

UTILITY  
SERIAL  
NUMBER

08 555376

PATENT DATE

JUL 15 1997

PATENT  
NUMBERSERIAL NUMBER  
08/555,376FILING DATE  
11/09/95CLASS  
395

SUBCLASS

620

GROUP ART UNIT  
2307

EXAMINER

Lomere

APPLICANTS

JAMES R. WOODHILL, HOUSTON, TX; LOUIS R. WOODHILL, RICHMOND, TX;  
WILLIAM R. MORE JR., HOUSTON, TX; JAY H. BERLIN, HOUSTON, TX.

\*\*CONTINUING DATA\*\*\*\*\*

VERIFIED THIS APPLN IS A CON OF 08/085,596 07/01/93 ABN

Verified, JRH

\*\*FOREIGN/PCT APPLICATIONS\*\*\*\*\*

VERIFIED

None JRH

FOREIGN FILING LICENSE GRANTED 01/17/96

Foreign priority claimed  
35 USC 119 conditions met☐ yes ☒ no  
☐ yes ☒ noAS  
FILEDSTATE OR  
COUNTRY

TX

SHEETS  
DRWGS.

14

TOTAL  
CLAIMS

4

INDEP.  
CLAIMS

2

FILING FEE  
RECEIVED

\$750.00

ATTORNEY'S  
DOCKET NO.

930855/ON

Verified and Acknowledged

Examiner's Initials  
JRH

ADDRESS

EDWARD L PENCOSKE  
KIRKPATRICK & LOCKHART  
1500 OLIVER BUILDING  
PITTSBURGH PA 15222

TITLE

SYSTEM & METHOD FOR DISTRIBUTED STORAGE MANAGEMENT ON NETWORKED  
COMPUTER SYSTEMS using binary object Identifiers.

12/29/96 JRH

U.S. DEPT. OF COMM./PAT. &amp; TM—PTO-436L (Rev.12-94)

PARTS OF APPLICATION  
FILED SEPARATELY2/2/97 1-8-97  
Applications Examiner

NOTICE OF ALLOWANCE MAILED

1-2-97

Jean R. Homere

Assistant Examiner

CLAIMS ALLOWED

Total Claims

18

Print Claim

1

ISSUE FEE

Amount Due

1290.00

Date Paid

3/12/97

PAUL V. KULIK  
PRIMARY EXAMINER  
GROUP 2300

Primary Examiner

DRAWING

Sheets Drwg.

14

Figs. Drwg.

1

Print Fig.

1

ISSUE  
BATCH  
NUMBER

EXHIBIT

Personal Web  
2007Label  
Area

PREPARED FOR ISSUE

WARNING: The information disclosed herein may be restricted. Unauthorized disclosure may be prohibited

UTILITY  
SERIAL  
NUMBER

08/085536

PATENT DATE

PATENT  
NUMBER

SERIAL NUMBER

08/085,596

FILING DATE | CLASS

07/01/93

395

SUBCLASS

600

GROUP ART UNIT

2307

EXAMINER

Horne

G. J. H.

APPLICANTS

JAMES R. WOODHILL, HOUSTON, TX; LOUIS R. WOODHILL, RICHMOND, TX;  
WILLIAM R. MORE JR., HOUSTON, TX; JAY H. BERLIN, HOUSTON, TX.

\*\*CONTINUING DATA\*\*\*\*\*  
VERIFIED

None JRH

\*\*FOREIGN/PCT APPLICATIONS\*\*\*\*\*  
VERIFIED

None JRH

FOREIGN FILING LICENSE GRANTED 10/05/93

Foreign priority claimed

☐ yes☒ no

35 USC 119 conditions met

☐ yes☒ noAS  
FILEDSTATE OR  
COUNTRY

TX

SHEETS  
DRWGS.

14

TOTAL  
CLAIMS

20

INDEP.  
CLAIMS

6

FILING FEE  
RECEIVED

\$1,062.00

ATTORNEY'S  
DOCKET NO.

93085

Verified and Acknowledged

Examiner's Initials

JRH

ADDRESS

CHRISTOPHER H. GEBHARDT  
KIRKPATRICK & LOCKHART  
1500 OLIVER BUILDING  
PITTSBURGH, PA 15222

TITLE

SYSTEM AND METHOD FOR DISTRIBUTED STORAGE MANAGEMENT ON NETWORKED  
COMPUTER SYSTEMS, Using Binary Object Identifiers

U.S. DEPT. of COMM.-Pat. &amp; TM Office - PTO-436L (rev. 10-78)

PARTS OF APPLICATION  
FILED SEPARATELY

Applications Examiner

NOTICE OF ALLOWANCE MAILED

CLAIMS ALLOWED

Total Claims

Print Claim

Assistant Examiner

ISSUE FEE

DRAWING

Amount Due

Date Paid

Sheets Drwg.

Figs. Drwg.

Print Fig.

ISSUE  
BATCH  
NUMBER

Primary Examiner

PREPARED FOR ISSUE

Label  
Area

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.

08/085,596

**SYSTEM AND METHOD FOR DISTRIBUTED STORAGE  
MANAGEMENT ON NETWORKED COMPUTER SYSTEMS**

**Transaction History**

<b>Date</b>	<b>Transaction Description</b>
07-01-1993	Initial Exam Team nn
08-02-1993	Notice Mailed--Application Incomplete--Filing Date Assigned
10-01-1993	Application Is Now Complete
10-18-1993	Application Captured on Microfilm
10-29-1993	Case Docketed to Examiner in GAU
09-13-1994	Case Docketed to Examiner in GAU
10-20-1994	Non-Final Rejection
10-21-1994	Mail Non-Final Rejection
02-23-1995	Information Disclosure Statement (IDS) Filed
02-23-1995	Information Disclosure Statement (IDS) Filed
02-23-1995	Response after Non-Final Action
02-23-1995	Request for Extension of Time - Granted
03-15-1995	Date Forwarded to Examiner
05-08-1995	Final Rejection
05-11-1995	Mail Final Rejection (PTOL - 326)
11-09-1995	Request for Extension of Time - Granted
01-11-1996	Aband. for Failure to Respond to O. A.
01-16-1996	Mail Abandonment for Failure to Respond to Office Action
02-01-1996	Abandonment for Purposes of Filing an FWC - File Combined with Child Application

# SYSTEM & METHOD FOR DISTRIBUTED STORAGE MANAGEMENT ON NETWORKED COMPUTER SYSTEMS USING BINARY OBJECT IDENTIFIERS

## Transaction History

Date	Transaction Description
12-14-1995	Initial Exam Team nn
01-23-1996	Application Captured on Microfilm
10-24-1996	Preliminary Amendment
12-04-1996	Case Docketed to Examiner in GAU
01-02-1997	Mail Notice of Allowance
01-02-1997	Notice of Allowance Data Verification Completed
02-05-1997	Amendment after Notice of Allowance (Rule 312)
02-28-1997	Date Forwarded to Examiner
03-06-1997	Mail Response to 312 Amendment (PTO-271)
03-06-1997	Response to Amendment under Rule 312
03-18-1997	Mailroom Date of Drawing(s)
03-20-1997	Issue Fee Payment Verified
04-14-1997	Drawing(s) Received at Publications
04-15-1997	Drawing(s) Matched to Application
04-18-1997	Drawing(s) Processing Completed
06-09-1997	Issue Notification Mailed
07-15-1997	Recordation of Patent Grant Mailed
04-02-2002	Correspondence Address Change
04-03-2002	Change in Power of Attorney (May Include Associate POA)
01-25-2005	Applicant Has Filed a Verified Statement of Small Entity Status in Compliance with 37 CFR 1.27

Date  
Entered  
or  
Counted

# CONTENTS

Date  
Received  
or  
Mailed

RECEIVED

OCT 21 1993

GROUP 2300

	1. Application _____ papers.	
	2. <u>Trans. Dec.</u>	<u>8-2-93</u>
	3. <u>Declaration &amp; Surcharge</u>	<u>09-03-93</u>
<u>10-20-94</u>	4. <u>By 3 mos</u>	<u>10-21-94</u>
	5. <u>Ext. of Time (1 mo.)</u>	<u>2/23/95</u>
	6. <u>Agmt A</u>	<u>2/23/95</u>
	7. <u>Prior Act</u>	<u>2/23/95</u>
<u>5-8-95</u>	8. <u>Final Rejection 3 mos w/AT</u>	<u>5-11-95</u>
	9. <u>Ext. Time (3 mo.)</u>	<u>11-9-95</u>
<u>1-11-96</u>	10. <u>Notice of Abandonment</u>	<u>1-16-96</u>
	11. _____	
	12. _____	
	13. _____	
	14. _____	
	15. _____	
	16. _____	
	17. _____	
	18. _____	
	19. _____	
	20. _____	
	21. _____	
	22. _____	
	23. _____	
	24. _____	
	25. _____	
	26. _____	
	27. _____	
	28. _____	
	29. _____	

Date  
Entered  
or  
Counted

# CONTENTS

Date  
Received  
or  
Mailed

RECEIVED

JAN 25 1996

GROUP 3200

Application ☒ papers.

12	Pre Amdt - B	11-09-95
13	Pre Amdt - C	11-09-95
14	Pre-Amtd D	10-24-96
12/31/96 15	reason for allow.	1-2-97, CB
16	Amdt E (B3/2)	2/05/97
17	Notice of Entry	3-6-97 CB
4/15/97 18	For the Extension (1/4/98) CB	3/18/97
19	PTO Grant JUL 15 1997	
20		
21		
22		

4

APPLICATION SERIAL NUMBER <b>08/555,376</b>		CROSS REFERENCE(S)				
APPLICANT'S NAME (PLEASE PRINT) <b>Wood Hill et al.</b>		CLASS	SUBCLASS (ONE SUBCLASS PER BLOCK)			
IF REISSUE, ORIGINAL PATENT NUMBER		<b>395</b>	<b>610</b>	<b>618</b>	<b>200.03</b>	<b>489</b>
		<b>395</b>	<b>497.04</b>	<b>185.01</b>	<b>183.17</b>	<b>182.04</b>
		<b>395</b>	<b>180</b>			
		<b>364</b>	<b>DIG.1</b>	<b>285</b>	<b>285.1</b>	<b>286.4</b>
INTERNATIONAL CLASSIFICATION						
<b>G</b>	<b>06F</b>		<b>11</b>	<b>00</b>		
GROUP ART UNIT <b>2307</b>		ASSISTANT EXAMINER (PLEASE STAMP OR PRINT FULL NAME) <b>JEAN R. HOLLIFRE</b>				
		PRIMARY EXAMINER (PLEASE STAMP OR PRINT FULL NAME) <b>PAUL KULIK</b>				

PTO 270  
(REV. 5-91)

ISSUE CLASSIFICATION SLIP

U.S. DEPARTMENT OF COMMERCE  
PATENT AND TRADEMARK OFFICE

CLASSIFIED		7	7-21-93
EXAMINER		434	7-31-93
TYPIST		T328	10-5-93
VERIFIER		358	10-7-93
CORPS CORR.			
SPEC. HAND		44	9-30-93
FILE MAINT.		TC	5-2-93
DRAFTING			

## INDEX OF CLAIMS

Claim	Date
Final Original	10 05
1	14 04
2	24 95
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	

SYMBOLS

✓ ..... Rejected

= ..... Allowed

- (Through numeral) Canceled

+ ..... Restricted

N ..... Non-elected

I ..... Interference

A ..... Appeal

O ..... Objected

Claim	Date
Final Original	
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	



POSITION	ID NO.	DATE
CLASSIFIER		
EXAMINER		552 1-16-96
TYPIST	343	1-17-96
VERIFIER	509	1/18/96
CORPS CORR.		
SPEC. HAND		
FILE MAINT.		
DRAFTING		

## INDEX OF CLAIMS

Claim	Date
Final	
Original	12
	30
	96
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	
15	
16	
17	
18	
19	
20	
1 21	=
2 22	=
3 23	=
4 24	=
5 25	=
6 26	=
7 27	=
8 28	=
9 29	=
10 30	=
11 31	=
12 32	=
13 33	=
14 34	=
15 35	=
16 36	=
17 37	=
18 38	=
39	
40	
41	
42	
43	

Claim	Date
Final	
Original	
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	

### SYMBOLS

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

08/085596

PATENT APPLICATION SERIAL NO. \_\_\_\_\_

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

160 KM 07/22/93 08085596

1 101

932.00 CK 93085



08/085596

PATENT

Docket No. 93085

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

## NEW APPLICATION TRANSMITTAL

Transmitted herewith for filing is the patent application of

Inventor(s):

1-00 et al  
James R. Woodhill et al.

**WARNING:** Patent must be applied for in the name(s) of all of the actual inventor(s). 37 CFR 1.41(a) and 1.53(b).

For (title):

SYSTEM AND METHOD FOR DISTRIBUTED STORAGE MANAGEMENT ON  
NETWORKED COMPUTER SYSTEMS

## 1. Type of Application

This new application is for a(n) (check one applicable item below):

- ☒ Original  
☐ Design  
☐ Plant

**WARNING:** Do not use this transmittal for a completion in the U.S. of an International Application under 35 U.S.C. 371(c)(4) unless the International Application is being filed as a divisional, continuation or continuation-in-part application.

**NOTE:** If one of the following 3 items apply then complete and attach ADDED PAGES FOR NEW APPLICATION TRANSMITTAL WHERE BENEFIT OF A PRIOR U.S. APPLICATION CLAIMED.

- ☐ Divisional  
☐ Continuation  
☐ Continuation-in-part (CIP)

## CERTIFICATION UNDER 37 CFR 1.10

I hereby certify that this New Application Transmittal and the documents referred to as enclosed therein are being deposited with the United States Postal Service on this date July 1, 1993 in an envelope as "Express Mail Post Office to Addressee" Mailing Label Number HB119745527US addressed to the: Commissioner of Patents and Trademarks, Washington, D.C. 20231.

Beth H. Retort

(Type or print name of person mailing paper)

  
(Signature of person mailing paper)

**NOTE:** Each paper or fee referred to as enclosed herein has the number of the "Express Mail" mailing label placed thereon prior to mailing. 37 CFR 1.10(b).

## 2. Benefit of Prior U.S. Application(s) (35 USC 120)

*NOTE: If the new application being transmitted is a divisional, continuation or a continuation-in-part of a parent case, or where the parent case is an International Application which designated the U.S., then check the following item and complete and attach ADDED PAGES FOR NEW APPLICATION TRANSMITTAL WHERE BENEFIT OF PRIOR U.S. APPLICATION(S) CLAIMED.*

- ☐ The new application being transmitted claims the benefit of prior U.S. application(s) and enclosed are ADDED PAGES FOR NEW APPLICATION TRANSMITTAL WHERE BENEFIT OF PRIOR U.S. APPLICATION(S) CLAIMED.

