) Some* c) None of:
 - 1. Certified copies of the priority documents have been received.
 - 2. Certified copies of the priority documents have been received in Application No. _____.
 - 3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
- *See the attached detailed Office action for a list of the certified copies not received.
- 14) Acknowledgement is made of a claim for domestic priority under 35 U.S.C. § 119(e).

Attachment(s)

- 15) Notice of References Cited (PTO-892)
- 16) Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 17) Information Disclosure Statement(s) (PTO-1449) Paper No(s) 6
- 18) Interview Summary (PTO-413) Paper No(s) _____
- 19) Notice of Informal Patent Application (PTO-152)
- 20) Other _____

Serial Number: 09/283,160

Page 2

Art Unit: 2177

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.