## 3. Papers Enclosed Which Are Required For Filing Date Under 37 CFR 1.53(b) (Regular) or 37 CFR 1.153 (Design) Application

43 Pages of specification

8 Pages of claims

1 Pages of Abstract

14 Sheets of drawing

☐ formal

☒ informal

**WARNING:** *DO NOT submit original drawings. A high quality copy of the drawings should be supplied when filing a patent application. The drawings that are submitted to the Office must be on strong, white, smooth, and non-shiny paper and meet the standards according to § 1.84. If corrections to the drawings are necessary, they should be made to the original drawing and a high-quality copy of the corrected original drawing then submitted to the Office. Only one copy is required or desired. Comments on proposed new 37 CFR 1.84. Notice of March 9, 1988 (1990 O.G. 57-62).*

**NOTE:** *"Identifying indicia such as the serial number, group and unit, title of the invention, attorney's docket number, inventor's name, number of sheets, etc., not to exceed 2 3/4 inches (7.0 cm.) in width may be placed in a centered location between the side edges within three fourths inch (19.1 mm.) of the top edge. Either this marking technique on the front of the drawing or the placement, although not preferred, of this information and the title of the invention on the back of the drawings is acceptable." Proposed 37 CFR 1.84(1). Notice of March 9, 1988 (1990 O.G. 57-62).*

## 4. Additional papers enclosed

- ☐ Preliminary Amendment
- ☐ Information Disclosure Statement
- ☐ Form PTO-1449
- ☐ Citations
- ☐ Declaration of Biological Deposit
- ☐ Authorization of Attorney(s) to Accept and Follow Instructions from Representative
- ☐ Special Comments
- ☐ Other

5. Declaration or oath

☐ Enclosed

executed by (check all applicable boxes)

☐ inventor(s).

☐ legal representative of inventor(s), 37 CFR 1.42 or 1.43

☐ joint inventor or person showing a proprietary interest on behalf of inventor who refused to sign or cannot be reached.

☐ this is the petition required by 37 CFR 1.47 and the statement required by 37 CFR 1.47 is also attached. See item 13 below for fee.

☒ Not Enclosed.

*WARNING: Where the filing is a completion in the U.S. of an International Application but where a declaration is not available or where the completion of the U.S. application contains subject matter in addition to the International Application the application may be treated as a continuation or continuation-in-part, as the case may be, utilizing ADDED PAGE FOR NEW APPLICATION TRANSMITTAL WHERE BENEFIT OF PRIOR U.S. APPLICATION CLAIMED*

☐ Application is made by a person authorized under 37 CFR 1.41(c) on behalf of all the above named inventor(s). (The declaration or oath, along with the surcharge required by 37 CFR 1.16(e) can be filed subsequently)

*NOTE: It is important that all the correct inventor(s) are named for filing under 37 CFR 1.41(c) and 1.53(b).*

☐ Showing that the filing is authorized. (Not required unless called into question 37 CFR 1.41(d).

6. Inventorship Statement

*WARNING: If the named inventors are each not the inventors of all the claims an explanation, including the ownership of the various claims at the time the last claimed invention was made, should be submitted*

The inventorship for all the claims in this application are:

☒ The same

or

☐ Are not the same. An explanation, including the ownership of the various claims at the time the last claimed invention was made,

☐ is submitted.

☐ will be submitted.

7. Language

*NOTE: An application including a signed oath or declaration may be filed in a language other than English. A verified English translation of the non-English language application and the processing fee of \$130.00 required by 37 CFR 1.17(k) is required to be filed with the application or within such time as may be set by the Office. 37 CFR 1.52(d).*

*NOTE: A non-English oath or declaration in the form provided or approved by the PTO need not be translated. 37 CFR 1.69(b).*

☒ English

☐ non-English

☐ the attached translation is a verified translation. 37 CFR 1.52(d).

8. Assignment

☒ An assignment of the invention to LEGENT Corporation

☐ is attached. A separate ☐ "COVER SHEET FOR ASSIGNMENT (DOCUMENT) ACCOMPANYING NEW PATENT APPLICATION" or ☐ FORM PTO 1906 is also attached.

☒ will follow.

NOTE: "If an assignment is submitted with a new application, send two separate letters-one for the application and one for the assignment." Notice of May 4, 1990 (1114 O.G. 77-78).

9. Certified Copy

Certified copy(ies) of application(s)

(country)	(appln. no.)	(filed)
(country)	(appln. no.)	(filed)
(country)	(appln. no.)	(filed)

from which priority is claimed

☐ is(are) attached.

☐ will follow.

NOTE: The foreign application forming the basis for the claim for priority must be referred to in the oath or declaration. 37 CFR 1.55(a) and 1.63.

NOTE: This item is for any foreign priority for which the application being filed directly relates. If any parent U.S. application or International Application from which this application claims benefit under 35 U.S.C. 120 is itself entitled to priority from a prior foreign application then complete item 18 on the ADDED PAGES FOR NEW APPLICATION TRANSMITTAL WHERE BENEFIT OF PRIOR U.S. APPLICATION(S) CLAIMED.

10. Fee Calculation (37 CFR 1.16)

A. ☒ Regular application

CLAIMS AS FILED					
Number filed		Number Extra		Rate	Basic Fee 37 CFR 1.16(a) \$710.00
Total Claims (37 CFR 1.16(c))	20 - 20 =	0	X	\$ 22.00	0
Independent Claims (37 CFR 1.16(b))	6 - 3 =	3	X	\$ 74.00	222.00
Multiple dependent claim(s), if any (37 CFR 1.16(d))				\$230.00	

☐ Amendment cancelling extra claims enclosed.

☐ Amendment deleting multiple-dependencies enclosed.

☐ Fee for extra claims is not being paid at this time.

NOTE: If the fees for extra claims are not paid on filing they must be paid or the claims cancelled by amendment, prior to the expiration of the time period set for response by the Patent and Trademark Office in any notice of fee deficiency. 37 CFR 1.16(d).

Filing Fee Calculation

\$ 932.00

(Application Transmittal [4-1]—page 4 of 7)

- B. ☐ **Design application**  
(\$280.00—37 CFR 1.16(f))

Filing Fee Calculation \$ \_\_\_\_\_

- C. ☐ **Plant application**  
(\$460.00—37 CFR 1.16(g))

Filing fee calculation \$ \_\_\_\_\_

11. **Small Entity Statement(s)**

- ☐ Verified Statement(s) that this is a filing by a small entity under 37 CFR 1.9 and 1.27 is(are) attached.

Filing Fee Calculation (50% of A, B or C above) \$ \_\_\_\_\_

NOTE: Any excess of the full fee paid will be refunded if a verified statement and a refund request are filed within 2 months of the date of timely payment of a full fee. 37 CFR 1.28(a).

12. **Request for International-Type Search (37 CFR 1.104(d)) (complete, if applicable)**

- ☐ Please prepare an international-type search report for this application at the time when national examination on the merits takes place.

13. **Fee Payment Being Made At This Time**

- ☐ Not Enclosed

- ☐ No filing fee is to be paid at this time. (This and the surcharge required by 37 CFR 1.16(e) can be paid subsequently.)

- ☒ Enclosed

- ☒ basic filing fee \$ 932.00

- ☐ recording assignment  
(\$40.00; 37 CFR 1.21(h)) \$ \_\_\_\_\_

- ☐ petition fee for filing by other  
than all the inventors or person  
on behalf of the inventor where  
inventor refused to sign or cannot  
be reached. (\$130.00; 37 CFR  
1.47 and 1.17(h)) \$ \_\_\_\_\_

- ☐ for processing an application with  
a specification in a non-English  
language. (\$130.00; 37 CFR 1.52(d) and  
1.17(k)) \$ \_\_\_\_\_

- ☐ processing and retention fee  
(\$130.00; 37 CFR 1.53(d) and 1.21(l))

- ☐ fee for international-type search report (\$35.00;  
37 CFR 1.21(e)). \$ \_\_\_\_\_

NOTE: 37 CFR 1.21(l) establishes a fee for processing and retaining any application which is abandoned for failing to complete the application pursuant to 37 CFR 1.53(d) and this, as well as the changes to 37 CFR 1.53 and 1.78, indicate that in order to obtain the benefit of a prior U.S. application, either the basic filing fee must be paid or the processing and retention fee of § 1.21(l) must be paid within 1 year from notification under § 53(d).

**Total fees enclosed** \$ 932.00

14. Method of Payment of Fees

- ☒ Check in the amount of \$ 932.00
- ☐ Charge Account No. \_\_\_\_\_ in the amount of \$ \_\_\_\_\_. A duplicate of this transmittal is attached.

NOTE: Fees should be itemized in such a manner that it is clear for which purpose the fees are paid. 37 CFR 1.22(b).

15. Authorization to Charge Additional Fees

**WARNING:** If no fees are to be paid on filing the following items should **not** be completed.

**WARNING:** Accurately count claims, especially multiple dependent claims, to avoid unexpected high charges, if extra claim charges are authorized.

- ☒ The Commissioner is hereby authorized to charge the following additional fees by this paper and during the entire pendency of this application to Account No. 11-1110:

- ☒ 37 CFR 1.16(a), (f) or (g) (filing fees)
- ☒ 37 CFR 1.16(b), (c) and (d) (presentation of extra claims)

NOTE: Because additional fees for excess or multiple dependent claims not paid on filing or on later presentation must only be paid or these claims cancelled by amendment prior to the expiration of the time period set for response by the PTO in any notice of fee deficiency (37 CFR 1.16(d)), it might be best not to authorize the PTO to charge additional claim fees, except possibly when dealing with amendments after final action.

- ☒ 37 CFR 1.16(e) (surcharge for filing the basic filing fee and/or declaration on a date later than the filing date of the application)
- ☒ 37 CFR 1.17 (application processing fees)

**WARNING:** While 37 CFR 1.17(a), (b), (c) and (d) deal with extensions of time under § 1.136(a) this authorization should be made only with the knowledge that: "Submission of the appropriate extension fee under 37 C.F.R. 1.136(a) is to no avail unless a request or petition for extension is filed." (Emphasis added). Notice of November 5, 1985 (1060 O.G. 27).

- ☐ 37 CFR 1.18 (issue fee at or before mailing of Notice of Allowance, pursuant to 37 CFR 1.311(b))

NOTE: Where an authorization to charge the issue fee to a deposit account has been filed before the mailing of a Notice of Allowance, the issue fee will be automatically charged to the deposit account at the time of mailing the notice of allowance. 37 CFR 1.311(b).

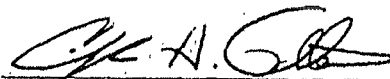
NOTE: 37 CFR 1.28(b) requires "Notification of any change in loss of entitlement to small entity status must be filed in the application . . . prior to paying, or at the time of paying, . . . issue fee". From the wording of 37 CFR 1.28(b): (a) notification of change of status must be made even if the fee is paid as "other than a small entity" and (b) no notification is required if the change is to another small entity.

16. Instructions As To Overpayment

- ☐ credit Account No. \_\_\_\_\_
- ☒ refund

Reg. No. 33,130

Tel. No. (412) 355-8641



SIGNATURE OF ATTORNEY

Christopher H. Gebhardt

Type or print name of attorney

Kirkpatrick & Lockhart

P.O. Address 1500 Oliver Building  
Pittsburgh, PA 15222



☐ **Incorporation by reference of added pages**

*Check the following item if the application in this transmittal claims the benefit of prior U.S. application(s) (including an international application entering the U.S. stage as a continuation, divisional or C-I-P application) and complete and attach the ADDED PAGES FOR NEW APPLICATION TRANSMITTAL WHERE BENEFIT OF PRIOR U.S. APPLICATION(S) CLAIMED*

- ☐ Plus Added Pages For New Application Transmittal Where Benefit Of Prior U.S. Application(s) Claimed

Number of pages added \_\_\_\_\_

- ☐ Plus Added Pages For Papers Referred To In Item 4 Above

Number of pages added \_\_\_\_\_

- ☐ Plus "Assignment Cover Letter Accompanying New Application"

Number of pages added \_\_\_\_\_

☒ **Statement Where No Further Pages Added**

*(If no further pages form a part of this Transmittal then end this Transmittal with this page and check the following item)*

- ☒ This transmittal ends with this page.



~~68/0833740~~

TITLE

SYSTEM AND METHOD FOR DISTRIBUTED STORAGE MANAGEMENT  
ON NETWORKED COMPUTER SYSTEMS

Inventors:

James R. Woodhill  
Louis R. Woodhill  
William Russell More, Jr.  
Jay Harris Berlin

Ins  
B1



BACKGROUND OF THE INVENTION

08-555,376  
~~08-555,376~~

Field of the Invention

The present invention is directed generally to a system and method for distributed storage management on a networked computer system and, more specifically, to a system and method for distributed storage management on a networked computer system including a remote backup file server and one or more local area networks in communication with the remote backup file server.

Description of the Background of the Invention

Backup copies of information stored on a computer system must be made so that if a failure occurs which causes the original copies of the data to be lost, the lost data can be recovered as it existed at the time when the last backup copy was made. Backup/restore systems have a long history on all types of computer systems from mainframes to minicomputers, local area network file servers and desktop workstations.

Historically, backup systems have operated by making copies of a computer system's files on a special backup input/output device such as a magnetic tape drive, floppy diskette drive, or optical disk drive. Most systems allow full backup, partial backup (e.g., specified drives, directories, or files), or incremental backups based on files changed after a certain date or time. Copies of files made during a backup procedure are stored on these special backup devices and are then later retrieved during a restore operation either under file names derived from the original file, from the date/time of the backup operation or from a serially-incremented number. The backup procedure is typically accomplished on an individual computer/file server basis, rather than through a single

the computer resources of two computers at most (the one processing the files to be backed up and the one with the backup device attached) are employed to effect the backup process, regardless of the actual number of computers networked together.

Today, the absolute numbers of computers networked together by organizations are increasing rapidly as is the number of different types of computers and operating systems in use. At the same time, the number of storage devices and the capacities incorporated into each of these units is growing even more rapidly. In this environment, the backup/restore approaches which have been traditionally used have become less reliable, more expensive, and more consumptive of human time and attention.

Thus, the need exists for a system designed to overcome the limitations of the existing backup/restore systems that have the following characteristics: (1) is capable of operating on a networked computer system incorporating various types of computers and operating systems; (2) is capable of accommodating a large array of large capacity storage devices; (3) is reliable; (4) is capable of operating with a minimum amount of human intervention; and (5) is relatively inexpensive.

#### SUMMARY OF THE INVENTION

The present invention is directed to a system for the distributed management of the storage space and data on a networked computer system wherein the networked computer system includes at least two storage devices for storing data files comprised of one or more binary objects. The distributed storage management system includes means for selectively copying the binary objects stored on one of the storage devices

to another of the storage devices and means for calculating a current value for a binary object identifier for selected binary objects stored on the storage devices wherein the calculation of the binary object identifier is based upon the actual data contents of the associated binary object. The distributed storage management system further includes means for storing the current value of the binary object identifier as a previous value of the binary object identifier, means for comparing the current value of the binary object identifier associated with a particular binary object to one or more previous values of the binary object identifier associated with that particular binary object and means for commanding the means for selectively copying binary objects in response to the means for comparing.

The present invention is further directed to a method for the management of the storage space and data on a computer system wherein the computer system includes at least two storage area for storing data files comprised of one or more binary objects. The storage space management method includes the following steps: (1) selectively copying the binary objects stored in one of the storage areas to another of the storage areas; (2) calculating a current value for a binary object identifier for selected binary objects stored in the storage areas wherein the calculation of the binary object identifier is based upon the actual data contents of the associated binary object; (3) storing the current value of the binary object identifier as a previous value of the binary object identifier; (4) comparing the current value of the binary object identifier associated with a particular binary object to one or more previous values of the binary object identifier associated with that particular binary object; and (5) controlling the step for

selectively copying binary objects in response to the step for comparing.

5 The system and method of the present invention for the management of the storage space on a computer system provide a backup/restore system that is capable of operating on a networked computer system incorporating various types of computers and operating systems, is capable of accommodating a large array of large capacity storage devices, is reliable, is capable of operating with a minimum amount of human  
10 intervention and is relatively inexpensive. These and other advantages and benefits of the present invention will become apparent from the description of a preferred embodiment hereinbelow.

#### BRIEF DESCRIPTION OF THE DRAWINGS

15 For the present invention to be clearly understood and readily practiced, a preferred embodiment will now be described, by way of example only, with reference to the accompanying figures wherein:

20 FIG. 1 illustrates a simplified representation of a networked computer system in which the system and method of the present invention may be employed;

25 FIG. 2 illustrates the manner in which the Distributed Storage Manager program of the present invention allocates the storage space on each of the storage devices illustrated in FIG. 1;

FIG. 3 illustrates the File Database utilized by the Distributed Storage Manager program of the present invention;

30 FIG. 4 illustrates the Backup Queue Database utilized by the Distributed Storage Manager program of the present invention; and

FIGS. 5a - 5l illustrate flow charts explaining the operation of the Distributed Storage Manager program of the present invention.

#### DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

5           FIG. 1 illustrates a simplified representation of a  
typical networked computer system 10 in which the system and  
method of the present invention for distributed storage  
management on networked computer systems may be employed. A  
remote backup file server 12 is in communication, via data path  
10       13, with a wide area network 14. The wide area network 14 is,  
in turn, in communication with a plurality of local area  
networks 16 via data paths 15. Those of ordinary skill in the  
art will recognize that any number of wide area networks 14 may  
be in communication with remote backup file server 12 and that  
15       any number of local area networks 16 (from 1 to more than 100)  
may be in communication with each wide area network 14. Those  
of ordinary skill in the art will also recognize that the means  
for communication between remote backup file server 12, wide  
area network 14 and local area networks 16 over data paths 13  
20       and 15 is well known.

Each local area network 16 includes multiple user  
workstations 18 and local computers 20 each in communication  
with their respective local area network 16 via data paths 17.  
Again, those of ordinary skill in the art will recognize that  
25       the means for communication between user workstations 18, local  
computers 20 and local area networks 16 via data paths 17 is  
well known. The storage space on each disk drive 19 on each  
local computer 20 in the networked computer system 10 is  
allocated as follows and as is shown in FIG. 2: (1) operating  
30       system files 22; (2) a Distributed Storage Manager program 24  
which embodies the system and method of the present invention

(the operation of which is described in detail hereinbelow);  
(3) a File Database 25 (the structure of which is described in  
detail hereinbelow); (4) a Backup Queue Database 26 (the  
structure of which is described in detail hereinbelow); (5)  
5 local computer data files 28; (6) free disk space 30 and (7)  
compressed storage files 32 (created by the Distributed Storage  
Manager program 24 of the present invention as is explained  
more fully hereinbelow).

The Distributed Storage Manager program 24 of the present  
10 invention builds and maintains the File Database 25 on one of  
the disk drives 19 on each local computer 20 in the networked  
computer system 10 according to the structure illustrated in  
FIG. 3. The File Database 25 stores information relating to  
each file that has been backed up by the Distributed Storage  
15 Manager program 24 since the initialization of that program on  
each local computer 20. The File Database 25 is comprised of  
three levels of records organized according to a predefined  
hierarchy. The top level record, File Identification Record  
34, includes identification information for each file that has  
20 been backed up by Distributed Storage Manager program 24. File  
Identification Record 34 contains the following elements: (1)  
Record Type 36 (identifies the file as either a directory file  
or a regular file); (2) File Location 38 (name of the directory  
in which the file resides); (3) File Name 40 (name of the  
25 file); (4) Migration Status 41 (explained more fully  
hereinbelow); and (5) Management Class 43 (explained more fully  
hereinbelow).

For each File Identification Record 34 in File Database  
25, one or more Backup Instance Records 42 are created that  
30 contain information about the file (identified by File  
Identification Record 34) at the time that the file is backed  
up. Each time that a file is backed up, a Backup Instance



Record 42 is created for that file. Each Backup Instance  
Record 42 consists of the following elements: (1) Link to File  
Identification Record 44; (2) Backup Cycle Identifier 46  
(identifies the particular backup cycle during which the Backup  
Instance Record 42 is created); (3) File Size 48; (4) Last  
Modified Date/Time 50; (5) Last Access Date/Time 52; (6) File  
Attributes 54 (e.g., read-only, system, hidden); (7) Delete  
Date 56 (date on which the file was deleted); and (8) Insert  
Date 57 (date on which the Backup Instance Record 42 was  
created).

Associated with each Backup Instance Record 42 is one or  
more Binary Object Identification Records 58. The Distributed  
Storage Manager program 24 views a file as a collection of data  
streams. A data stream is defined as a distinct collection of  
data within the file that may be changed independently from  
other distinct collections of data within the file. For  
example, a file may contain its normal data and may also  
contain extended attribute data. A user may change the  
extended attribute data without modifying any of the normal  
data or vice versa. The Distributed Storage Manager program 24  
further divides each data stream into one or more binary  
objects. If the size of the data stream is equal to or less  
than a previously defined convenient maximum binary object size  
(currently one (1) megabyte), then a single binary object  
represents the data stream. If the data stream is larger than  
the maximum binary object size, then the Distributed Storage  
Manager program 24 divides the data stream into multiple binary  
objects, all but the last of which are equal in size to the  
maximum binary object size. A Binary Object Identification  
Record 58 is created for each binary object that comprises the  
file which was backed up during the backup cycle identified by  
the Backup Cycle Identifier 46 of a particular Backup Instance

Record 42. Each Binary Object Identification Record 58 includes the following components: (1) Link to Backup Instance Record 60; (2) Binary Object Stream Type 62 (e.g., data, extended attributes, security); (3) Binary Object Size 64; (4) Binary Object CRC32 66 (explained more fully hereinbelow); (5) Binary Object LRC 68 (explained more fully hereinbelow); (6) Binary Object Hash 70 (explained more fully hereinbelow); and (7) Binary Object Offset 72 (explained more fully hereinbelow). The Binary Object Size 64, Binary Object CRC32 66, Binary Object LRC 68 and Binary Object Hash 70 comprise the Binary Object Identifier 74 which is a unique identifier for each binary object to be backed up and is discussed in more detail below.

The Distributed Storage Manager program 24 also builds and maintains the Backup Queue Database 26 on one of the disk drives 19 on each local computer 20 in the networked computer system 10 according to the structure illustrated in FIG. 4. Each entry (Backup Queue Record 75) in the Backup Queue Database 26 is comprised of the following components: (1) Record Type 76 (identifies the file as either a directory file or a regular file); (2) File Location 78 (name of the directory in which the file resides); (3) File Name 80 (name of the file); (4) File Status 82 ("new", "modified" or "deleted"); (5) File Size 84; (6) Last Modified Date/Time 86; (7) Last Access Date/Time 88; (8) File Attributes 90 (e.g., read-only, system, hidden); and (9) File Priority 92 (explained more fully hereinbelow).

The operation of the Distributed Storage Manager program 24 may be illustrated by way of the flow charts depicted in FIGS. 5a through 5l. For explanation purposes, the Distributed Storage Manager program 24 is divided into several distinct

skill in the art will recognize, however, that each of the distinct functions operates in cooperation with the other functions to form a unitary computer program. Those of ordinary skill in the art will also recognize that the following discussion illustrates the operation of the Distributed Storage Manager program 24 on a single local computer 20, although it should be understood that the Distributed Storage Manager program 24 operates in the same fashion on each local computer 20 on the networked computer system 10. The Distributed Storage Manager program 24 can either be executed on user demand or can be set to execute periodically on a user-defined schedule.

#### 1. Identification of Binary Objects to be Backed Up

In the flow chart of FIG. 5a, execution of the Distributed Storage Manager program 24 begins at step 100 where the Backup Queue Database 26 is built by creating a Backup Queue Record 75 for each File Identification Record 34 found in File Database 25. In this way, a list of files that were backed up during the previous backup cycle is established so that it can be determined which files need to be backed up during the current backup cycle. To create each Backup Queue Record 75, the Backup Instance Record 42 representing the most recent backup of the file represented by each File Identification Record 34 is located. This determination is made by examining the Backup Cycle Identifier 46 in each Backup Instance Record 42. The Backup Cycle Identifier 46 may represent either a date (month/day/year) or numerical value assigned to a particular backup cycle. The Backup Queue Record 75 is comprised of certain of the data fields of both the File Identification Record 34 and the Backup Instance Record 42. During the process of creating each Backup Queue Record 75 the File

Status field 82 is set to "DELETED". However, if the Delete Date field 56 of the most recent Backup Instance Record 42 associated with the File Identification Record 34 currently being processed is non-zero, indicating that the file has been previously deleted, then no Backup Queue Record 75 is created for that File Identification Record 34. If the backup that is currently being processed for the local computer 20 is not a full backup (i.e., all files on all disk drives 19 on the local computer 20), then the Distributed Storage Manager program 24 will only create Backup Queue Records 75 for those files that match the backup specifications. For example, if only those files that have a file extension of ".EXE" are to be backed up, then only File Identification Records 34 that correspond to ".EXE" files will be processed.

Program control then continues with step 102 where the Distributed Storage Manager program 24 of the present invention scans all disk drives 19 on the local computer 20 that are to be backed up. This operation consists of scanning the directory hierarchy on each disk drive 19 on the local computer 20 and returning to the Distributed Storage Manager program 24 certain file block information for each of the directory files and regular files that are stored on the disk drives 19 to be backed up. A typical computer operating system maintains a file block for each file stored on the system which includes information such as file location, file type, user-assigned file name, file size, creation date and time, modify date and time, access date and time and file attributes. This operation may be controlled by some parameters that indicate which drives, directories and files are to be backed up during a backup operation. However, the default operation is to back up all files on all disk drives 19 on the local computer 20. Program control then continues with step 103 where the

Distributed Storage Manager program 24 determines whether the file block information for an additional file has been located on the disk drives 19. If an additional file has been located, program control continues with step 104. If an additional file  
5 has not been located, program control continues with step 116.

In step 104, the Distributed Storage Manager program 24 determines whether a Backup Queue Record 75 exists for the located file by comparing the file's file block information to the information stored in Backup Queue Database 26. If such a  
10 Backup Queue Record 75 does not exist (i.e., this is the first time this file will be backed up), program control continues with step 106 where a Backup Queue Record 75 for the file is created using the information contained within the file's file block. The File Status field 82 for the newly created Backup  
15 Queue Record 75 is set to "NEW". Program control then continues with step 108 where a user-defined priority is assigned to the file and stored in the File Priority field 92 of the Backup Queue Record 75. This user-defined priority may be assigned to the file by methods that are well-known to those  
20 of ordinary skill in the art. The use of the File Priority field 92 by the Distributed Storage Manager program 24 is discussed in more detail hereinbelow. Program control is then returned to step 102.

If the Distributed Storage Manager program 24 determines,  
25 in step 104, that a Backup Queue Record 75 exists in the Backup Queue Database 26 for the located file, program control continues with step 110 where it is determined whether any change has been made to the file. This determination is made by comparing the information in the file's file block with the  
30 information stored in the file's Backup Queue Record 75. If any of the values have changed, program control continues with step 112 where File Status field 82 is set to "MODIFIED" and

the fields in the Backup Queue Record 75 are updated from the file's file block information. Program control then continues with step 108 where a user-defined priority is assigned to the file and stored in File Priority field 92; program control is then returned to step 102. If the determination is made in step 110 that no change has been made to the file, then, in step 114, the Backup Queue Record 75 is deleted from the Backup Queue Database 26 since the file does not need to be backed up. Following step 114, program control is returned to step 102.

If the Distributed Storage Manager program 24 determines, in step 103, that an additional file has not been located, program control continues with step 116. In step 116, the Distributed Storage Manager program 24 reads each Backup Queue Record 75 in Backup Queue Database 26, one at a time. The Backup Queue Records 75 in Backup Queue Database 26 represent all of the files that must be backed up by the Distributed Storage Manager program 24 during the present backup cycle. Program control continues with step 117 where the Distributed Storage Manager program 24 determines whether a next Backup Queue Record 75 has been located in Backup Queue Database 26. If a next Backup Queue Record 75 has been located, program control continues with step 118; otherwise, program control continues with step 119, where the routine illustrated by the flow chart of FIG. 5a is terminated. In step 118, the Distributed Storage Manager program 24 determines whether the File Status field 82 in the Backup Queue Record 75 currently being processed is set to "DELETED". If the File Status field 82 is set to "DELETED", program control continues with step 120 where the Delete Date field 56 in the most recent Backup Instance Record 42 associated with the file identified by the Backup Queue Record 75 currently being processed is set to the current date. A list of all Binary Object Identification

Records 58 associated with the Backup Instance Record 42 for the file identified by the Backup Queue Record 75 currently being processed is placed in a delete queue (not shown) that will be used by Distributed Storage Manager program 24 to delete all Binary Object Identification Records 58 for binary objects that have been deleted from the disk drives 19 of local computer 20. Program control then continues with step 122 where the Backup Queue Record 75 currently being processed is deleted from the Backup Queue Database 26. Program control is then returned to step 116.

If the Distributed Storage Manager program 24 determines, in step 118, that the File Status field 82 of the Backup Queue Record 75 currently being processed is not set to "DELETED", program control continues with step 124 where the Distributed Storage Manager program 24 determines whether the File Status field 82 of the Backup Queue Record 75 currently being processed is set to "NEW". If the File Status field 82 is set to "NEW", program control continues with step 126 where a File Identification Record 34 is created in File Database 25 using the information stored in the Backup Queue Record 75 currently being processed. Program control then continues with step 130. If the Distributed Storage Manager program 24 determines, in step 124, that the File Status field 82 of the Backup Queue Record 75 currently being processed is not set to "NEW" (i.e., the file has been modified since the last backup cycle), program control continues with step 128 where the File Identification Record 34 associated with the file identified by the Backup Queue Record 75 currently being processed is located in the File Database 25. Program control then continues with step 130. In step 130, the Distributed Storage Manager program 24 creates a new Backup Instance Record 42 in the File Database 25 for the file identified by the Backup Queue Record 75

currently being processed. The Backup Instance Record 42 is created using information stored in the associated File Identification Record 34 and the Backup Queue Record 75 currently being processed. The Backup Cycle Identifier 46 is set to indicate that the file is to be backed up during the current backup cycle. The Delete Date field 56 is initialized to "zero". The Insert Date field 57 is set to the current date.

Program control then continues with step 132 where the Distributed Storage Manager program 24 separates the file identified by the Backup Queue Record 75 currently being processed into its component data streams. Each data stream is then processed individually. Those of ordinary skill in the art will recognize that these data streams may represent regular data, extended attribute data, access control list data, etc. Program control continues with step 134 where the Distributed Storage Manager program 24 determines whether each of the data streams currently being processed is larger than the maximum binary object size (currently one (1) megabyte). If the data stream is larger than one (1) megabyte, program control continues with step 136 where the data stream currently being processed is segmented into multiple binary objects smaller in size than one (1) megabyte. Either following step 136 or, if the determination is made in step 134 that the data stream currently being processed is not larger than one (1) megabyte (and, thus, the data stream is represented by a single binary object), program control continues with step 138.

In step 138, a Binary Object Identification Record 58 is created in File Database 25 for each of the binary objects currently being processed. Each of these Binary Object Identification Records 58 are associated with the Backup Instance Record 42 created in step 130. The Binary Object



Identifier 74 portion of each Binary Object Identification Record 58 is comprised of the Binary Object Size field 64, the Binary Object CRC32 field 66, the Binary Object LRC field 68 and the Binary Object Hash field 70. Each of the fields of the Binary Object Identifier 74 may be four (4) bytes in length and is calculated from the contents of each binary object. The Binary Object Size field 64 may be set equal to the byte-size of the binary object. The Binary Object CRC32 field 66 may be set equal to the standard 32-bit Cyclical Redundancy Check number calculated against the contents of the binary object taken one (1) byte (8 bits) at a time. Those of ordinary skill in the art will readily recognize the manner in which the Cyclical Redundancy Check number is calculated. The Binary Object LRC field 68 may be set equal to the standard Longitudinal Redundancy Check number calculated against the contents of the binary object taken four (4) bytes (32 bits) at a time using the following algorithm:

```
BINARY OBJECT LRC = (initialized value)
for each double word (32 bits) of the binary object data:
    LRC = LRC (XOR) double word of binary object data
end loop
```

The Binary Object Hash field 70 is calculated against the contents of the binary object taken one (1) word (16 bits) at a time using the following algorithm:

```
HASH = (initialized value)
for each word (16 bits) of the binary object:
    rotate current HASH value by 5 bits
    HASH = HASH + 1
    HASH = HASH + (current word (16 bits) of binary object)
end loop
```

Since the Binary Object Identifier 74 is used to uniquely identify a particular binary object, it is important that the possibility of two different binary objects being assigned the

same Binary Object Identifier 74 be very small. This is the reason for implementing the Binary Object Identifier 74 using 128 bits and four separate calculations. Although a Binary Object Identifier 74 may be calculated in various ways, the key notion is that the Binary Object Identifier is calculated from the contents of the data instead of from an external and arbitrary source. By incorporating the Binary Object Size field 64 within the Binary Object Identifier 74, only binary objects that are exactly the same size can generate duplicate Binary Object Identifiers 74. Further, the calculations used to determine the Binary Object CRC32 field 66, the Binary Object LRC field 68 and the Binary Object Hash field 70 are relatively independent of each other. Using the calculations set forth above, the probability that the Distributed Storage Manager program 24 will generate the same Binary Object Identifier 74 for two different binary objects is extremely low. Those of ordinary skill in the art will recognize that there exist many different ways of establishing the Binary Object Identifier 74 (e.g., establishing a Binary Object Identifier 74 of a different length or utilizing different calculations) and that the procedure set forth above is only one way of establishing the Binary Object Identifier 74. The critical feature to be recognized in creating a Binary Object Identifier 74 is that the identifier should be based on the contents of the binary object so that the Binary Object Identifier 74 changes when the contents of the binary object changes. In this way, duplicate binary objects, even if resident on different types of computers in a heterogeneous network, can be recognized from their identical Binary Object Identifiers 74.

Program control then continues with step 140 where the Distributed Storage Manager program 24 identifies which binary

objects must be backed up during the current backup cycle. If the File Status field 82 of the Backup Queue Record 75 currently being processed is set to "NEW", then all binary objects associated with the file identified by the Backup Queue Record 75 currently being processed must be backed up during the current backup cycle. If the File Status field 82 is set to "MODIFIED", then only those binary objects associated with the file that have changed must be backed up. Those binary objects that have changed are identified by comparing the Binary Object Identifiers 74 calculated in step 138 with the corresponding Binary Object Identifiers 74 associated with the next most recent Backup Instance Record 42 for the file identified by the Backup Queue Record 75 currently being processed. The Binary Object Identifiers 74 calculated in step 138 are compared against their counterparts in the File Database 25 (e.g., the Binary Object Identifier 74 (as calculated in step 138) that identifies the first binary object in the file (as determined by the Binary Object Stream Type field 62 and the Binary Object Offset field 72) is compared to the Binary Object Identifier 74 (associated with the next most recent Backup Instance Record 42) for the first binary object in the file). This procedure allows the Distributed Storage Manager program 24 to determine which parts of a file have changed and only back up the changed data instead of backing up all of the data associated with a file when only a small portion of the file has been modified. Program control is then returned to step 116.

## **2. Concurrent Onsite/Offsite Backup**

The Distributed Storage Manager program 24 performs two concurrent backup operations. In most cases, the Distributed Storage Manager program 24 stores a compressed copy of every

binary object it would need to restore every disk drive 19 on every local computer 20 somewhere on the local area network 16 other than on the local computer 20 on which it normally resides. At the same time, the Distributed Storage Manager program 24 transmits every new or changed binary object to the remote backup file server 12. Binary objects that are available in compressed form on the local area network 16 can be restored very quickly while the much greater storage capacity on the remote backup file server 12 ensures that at least one copy of every binary object is stored and that a disaster that destroys an entire site would not destroy all copies of that site's data.

The Concurrent Onsite/Offsite Backup routine begins at step 200 of the flow chart illustrated in FIG. 5b where the Distributed Storage Manager program 24 compiles a list of those binary objects that are to be backed up during the current backup cycle. Those binary objects which must be backed up during the current backup cycle are identified in step 140 of the flow chart of FIG. 5a. Those of ordinary skill in the art will recognize, however, that the Concurrent Onsite/Offsite Backup routine may be performed independently of the routine illustrated in FIG. 5a. Program control then continues with step 202 where the Distributed Storage Manager program 24 identifies whether there are any additional binary objects to be processed. If no additional binary objects are to be processed, program control is transferred to step 204 where the Concurrent Onsite/Offsite Backup routine is terminated. Otherwise, program control continues with step 206 where the binary object currently being processed is compressed and stored in a compressed storage file 32 (FIG. 2) on one of the disk drives 19 on a local computer 20 on the local area network 16 other than the local computer 20 on which the binary object

is currently stored. The compressed storage file 32 is used to allow the Distributed Storage Manager program 24 to pack several smaller compressed binary objects into a larger unit for storage. This is required to reduce the number of files that the Distributed Storage Manager program 24 must manage and to ensure that the Distributed Storage Manager program 24 does not create many "small" files since most file systems allocate some minimum amount of space to store a file even if the actual file contains less data than the allocated space. The purpose behind storing the backup copy of a binary object on a disk drive 19 on a different local computer 20 is to ensure that if the first disk drive 19 or local computer 20 fails, the backup copies of the binary objects are not lost along with the original copies of the binary objects.

Program control then continues with step 208 where each compressed storage file 32, when it reaches a maximum manageable size (e.g., two (2) megabytes), is transmitted to the remote backup file server 12 (FIG. 1) over wide area network 14 for long-term storage and retrieval. Upon arrival of the compressed storage file 32 at the remote backup file server 12, software resident on the remote backup file server 12 routes the compressed storage file 32 for ultimate storage to magnetic tape or other low cost storage media. The backup copy of a binary object which is maintained in the compressed storage file 32 on one of the disk drives 19 on one of the local computers 20 is only the most recent version of each binary object that is backed up while the backup copy of the binary object stored on the remote backup file server 12 is kept until it is no longer needed. Since most restores of files on a local area network 16 consist of requests to restore the most recent backup version of a file, the local copies of binary objects serve to handle very fast restores for most

restore requests that occur on the local area network 16. If the local backup copy of a file does not exist or a prior version of a file is required, it must be restored from the remote backup file server 12. Program control then continues with step 210 where the Distributed Storage Manager program 24 determines whether sufficient space is available in the space allocated for compressed storage files 32 on the disk drives 19 on local computers 20 for storage of the binary object currently being processed. If sufficient space is available, program control is returned to step 200. Otherwise, the binary object currently being processed is deleted from the disk drive 19 on which it was stored after transmission to the remote backup file server 12 has been completed. Program control is then returned to step 200.

### 3. File Prioritization

The file prioritization process performed by the Distributed Manager Storage program 24 is handled by four interrelated routines of that program: (1) Backup/Restore Routine; (2) Compression Routine; (3) Local Storage Routine; and (4) Resource Allocation Routine. Each routine will be described in turn. In the following discussion, when one of the four routines is discussed, it should be understood that it is the Distributed Storage Manager program 24 that is executing the functions of that routine. The Backup/Restore Routine, the Local Storage Routine and the Compression Routine may be executed on each of the local computers 20 on the networked computer system 10 while the Resource Allocation Routine is executed on only one of the local computers 20 on the networked computer system 10. This execution scheme permits the resources of any available local computer 20 on any of the local area networks 16 to be utilized according to its

availability. Furthermore, more than one local computer 20 may be utilized to complete any high-priority tasks required to be completed within a specified time frame. An advantage of the process of prioritization of files is that it allows the  
5 Distributed Storage Manager program 24 to effectively deal with a situation where local storage and wide area network transmission resources are limited. The Distributed Storage Manager program 24 is also able to keep track of data which is not stored locally or transmitted to the remote backup file  
10 server 12 in any given backup cycle and then attempt to resume these processes during the next backup cycle.

The Backup/Restore Routine is illustrated in the flow chart shown in FIG. 5c. In step 300, the Distributed Storage Manager program 24 initiates the Backup/Restore Routine by  
15 locating the highest priority binary object scheduled for backup on the local computer 20 on which the Backup/Restore Routine is executing. The identities of the binary objects to be backed up and their respective priorities were determined by the Distributed Storage Manager program 24 in the flow chart of  
20 FIG. 5a. Those of ordinary skill in the art will recognize, however, that the file prioritization routines of the Distributed Storage Manager program 24 may be utilized independently of the process illustrated in the flow chart of FIG. 5a. Program control then continues with step 302 where  
25 the Backup/Restore Routine of the Distributed Storage Manager program 24 determines whether a binary object to be backed up has been located. If not, program control continues with step 304 where the Backup/Restore Routine is terminated. Otherwise, program control continues with step 306 where the  
30 Backup/Restore Routine of the Distributed Storage Manager program 24 sends a message to the Resource Allocation Routine indicating the priority of the highest priority binary object

that the Backup/Restore Routine has located. Program control then continues with step 308 where the Backup/Restore Routine waits for a message from the Resource Allocation Routine indicating which Compression Routine is available to compress and store the highest priority binary object located by the Backup/Restore Routine. In this way, the Distributed Storage Manager program 24 is able to perform not only local computer based file prioritization but also networked computer system based file prioritization. This is accomplished by having the Resource Allocation Routine examine the priority of the highest priority binary object located by each Backup/Restore Routine and then allocating compression resources to the Backup/Restore Routine which has the highest priority binary object to compress.

Program control continues with step 310 where the Backup/Restore Routine receives a message from the Resource Allocation Routine indicating that a Compression Routine is available for binary object compression. The Backup/Restore Routine then sends a list of up to forty (40) binary objects or up to one (1) megabyte of uncompressed binary objects to the Compression Routine starting with the highest priority binary object that the Backup/Restore routine has identified for backup. The reason to limit the number or size of binary objects that are sent to a Compression Routine is to allow the Compression Routine to work for a limited amount of time on the binary objects for one Backup/Restore Routine before becoming available to work on another Backup/Restore Routine's binary objects.

The Compression Routine performed by the Distributed Storage Manager program 24 is illustrated in the flow chart depicted in FIG. 5d. Program control begins at step 312 where the Compression Routine of the Distributed Storage Manager



program 24 sends a message to the Resource Allocation routine indicating that the Compression Routine is available to compress binary objects. Program control then continues with step 314 where the Compression Routine waits for a compress message from a Backup/Restore Routine indicating which binary objects are to be compressed. The compress message includes the File Name 40 from Identification Record 34, the Binary Object Stream Type field 62 from the Binary Object Identification Record 58 and the Binary Object Offset field 72 from the Binary Object Identification Record 58 for each binary object to be compressed. The binary objects are placed in a compression queue (not shown) for processing by the Compression Routine. Program control then continues with step 316 where the Compression Routine sends a message to the Resource Allocation Routine to determine which Local Storage Routine has space available for storage of compressed binary objects. Program control then continues with step 318 where the Compression Routine requests allocation of a compressed storage file 32 from the Local Storage Routine that has indicated availability of storage space. Program control continues with step 320 where the binary object is compressed and stored in the allocated compressed storage file 32. Program control then continues with step 322 where the Compression Routine determines whether there are more binary objects in the compress queue. If there are no more binary objects present in the compress queue, program control returns to step 312. If more binary objects are present, program control continues with step 324 where the Compression Routine determines whether the allocated compressed storage file 32 is full. If not, program control is returned to step 320. Otherwise, program control is returned to step 316 where the Compression Routine sends a message to the Resource Allocation Routine to determine which

Local Storage Routine has space available for storage of compressed binary objects.

5       The Local Storage Routine executed by the Distributed Storage Manager program 24 is illustrated in the flow chart depicted in FIG. 5e. The Local Storage Routine is responsible for managing storage file space on a particular local computer 20. Program control beings at step 326 where the Local Storage Routine of Distributed Storage Manager program 24 sends a message to the Resource Allocation Routine indicating the amount of storage space it has available for allocation of compressed storage files 32. The Local Storage Routine determines the amount of space it has available for allocation of compressed storage files 32 by determining the total amount of free space on its disk drives 19 and then determining how  
10       must space must be left as "free space". The amount of required "free space" is user-specified. Program control continues with step 328 where the Local Storage Routine waits for a request from a Compression Routine for allocation of a compressed storage file 32. Upon receipt of such a request,  
15       program control continues with step 330 where the requested compressed storage file 32 (e.g., two (2) megabytes in size) is allocated. The Local Storage Routine then returns a message to the requesting Compression Routine indicating the name and location of the compressed storage file 32 that has been  
20       allocated. Program control is then returned to step 326.

25       The Resource Allocation Routine performed by the Distributed Storage Manager program 24 of the present invention is depicted in the flow chart of FIG. 5f. The Resource Allocation Routine is a process that responds to messages from  
30       other routines of the Distributed Storage Manager program 24 and allocates resources between resource requesters and resource providers. Program control begins with step 332 where

the Resource Allocation Routine executed by the Distributed Storage Manager program 24 waits for a message from a Distributed Storage Manager program 24 routine. When a message is received, program control continues with step 334 where the Resource Allocation Routine determines whether the message is from a Backup/Restore Routine transmitting information relating to its highest priority binary object for compression. If such a message is received, program control continues with step 336 where the Resource Allocation Routine stores this information in an internal table containing Backup/Restore Routine status information. The Resource Allocation Routine then scans this status information table to ascertain which Backup/Restore Routine has the highest priority binary object for storage. Program control then continues with step 338 where the Resource Allocation Routine determines whether any Compression Routine is available to process the highest priority binary object. If no Compression Routine is available for processing, program control is returned to step 332. If an available Compression Routine is located, program control continues with step 340 where the Resource Allocation Routine transmits a message to the requesting Backup/Restore Routine indicating which Compression Routine is available to compress the binary object. In addition, the Resource Allocation Routine marks the Compression Routine as "working" in an internal table containing Compression Routine information. Program control is then returned to step 332.

If the Resource Allocation Routine determines, in step 334, that the received message is not from a Backup/Restore Routine, program control continues with step 342 where the Resource Allocation Routine determines whether the received message is from a Compression Routine indicating that the transmitting Compression Routine is available for processing.

If the received message is from a Compression Routine, program control continues with step 344 where the Resource Allocation Routine marks the transmitting Compression Routine as "available" in its internal table containing Compression Routine information. Program control then continues with step 336.

If the Resource Allocation Routine determines, in step 342, that the received message has not been transmitted from a Compression Routine indicating its availability for processing, program control continues with step 346 where the Resource Allocation Routine determines whether the received message is from a Local Storage Routine indicating the amount of storage space that the Local Storage Routine has available. If the received message is from a Local Storage Routine, Program control continues with step 348 where the Resource Allocation Routine locates the transmitting Local Storage Routine in an internal table containing Local Storage Routine information and saves the storage space information transmitted by the Local Storage Routine. Program control then continues with step 354.

If the Resource Allocation Routine determines, in step 346, that the received message is not from a Local Storage Routine, program control continues with step 350 where the Resource Allocation Routine determines whether the received message is from a Compression Routine requesting a compressed storage file 32. If the Resource Allocation Routine determines that such a message was received, program control continues with step 352 where the identity of the requesting Compression Routine is added to a "space request list". Program control then continues with step 354. If the Resource Allocation Routine determines, in step 350, that the received message is not from a Compression Routine requesting a compressed storage file 32,

In step 354, the Resource Allocation Routine determines whether any Compression Routines are waiting for allocation of a compressed storage file 32 by examining the "space request list". If no such Compression Routines are in the "space request list", program control is returned to step 332. If the identity of such a Compression Routine is found in the "space request list", program control continues with step 356 where the Resource Allocation Routine determines whether any of the Local Storage Processors has any available space by examining the information in its internal table containing Local Storage Routine information. When a Compression Routine requests a compressed storage file 32, the Compression Routine identifies the name of the local computer 20 on which the Backup/Restore Routine is executing and on whose behalf it is compressing binary objects. This allows the Compression Routine to request a compressed storage file 32 on a local computer 20 other than the local computer 20 on which the binary object to be stored resides. Otherwise, the backup copy of the binary object may be stored on the same local computer 20 as the original binary object whereby a disk drive failure would result in losing both the original and backup copies. The Resource Allocation Routine uses the information supplied by the Compression Routine to ensure that the requested compressed storage file 32 is allocated on a local computer 20 other than the local computer 20 from which the binary object originated. If available storage space is located, program control continues with step 358 where the Resource Allocation Routine transmits a message to the next Compression Routine in the "space request list" indicating which Local Storage Routine has allocated an available compressed storage file 32. Program control is then returned to step 354.

If the Resource Allocation Routine determines, in step 356, that no storage space is available, program control continues with step 360 where the Resource Allocation Routine determines whether there are any compressed storage files 32 that are maintained by any of the Local Storage Routines which have a lower priority than the binary object currently being processed. If so, program control continues with step 362 where a message is transmitted to the Local Storage Routines instructing the Local Storage Routines to delete some of the low-priority compressed storage files 32 to make room for higher priority binary objects. After these lower-priority compressed storage files 32 are deleted by the Local Storage Routines, the Local Storage Routines will transmit new status messages to the Resource Allocation Routine. Program control is then returned to step 332. If no lower-priority compressed storage files 32 are located in step 360, program control continues with step 364 where the Resource Allocation Routine transmits a message to the Local Storage Routines with instructions that from that time forward, any allocated compressed storage files 32 are to be deleted after the contents of the compressed storage files 32 have been successfully transmitted to the remote backup file server 12 for long-term storage. Program control is then returned to step 332.

#### 4. Granularization of Files

The most important class of "large" files on computer systems such as networked computer system 10 is databases. Typically, on a given day, only a small percentage of the data in a large database is changed by the users of that database. However, it is likely that some data will be changed in each one of the (1) megabyte binary object segments that are created

in step 136 of the flow chart depicted in FIG. 5a. As a result, in most cases, the entire "large" database file would have to be backed up to the remote backup file server 12. However, the Distributed Storage Manager program 24 of the present invention utilizes a technique of subdividing large database files into "granules" and then tracks changes from the previous backup copy at the "granule" level. The "granule" size utilized by the Distributed Storage Manager program 24 may be one (1) kilobyte although those of ordinary skill in the art will recognize that any "granule" size that produces the most efficient results (in terms of processing time and amount of data that must be backed up) may be utilized. This technique of subdividing files into "granules" is only used to reduce the amount of data that must be transmitted to the remote backup file server 12 and is not utilized in making backup copies of binary objects for storage on local computers 20.

The operation of the Distributed Storage Manager program 24 in subdividing files into "granules" is illustrated in the flow chart depicted in FIG. 5g. This "granularization" procedure is performed for "large" files following step 136 of the flow chart of FIG. 5a. Program control begins at step 400 where the Distributed Storage Manager program 24 identifies whether the binary object currently being processed is a segment of a "large" database-like file. Program control then continues with step 402 where the Distributed Storage Manager program 24 determines whether this is the first time that the binary object currently being processed is being backed up using the "granularization" technique. If so, program control continues with step 404 where the Distributed Storage Manager program 24 creates a "shadow file" which contains a "contents identifier" for each "granule" in the binary object currently being processed. Each "contents identifier" is composed of a

standard 32-bit Cyclical Redundancy Check number which is calculated against the contents of the "granule" and a 32-bit hash number which is calculated against the contents of the "granule" in the same manner described in relation to step 138 of the flow chart depicted in FIG. 5a. Those of ordinary skill in the art will readily recognize the manner in which the Cyclical Redundancy Check number is calculated. Each time that the binary object is to be backed up, the Distributed Storage Manager program 24 can calculate the "contents identifier" for each "granule" in the binary object and then compare it to the "contents identifier" of the "granule" the last time the binary object was backed up and determine if the "granule" has changed. This allows the Distributed Storage Manager program 24 to determine what data within a binary object has changed and only back up the changed data instead of the entire binary object. Program control then continues with step 406 where the Distributed Storage Manager program 24 calculates a "change identifier" for each "granule" of the binary object and stores it in the "shadow file" for that binary object. Program control then continues with step 408 where the binary object is compressed into a compressed storage file 32 which becomes the most recent complete copy of the binary object for later reconstitution of the binary object as is discussed more fully hereinbelow. The contents of the compressed storage file 32 is then transmitted to the remote backup file server 12 for long-term storage and retrieval. Program control is then returned to step 400.

If the Distributed Storage Manager program 24 determines, in step 402, that this is not the first time that the binary object currently being processed is being backed up using the "granularization" technique, program control continues with step 410 where the Distributed Storage Manager program 24



calculates the "contents identifier" for each "granule".

Program control continues with step 412 where each newly-calculated "contents identifier" is compared to the

corresponding "contents identifier" for the "granule" in the

5 "shadow file". If the two values are equal, program control continues with step 414 where the Distributed Storage Manager program 24 determines whether the last "granule" of the binary object has been processed. If so, program control is returned to step 400; otherwise, program control continues at step 410.

10 If the "contents identifiers" are not found to be equal in step 412, the "granule" has changed and program control continues with step 416 where the "shadow file" is updated with the newly-calculated "contents identifier" for the "granule".

Program control then continues with step 418 where the changed  
15 "granule" is compressed into a compressed storage file 32 using a special format that identifies the "granule". All changed "granules" for the "data stream" currently being processed are packed together in the same compressed storage file 32. The contents of the compressed storage file 32 is then transmitted  
20 to the remote backup file server 12 for long-term storage and retrieval. If the Distributed Storage Manager program 24 determines that a large percentage of the "granules" in the binary object have changed (e.g., 80%), then the entire binary object is backed up to the remote backup file server 12.

25 The operation of the Distributed Storage Manager program 24 in reconstituting, on a local computer 20, a binary object that has been transmitted to the remote backup file server 12 using the "granularization" technique illustrated in FIG. 5g is illustrated in the flow chart depicted in FIG. 5h. Program  
30 control begins at step 420 where the Distributed Storage Manager program 24 creates a work area on the remote backup file server 12 that is equal in size to the total uncompressed

size of the binary object that is to be reconstituted. Program control continues with step 422 where the most recent complete copy of the binary object to be reconstituted is located on the remote backup file server 12 and is decompressed into the work area. Program control then continues with step 424 where the Distributed Storage Manager program 24 creates a bitmap with one bit representing each granule of the binary object to be reconstituted. Initially, all bits in this bitmap are set to zero (0). Each bit in the bitmap is used to indicate whether the granule associated with that bit has been restored to the most recent complete copy of the binary object. Program control then continues with step 426 where the Distributed Storage Manager program 24 locates the most recent "granularized" copy of the binary object that was stored on the remote backup file server 12. Each time that step 426 is executed, the next most recent "granularized" copy of the binary object is located. This process continues until all bits in the bitmap are set to one (1) or until there are no more "granularized" copies of the binary object that are newer than the most recent complete copy of the binary object. At that point, the binary object will have been reconstituted and will be ready to be restored to the local computer 20. Following step 426, program control continues with step 428 where the Distributed Storage Manager program 24 determines whether another "granularized" copy of the binary object has been located. If so, program control continues with step 430 where the Distributed Storage Manager program 24 obtains the list of "granules" in the "granularized" copy of the binary object just located. If another "granularized" copy of the binary object is not located in step 428, program control continues with step 438 where the reconstituted binary object is restored to the local computer 20. Following step 430,

program control continues with step 432 where, starting with the first "granule" in the "granularized" copy of the binary object, the Distributed Storage Manager program 24 determines whether the bit for this "granule" in the bit map is set to zero (0). If the bit is set to one (1), a more recent copy of the "granule" has already been decompressed and copied into the work area. If the bit is set to zero (0), program control continues with step 434 where the "granule" is decompressed and copied into the work area at the correct location for that "granule". After copying the "granule" to the work area, the Distributed Storage Manager program 24 sets the bit within the bitmap for the "granule" to one (1). If the Distributed Storage Manager program 24 determines, in step 432, that the bit is not set to zero (0), program control continues with step 440 where the Distributed Storage Manager program 24 determines whether there are any more "granules" to be processed in the current set of "granules". If so, program control is returned to step 432; otherwise, program control is transferred to step 426. Following step 434, program control continues with step 436 where the Distributed Storage Manager program 24 determines whether all bits in the bitmap are now set to one (1). If so, program control continues with step 438 where the reconstituted binary object is restored to the local computer 20. If the Distributed Storage Manager program 24 determines, in step 436, that all bits in the bitmap are not set to one (1), program control continues with step 440.

The technique of "granularizing" "large" files also becomes useful when a current version of a file (comprised of current versions of binary objects) must be restored to a previous version of that file (comprised of previous versions of binary objects). Each binary object comprising the current version of the file can be restored to the binary object

comprising the previous version of the file by restoring and updating only those "granules" of the current version of the binary objects that are different between the current and previous versions of the binary objects. This technique is illustrated in the flow chart depicted in FIG. 5i. Program control begins at step 442 where the Distributed Storage Manager program 24 obtains from the user the identities of the current and previous versions of the file (comprised of binary objects) which needs to be restored. Program control continues with step 443 where the Distributed Storage Manager program 24 compiles a list of all binary objects comprising the current version of the user-specified file. This information is obtained from File Database 25. Program control then continues with step 444 where the Distributed Storage Manager program 24 calculates "contents identifiers" for each "granule" within the current version of each binary object as it exists on the local computer 20. Program control then continues with step 446 where the Distributed Storage Manager program 448 transmits an "update request" to the remote backup file server 12 which includes the Binary Object Identification Record 58 for the previous version of each binary object as well as the list of "contents identifiers" calculated in step 444. Program control continues with step 448 where the Distributed Storage Manager program 24 reconstitutes each previous version of the binary objects according to the technique illustrated in the flow chart depicted in FIG. 5h. Program control then continues with step 450 where the Distributed Storage Manager program 24, for each binary object, compares the "contents identifier" of the next "granule" in the work area of remote backup file server 12 against the corresponding "contents identifier" calculated in step 444. Program control continues with step 452 where the Distributed Storage Manager program 24 determines whether the

"contents identifiers" match. If so, program control is returned to step 450 since this "granule" is the same on the local computer 20 and on the remote backup file server 12. If the Distributed Storage Manager program 24 determines, in step 452, that the "contents identifiers" do not match, program control continues with step 454 where the Distributed Storage Manager program 24 transmits the "granule" to the local computer 20. Program control then continues with step 456 where the "granule" received by the local computer 20 is written directly to the current version of the binary object at the appropriate location. Program control then continues with step 458 where the Distributed Storage Manager program 24 determines whether there are any more "granules" to be examined for the binary object currently being processed. If so, program control is returned to step 450; otherwise the file restore routine is terminated at step 460. After all "granules" are received from the remote backup file server 12, the binary object has been restored to the state of the previous version.

## 5. Auditing and Reporting

The Distributed Storage Manager program 24 is able to perform self-audits on a periodic basis to ensure that the binary objects that have been backed up can be restored. To perform an audit, the Distributed Storage Manager program 24 executes the steps illustrated in the flow chart of FIG. 5j. Program control begins at step 500 where the Distributed Storage Manager program 24 initiates a restore of a randomly selected binary object identified by a Binary Object Identification Record 58 stored in File Database 25. Program control continues with step 502 where the selected binary object is restored from either a compressed storage file 32

residing on one of the disk drives 19 of one of the local computers 20 or from the remote backup file server 12. Program control then continues with step 504 where, as the binary object is being restored, a Binary Object Identifier 74 is calculated from the binary object instead of writing the binary object to one of the disk drives 19 of one of the local computers 20. Program control then continues with step 506 where the Distributed Storage Manager program 24 compares the Binary Object Identifier 74 calculated in step 504 to the original Binary Object Identifier 74 stored as part of the randomly selected Binary Object Identification Record 58. If the values are equal, program control continues with step 508 where the Distributed Storage Manager program 24 logs a successful audit restore. If the values are not equal, program control continues with step 510 where the Distributed Storage Manager program 24 generates an event indicating an audit failure.

## 6. Virtual Restore

The disk drives 19 associated with local computers 20 may have a very large storage capacity and may require a significant amount of time to be restored, especially if most or all of the data must be transmitted from the remote backup file server 12. To reduce the amount of time that a local computer 20 is "offline" during a full disk drive 19 restore, the Distributed Storage Manager program 24 employs a technique which allows a disk drive 19 associated with a local computer 20 to be only partially restored before being put back "online" for access by local computer 20. The user specifies to the Distributed Storage Manager program 24 that only those files that have been accessed in the last <n> days, <n> weeks or <n> months should be restored to the disk drive 19 before the disk

drive 19 is returned to the "online" state. Alternately, the user may specify that only files that are stored "locally" in compressed storage files 32 should be restored and that no files stored on the remote backup file server 12 should be restored before the disk drive 19 is returned to the "online" state. The overall result is a minimization of restore time in the event of disk drive 19 failure. This "virtual restore" technique generally works quite well since users who will begin accessing data on a particular disk drive 19 after it is put back "online" will most likely only be accessing data that had been "recently" accessed before failure of the disk drive 19.

The "virtual restore" process is illustrated in the flow chart depicted in FIG. 5k. Program control begins at step 600 where the Distributed Storage Manager program 24 obtains, from the user, the last access date that defines which files must be restored before the disk drive 19 can be returned to the "online" condition. Any files that were last accessed on or after this date will be restored before the disk drive 19 is placed "online". The specification of this date may be accomplished in any of the following ways: (1) actual date; (2) "within the last <n> days"; (3) "within the last <n> weeks"; or (4) "within the last <n> months". Alternately, the user may specify that only files that are currently backed up in compressed storage files 32 are to be restored as opposed to files stored on remote backup file server 12. Program control continues with step 602 where the Distributed Storage Manager Program 24 locates the most recent version of the File Database 25 for the disk drive 19 to be restored if the File Database 25 does not already exist on the disk drive 19. Program control then continues with step 604 where the next File Identification Record 34 in File Database 25 is read. Program control continues with step 606 where the Distributed Storage Manager

program 24 determines whether an additional File Identification Record 34 to be processed has been located in File Database 25. If not, program control continues with step 608 where the Distributed Storage Manager program 24 notifies the local computer 20 that the restored disk drive 19 may be placed "online" and terminates the virtual restore process. If another File Identification Record 34 has been located for processing, program control continues with step 610 where the Distributed Storage Manager program 24 locates the most recent Backup Instance Record 42 associated with the File Identification Record 34 currently being processed. In step 612, the Distributed Storage Manager program 24 determines whether the Last Access Date/Time field 52 in the Backup Instance Record 52 indicates that the file has been accessed since the user-specified last access date (step 600). If the file has been accessed on or since the user-specified last access date, program control continues with step 614 where the Distributed Storage Manager program 24 initiates the restoration of this file and sets the Migration Status field 41 in the File Identification Record 34 currently being processed to "NORMAL". Program control is then returned to step 604. If the Distributed Storage Manager program 24 determines, in step 612, that the file has not been accessed on or since the user-specified last access date, program control continues with step 616 where the Distributed Storage Manager program 24 sets the Migration Status field 41 in the File Identification Record 34 currently being processed to "MIGRATED". In this case, the file does not need to be restored. Program control is then returned to step 604.

Another feature of the virtual restore process is the ability to utilize the Migration Status field 41 in File Identification Record 34 for the performance of space



management. If a particular file has not been accessed on or since a user-specified last access date, the file can be backed up to the remote backup file server 12 and then deleted from the disk drives 19 associated with local computers 20. The Migration Status field 41 is then set to "MIGRATED". If a migrated file is later needed by a user, the file can be restored from the remote backup file server 12.

#### 7. Backup File Retention

The Distributed Storage Manager program 24 implements a backup file retention scheme wherein a retention pattern is maintained for each individual file that indicates which backup versions of a file are to be saved. A retention pattern for a file is defined as:

keep the last "d" daily backup copies of the file AND  
keep the last "w" weekly backup copies of the file AND  
keep the last "m" monthly backup copies of the file AND  
keep the last "q" quarterly backup copies of the file AND  
keep the last "y" yearly backup copies of the file.

By specifying the retention pattern in this way, all backup copies of a file that are needed to represent the backup of the file as it existed at the time it was backed up for the last "d" days, the last "w" weeks, the last "m" months, the last "q" quarters and the last "y" years are saved. However, by way of example, this may mean that only one backup copy of the file is saved to represent all "d" days (in the case where the file has not changed in the last "d" days), or this may mean that the last "d" daily backup copies of the file must be saved to represent the file as it existed for the last "d" daily backup cycles. The same principle is utilized for weekly, monthly, quarterly and yearly copies.

The backup file retention scheme utilized by the Distributed Storage Manager program 24 provides several unique benefits. First, this technique prevents undetected virus or application program damage to a file from destroying all good backup copies of a file. If a file is damaged and this condition is not noticed for several days, then a scheme which only maintains the last "n" versions of a file may result in the situation where an "undamaged" backup copy of the file is not available. The backup file retention scheme of the present invention allows backup copies of files to be kept that represent the file as it existed at various times during the past several days, weeks, months or even years. Second, the file retention scheme utilized by the Distributed Storage Manager program 24 eliminates the need for most archives. Most archives are designed to take a snapshot of a group of files as of a certain date, such as at the end of each month. The Distributed Storage Manager program's use of retention patterns eliminates the need for users to take periodic snapshots of their data using a special archive, since the Distributed Storage Manager program 24 handles this automatically.

In order for the Distributed Storage Manager program 24 to implement the backup file retention scheme, each file stored on the local computers 20 must be associated with a specific retention pattern. The Management Class field 43 in the File Identification Record 34 of File Database 25 specifies a management class for each file. In turn, each management class is associated with a specific file retention pattern. In this way, a specific retention pattern is associated with each file. Those of ordinary skill in the art will recognize that other methods of assigning a specific file retention pattern to a file may also be utilized.

The operation of the backup file retention scheme utilized by the Distributed Storage Manager program 24 is illustrated in the flow chart of FIG. 51. Program control begins at step 700 where the Distributed Storage Manager program 24 locates each File Identification Record 34 in the File Database 25. Program control continues at step 702 where the Distributed Storage Manager program 24 determines the required file retention pattern by examining the Management Class field 43 in the File Identification Record 34 currently being processed and then creates a "retention working list". The "retention working list" is a list of entries that specify the starting and ending dates for each backup copy that should be retained based upon the specified retention pattern. For example, if the user has specified that the last "d" daily backup copies must be retained, then the "retention working list" will contain "d" entries with the "start date" equal to the "end date" for each entry and the dates for the first entries set equal to the current date, the dates for the second entries set equal to the previous day's date, etc. For weekly entries, the "retention working list" will contain entries (one per weekly backup copy to be retained) with the "start date" set to the date that specifies the beginning of the prior week (based on the current date) and the "end date" set to the date that specifies the end of the prior week (based on the current date). If "w" weeks are to be retained, then "w" weekly "retention working list" entries will be created. At the end of this process, the "retention working list" will contain a list of "windows" which indicates the date ranges that a file must fall within in order to be retained.

Program control continues with step 704 where the Distributed Storage Manager program 24 locates the most recent Backup Instance Record 42 associated with the File

Identification Record 34 currently being processed. Program control continues with step 706 where the Distributed Storage Manager program 24 compares the date stored in the Insert Date field 57 of the Backup Instance Record 42 currently being  
5 processed with any "unused" date ranges set forth in the "retention working list" (if any of the "retention working list" entries have already been satisfied, they will be marked as "used" as is discussed more fully below in relation to step 708). If the date stored in the Insert Date field 57 does not  
10 fall within any of the "unused" "retention working list" date ranges, then program control continues with step 712 where the Backup Instance Record 42 is deleted. Otherwise, program control continues with step 708 where all "retention working list" entries satisfied by the date stored in the Insert Date  
15 Field 57 are marked as "used" to indicate that a Backup Instance Record 42 has been used to satisfy this entry. This ensures that an older Backup Instance Record 42 is not used to satisfy a retention pattern specification when a newer entry also satisfies the condition. The Distributed Storage Manager  
20 program 24 also checks to ensure that the file associated with the Backup Instance Record 42 has not been deleted prior to the "end date" of the window satisfied by the date stored in Insert Date field 57. This condition is satisfied by ensuring that the date stored in the Delete Date field 56 of the Backup  
25 Instance Record 42 currently being processed is after the "end date" of the window satisfied by the date stored in Insert Date field 57. If the file was deleted prior to the "end date" of the window, then the file cannot be used to satisfy the "retention working list" entry since that file did not exist on  
30 the "end date". Following either step 708 or step 712, program control continues with step 710 where the Distributed Storage Manager program 24 determines whether there are any additional

Backup Instance Records 42 associated with the File  
Identification Record 34 currently being processed. If so,  
program control is returned to step 704; otherwise, program  
control is returned to step 700.

5

While the present invention has been described in  
connection with an exemplary embodiment thereof, it will be  
understood that many modifications and variations will be  
readily apparent to those of ordinary skill in the art. This  
disclosure and the following claims are intended to cover all  
such modifications and variations.

10

We claim:

Sub A 2

1. A system for distributed management of the storage space and data on a networked computer system wherein the networked computer system includes at least two storage devices for storing data files comprised of one or more binary objects, said distributed storage management system comprising:

means for selectively copying the binary objects stored on one of the storage devices to another of the storage devices;

means for calculating a current value for a binary object identifier for selected binary objects stored on the storage devices wherein said calculation of said binary object identifier is based upon the actual data contents of the associated binary object;

means for storing said current value of said binary object identifier as a previous value of said binary object identifier;

means for comparing said current value of said binary object identifier associated with a particular binary object to one or more previous values of said binary object identifier associated with that particular binary object; and

means for commanding said means for selectively copying binary objects in response to said means for comparing.

2. The distributed storage management system of claim 1 wherein said means for calculating said current value for said binary object identifier includes means for calculating a current value for a binary object identifier comprised of at least two independently determined values.

3. The distributed storage management system of claim 1 wherein said means for calculating said current value

for said binary object identifier includes means for calculating a 128-bit binary value comprised of four 32-bit fields.

4. The distributed storage management system of claim 3 wherein said four 32-bit fields include a binary object identifier size field, a Cyclical Redundancy Check number field calculated against the contents of the binary object associated with said current value of said binary object identifier, a Longitudinal Redundancy Check number field calculated against the contents of the binary object associated with said current value of said binary object identifier, and a binary object hash number field calculated against the contents of the binary object associated with said current value of said binary object identifier.

5. The distributed storage management system of claim 1 wherein the networked computer system further includes at least two local computers each of which is in communication with at least one of the storage devices and wherein said distributed storage management system is cooperatively executable on at least two of the local computers.

6. The distributed storage management system of claim 1 wherein the networked computer system further includes a remote backup file server and wherein said means for selectively copying binary objects selectively copies binary objects stored on one of the storage devices to another of the storage devices and to the remote backup file server unless sufficient storage space is not available on another of the storage devices in which case the binary objects are copied only to the remote backup file server.

7. The distributed storage management system of claim 1 further including means for auditing the performance of said distributed storage management system, said means for auditing including:

second means for <sup>controlling</sup> ~~commanding~~ said means for selectively copying binary objects to recopy a previously copied binary object;

means for recalculating said binary object identifier for the recopied binary object;

means for comparing said recalculated binary object identifier to a previous value of said binary object identifier for the recopied binary object; and

means for reporting a failure if said recalculated binary object identifier is not identical to said previous value of said binary object identifier.

8. The distributed storage management system of claim 1 wherein said means for <sup>controlling</sup> ~~commanding~~ said means for selectively copying includes means for <sup>controlling</sup> ~~commanding~~ said means for selectively copying to copy a particular binary object only if said current value of said binary object identifier for that particular binary object is not identical to a previous value of said binary object identifier for that particular binary object in response to said means for comparing.

9. A system for distributed management of the storage space and data on a networked computer system wherein the networked computer system includes at least two storage devices for storing data files comprised of one or more binary objects, said distributed storage management system comprising:  
means for segmenting the binary objects into granules

Sub  
a3



Sub  
03  
means for selectively copying said granules stored on one of the storage devices to another of the storage devices;

means for calculating a current value for a granule identifier for selected granules stored on the storage devices wherein said calculation of said granule identifier is based upon the actual data contents of said associated granule;

means for storing said current value of said granule identifier as a previous value of said granule identifier;

means for comparing said current value of said granule identifier associated with a particular granule to one or more previous values of said granule identifier associated with said particular granule; and

means for commanding said means for selectively copying granules in response to said means for comparing.

10. The distributed storage management system of claim 9 further including means for reconstructing a binary object from a most recent complete copy of the binary object, said means for reconstructing including:

means for copying said granules copied by said means for selectively copying granules to said most recent complete copy of the binary object in order from most-recently copied granule to least-recently copied granule; and

means for generating a bitmap for controlling said means for copying said copied granules.

11. The distributed storage management system of claim 10 wherein said means for calculating said current value for said granule identifier includes means for calculating a 32-bit Cyclical Redundancy Check number calculated against the contents of said granule associated with said present value of said granule identifier and means for calculating a 32-bit

binary object hash number calculated against the contents of said granule associated with said present value of said granule identifier.

12. The distributed storage management system of claim 10 further including means for restoring a current version of a binary object to a previous version of that binary object, said means for restoring including:

means for calculating said granule identifier for each of said granules in the current version of the binary object;

means for comparing said calculated granule identifier to a previous value of said granule identifier for each of said granules in the current version of the binary object; and

means, responsive to said means for comparing said granule identifiers, for replacing those granules in the current version of the binary object for which said granule identifiers are not identical.

13. A system for distributed management of the storage space and data on a networked computer system wherein the networked computer system includes at least two storage devices for storing data files comprised of one or more binary objects and at least two local computers each of which is in communication with at least one of the storage devices, said distributed storage management system comprising:

means for selectively copying binary objects stored on one of the storage devices to another of the storage devices;

means for indicating which of said copied binary objects must be copied to a particular storage device in the

event of a failure of that storage device before that storage device is considered to be operable by the local computer with which that storage device is in communication; and

means for <sup>controlling</sup> ~~commanding~~ said means for selectively copying binary objects in response to said means for indicating.

14. The distributed storage management system of claim 13 wherein said means for indicating includes means for specifying a last access date such that only binary objects that have been accessed by the networked computer system on or after said last access date must be copied to a particular storage device before that storage device is considered to be operable.

15. A system for distributed management of the storage space and data on a networked computer system wherein the networked computer system includes at least two storage devices for storing data files comprised of one or more binary objects, said distributed storage management system comprising:

means for selectively copying the binary objects stored on one of the storage devices to another of the storage devices;

means for maintaining a file retention list wherein said file retention list includes a file retention pattern for each binary object copied by said means for selectively copying binary objects;

means for determining which of the binary objects copied by said means for selectively copying binary objects match each of said file retention patterns; and

means for deleting the binary objects from the storage devices in response to said means for determining.

16. The distributed storage management system of claim 15 wherein said file retention pattern includes daily, weekly, monthly, quarterly and yearly retention patterns.

17. A system for distributed management of the storage space and data on a networked computer system wherein the networked computer system includes a remote backup file server and at least two storage devices for storing data files comprised of one or more binary objects, said distributed storage management system comprising:

means for selectively copying the binary objects stored on one of the storage devices to another of the storage devices or to the remote backup file server; and

means for employing user-defined priorities to determine which binary objects are to be copied to another storage device and to determine a queuing sequence for copying binary objects to the remote backup file server.

Sub  
a4

18. A method for management of the storage space and data on a computer system wherein the computer system includes at least two storage areas for storing data files comprised of one or more binary objects, said method comprising the steps of:

selectively copying the binary objects stored in one of the storage areas to another of the storage areas;

calculating a current value for a binary object identifier for selected binary objects stored in the storage areas wherein said calculation of said binary object identifier is based upon the actual data contents of the associated binary object;

storing said current value of said binary object identifier as a previous value of said binary object identifier;

comparing said current value of said binary object identifier associated with a particular binary object to one or more previous values of said binary object identifier associated with that particular binary object; and

controlling said step for selectively copying binary objects in response to said step for comparing.

19. The method of claim 18 wherein said step for calculating said current value for said binary object identifier includes the step for calculating a current value for a binary object identifier comprised of at least two independently determined values.

20. The method of claim 18 wherein said step for calculating said current value for said binary object identifier includes the step for calculating a current value for a binary object identifier utilizing a 128-bit binary value comprised of four 32-bit fields and wherein said four 32-bit fields include a binary object identifier size field, a Cyclical Redundancy Check number field calculated against the contents of the binary object associated with said binary object identifier, a Longitudinal Redundancy Check number field calculated against the contents of the binary object associated with said binary object identifier, and a binary object hash number field calculated against the contents of the binary object associated with said binary object identifier.

ADD D.

— — — — —

20

395/60

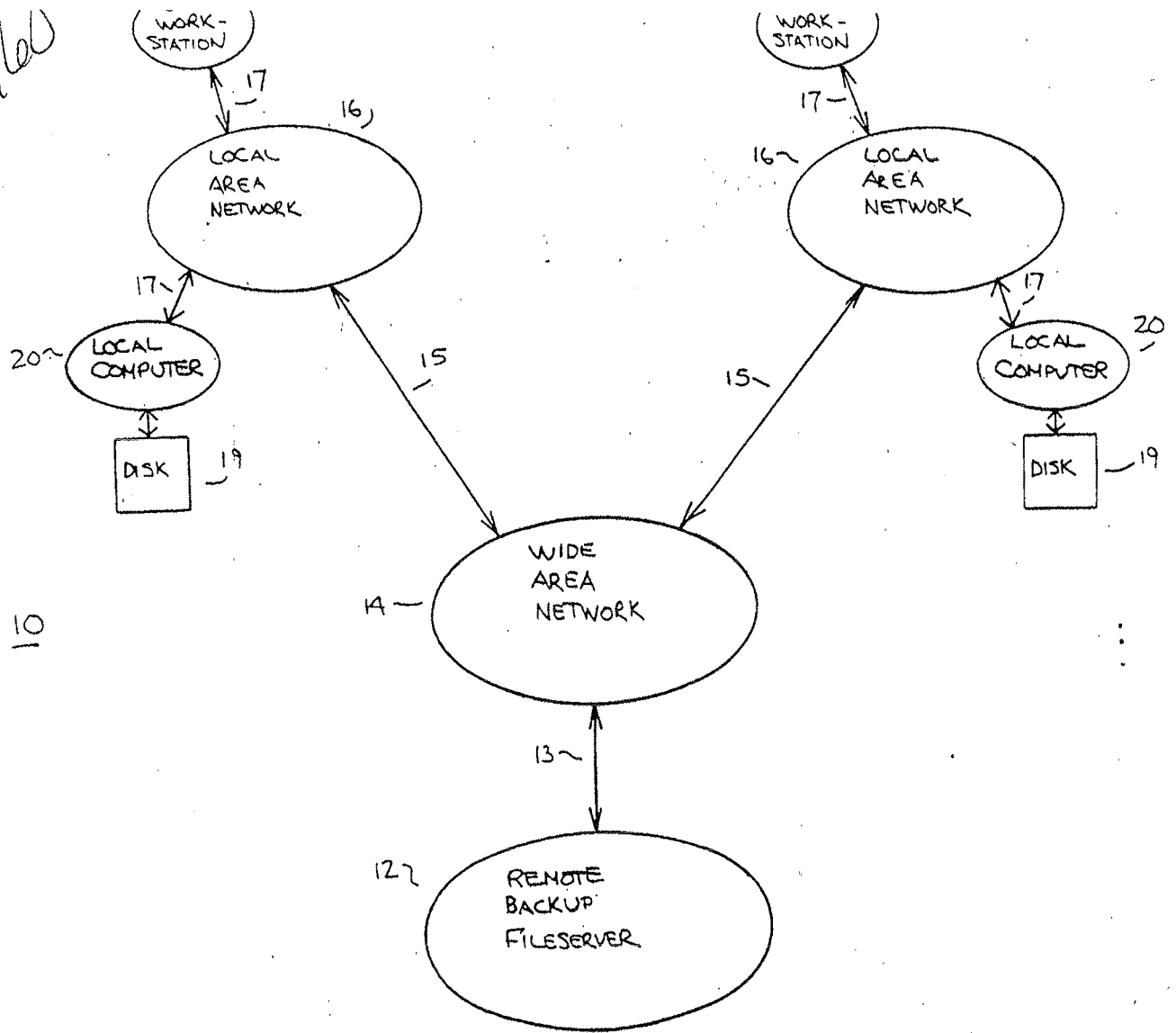


FIG. 1

19

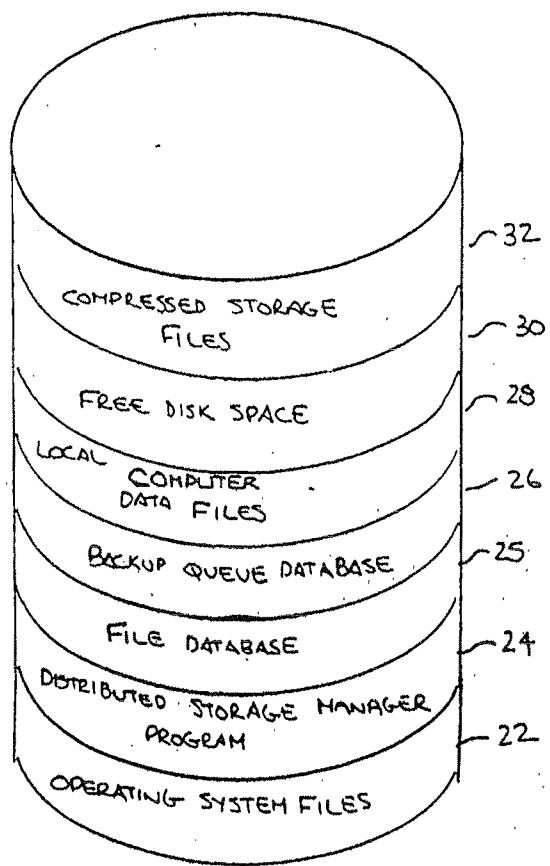
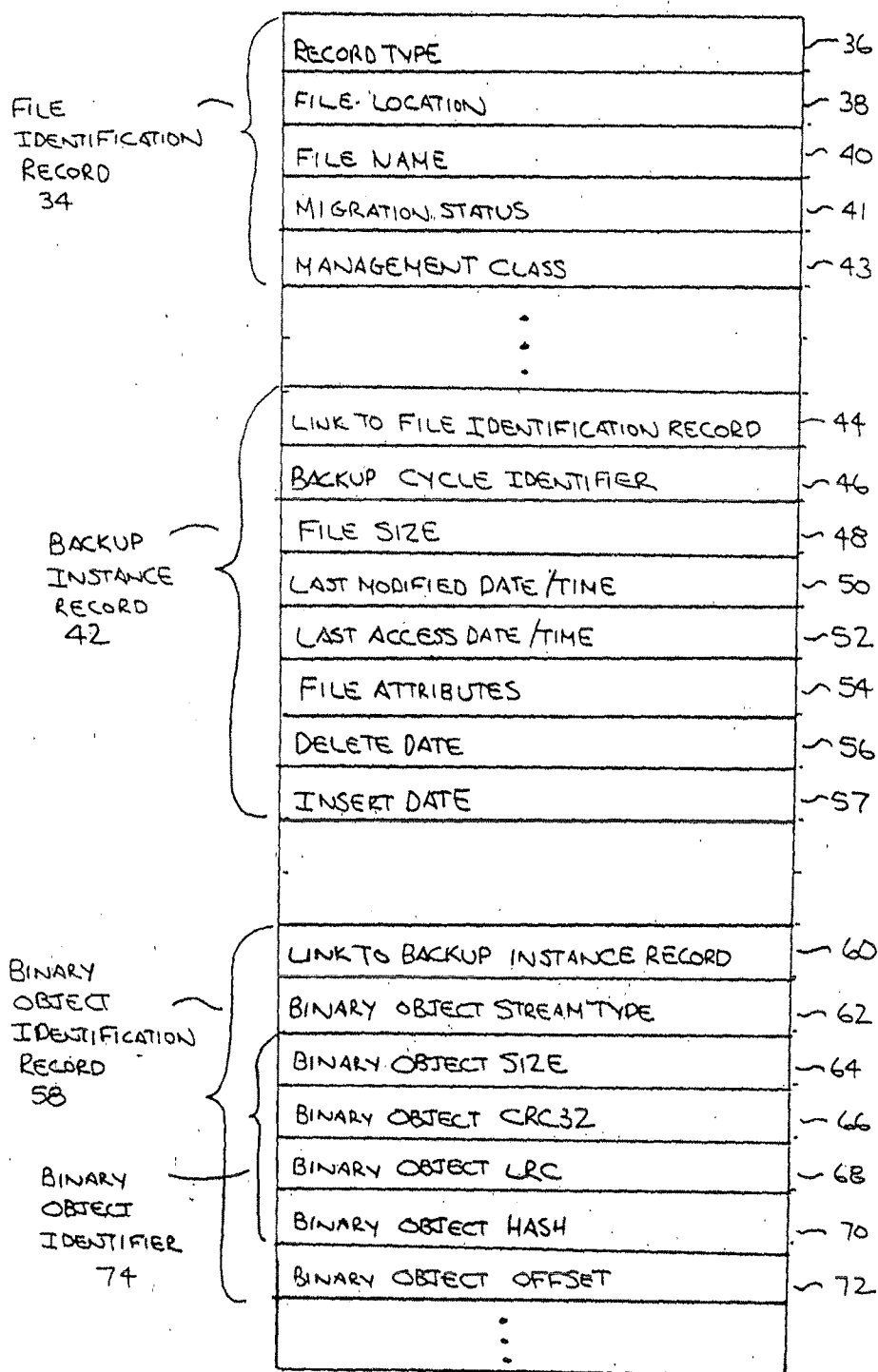


FIG. 2





25

FIG. 3

BACKUP  
QUEUE  
RECORD  
75

{	RECORD TYPE	~ 76
	FILE LOCATION	~ 78
	FILE NAME	~ 80
	FILE STATUS	~ 82
	FILE SIZE	~ 84
	LAST MODIFIED DATE/TIME	~ 86
	LAST ACCESS DATE/TIME	~ 88
	FILE ATTRIBUTES	~ 90
	FILE PRIORITY	~ 92
	...	...

FIG. 4

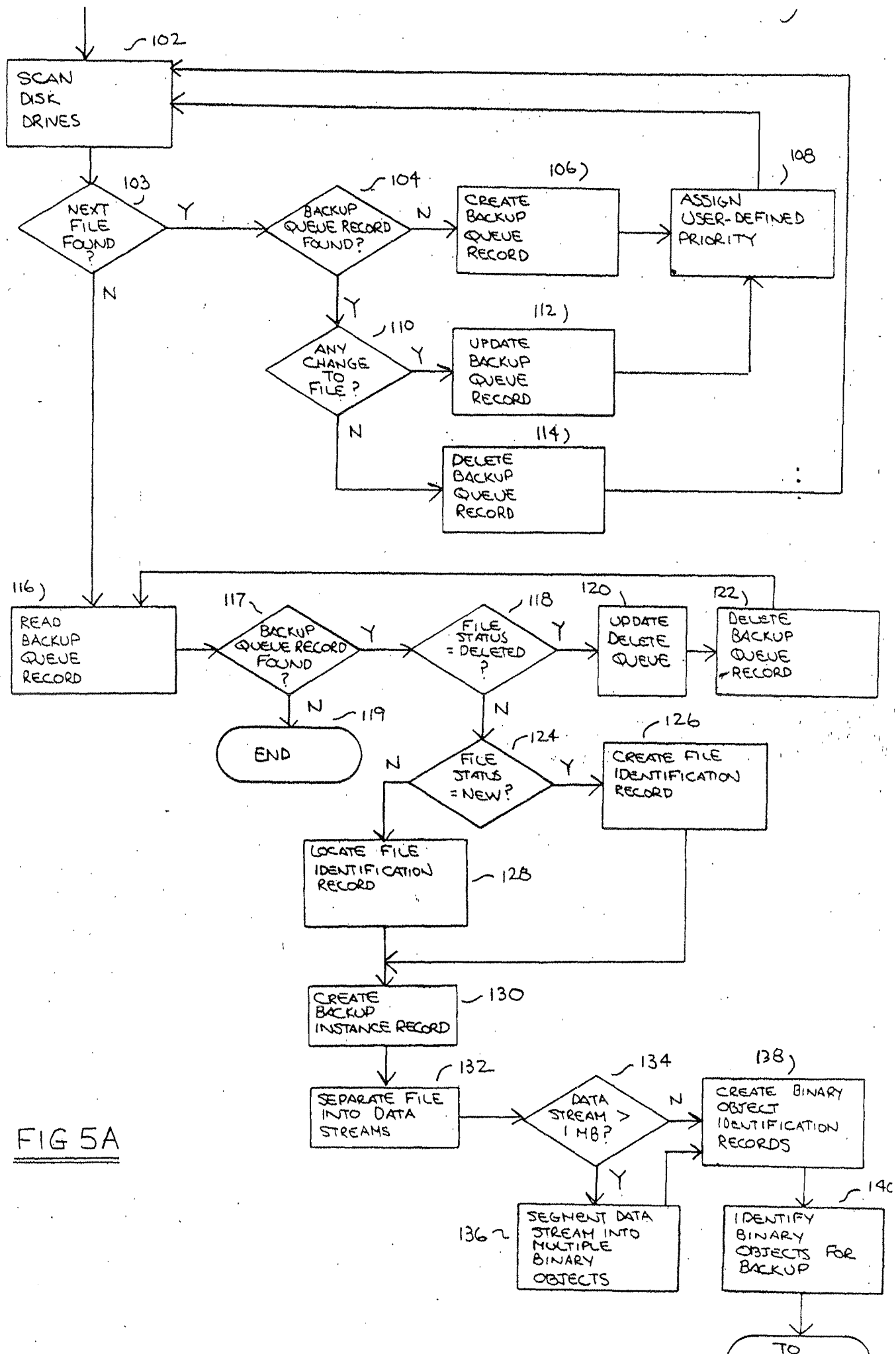


FIG 5A

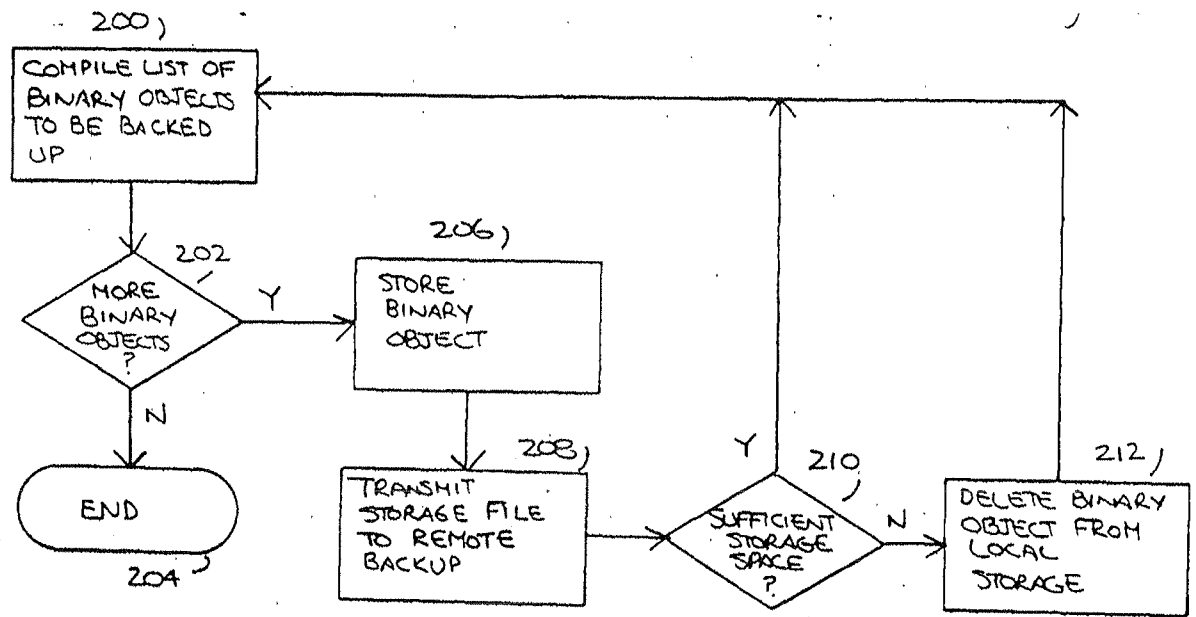


FIG. 5B

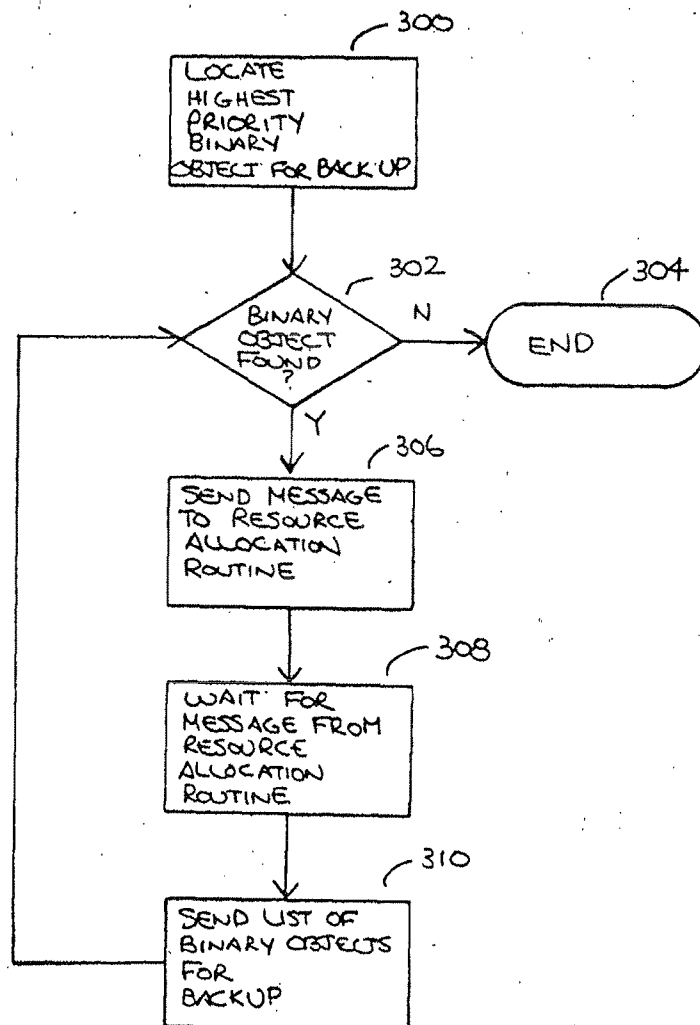


FIG. 5C

08-555,376

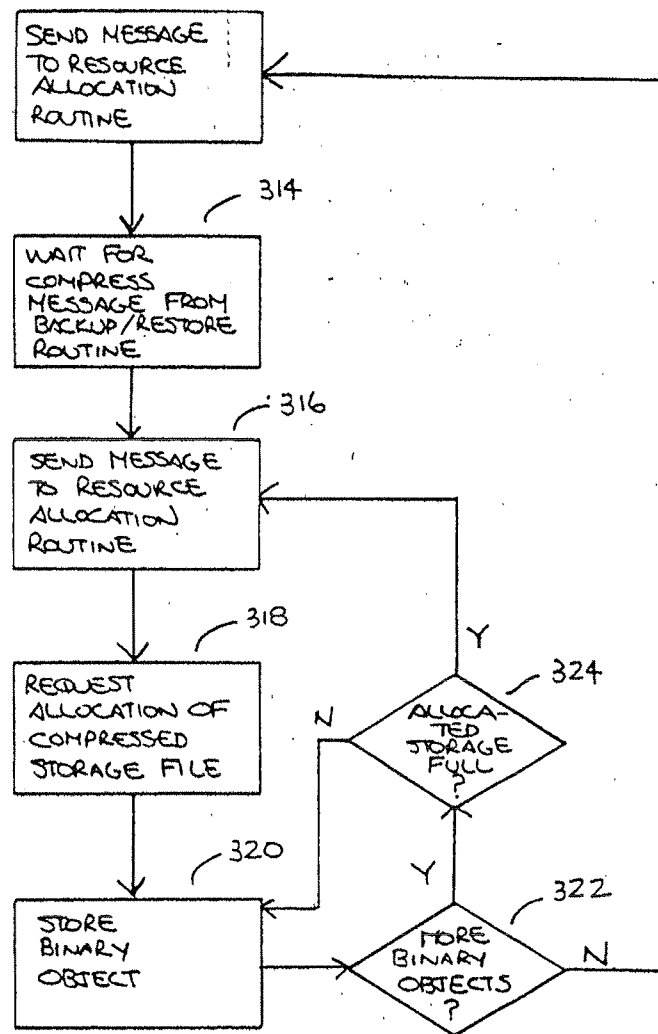


FIG. 5D

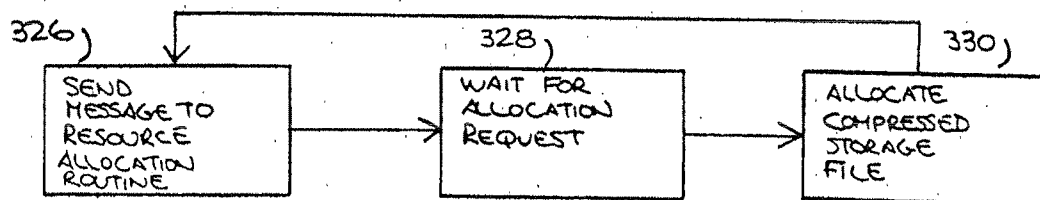


FIG. 5E

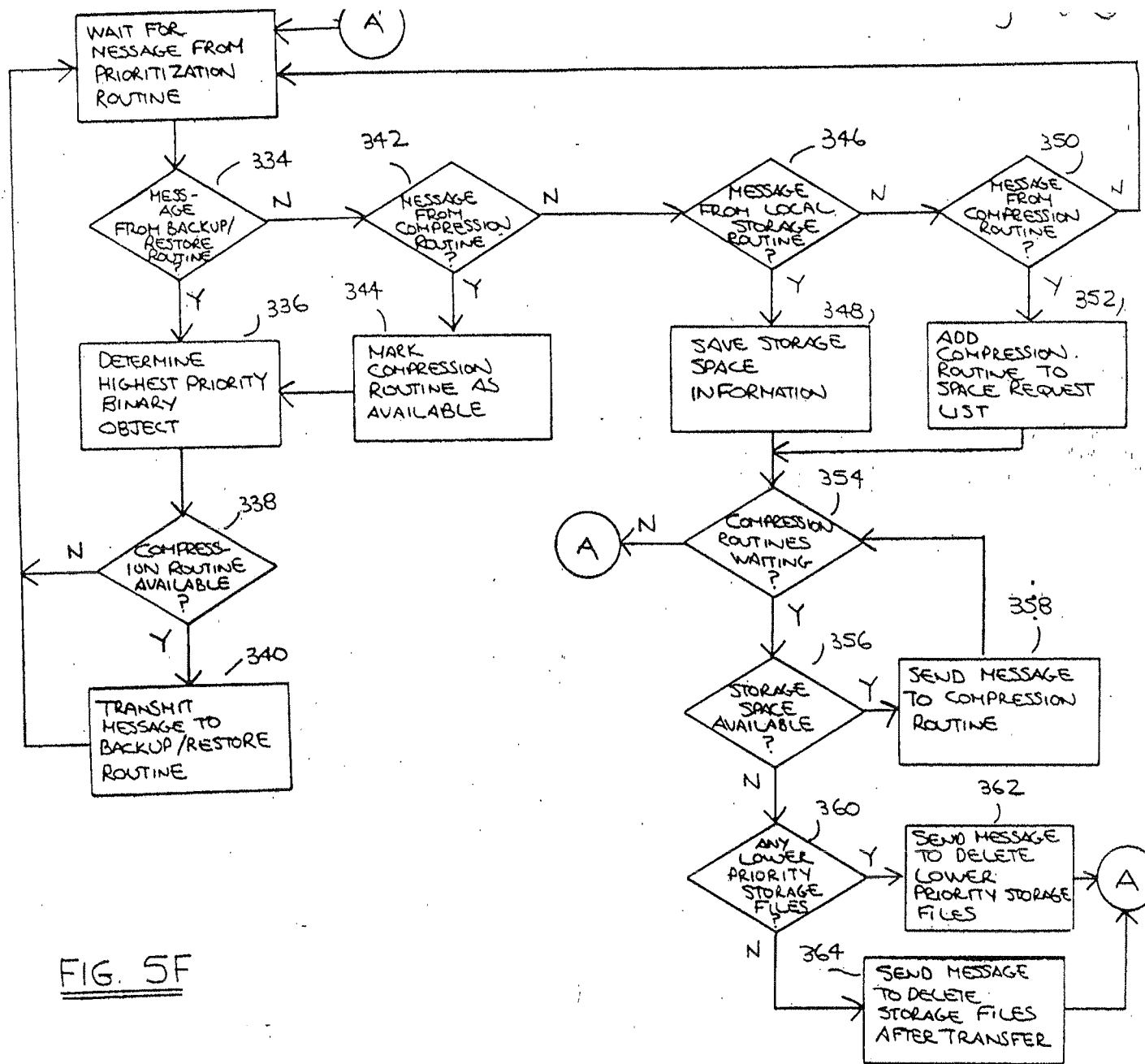


FIG. 5F

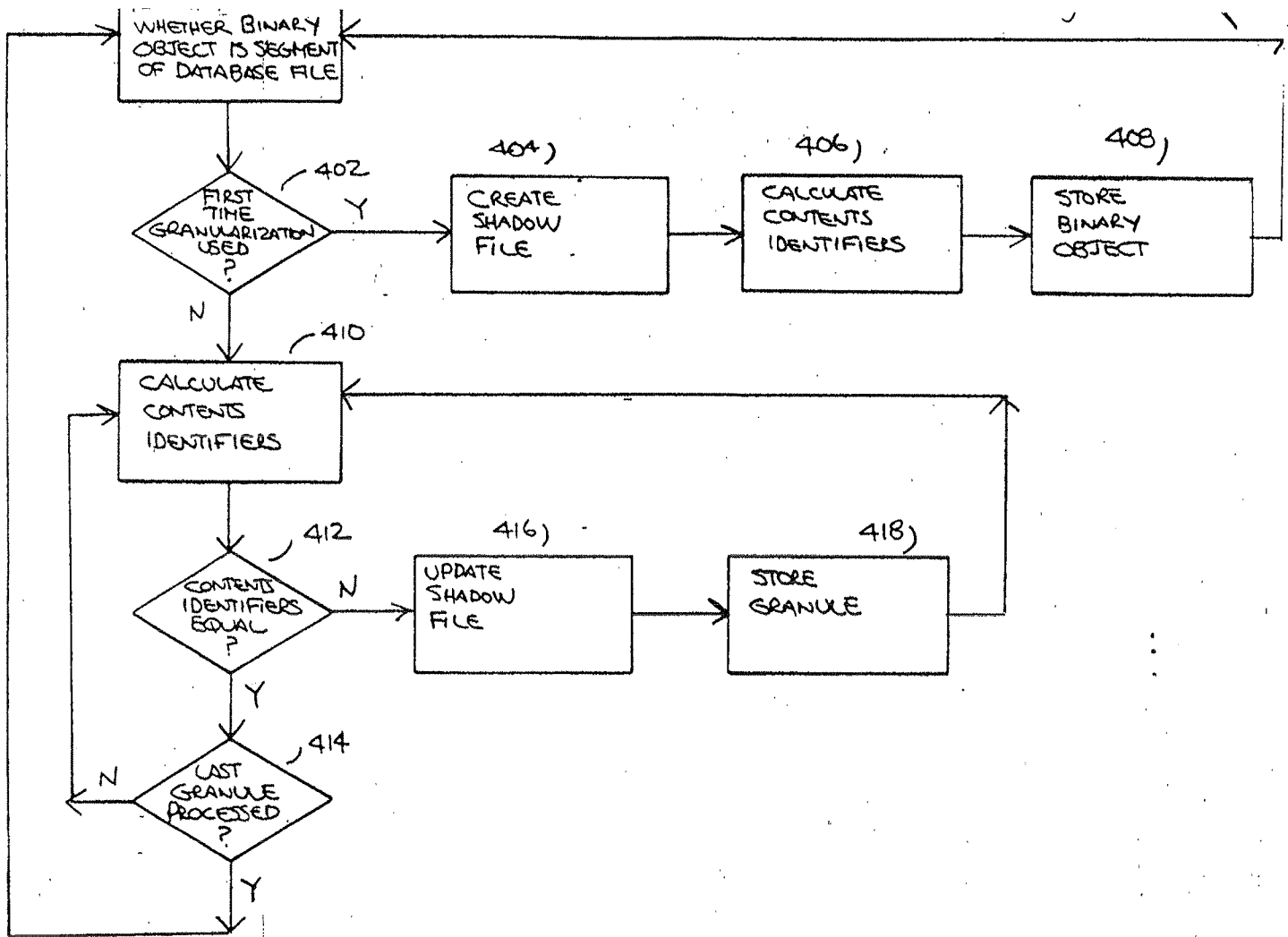


FIG. 5G

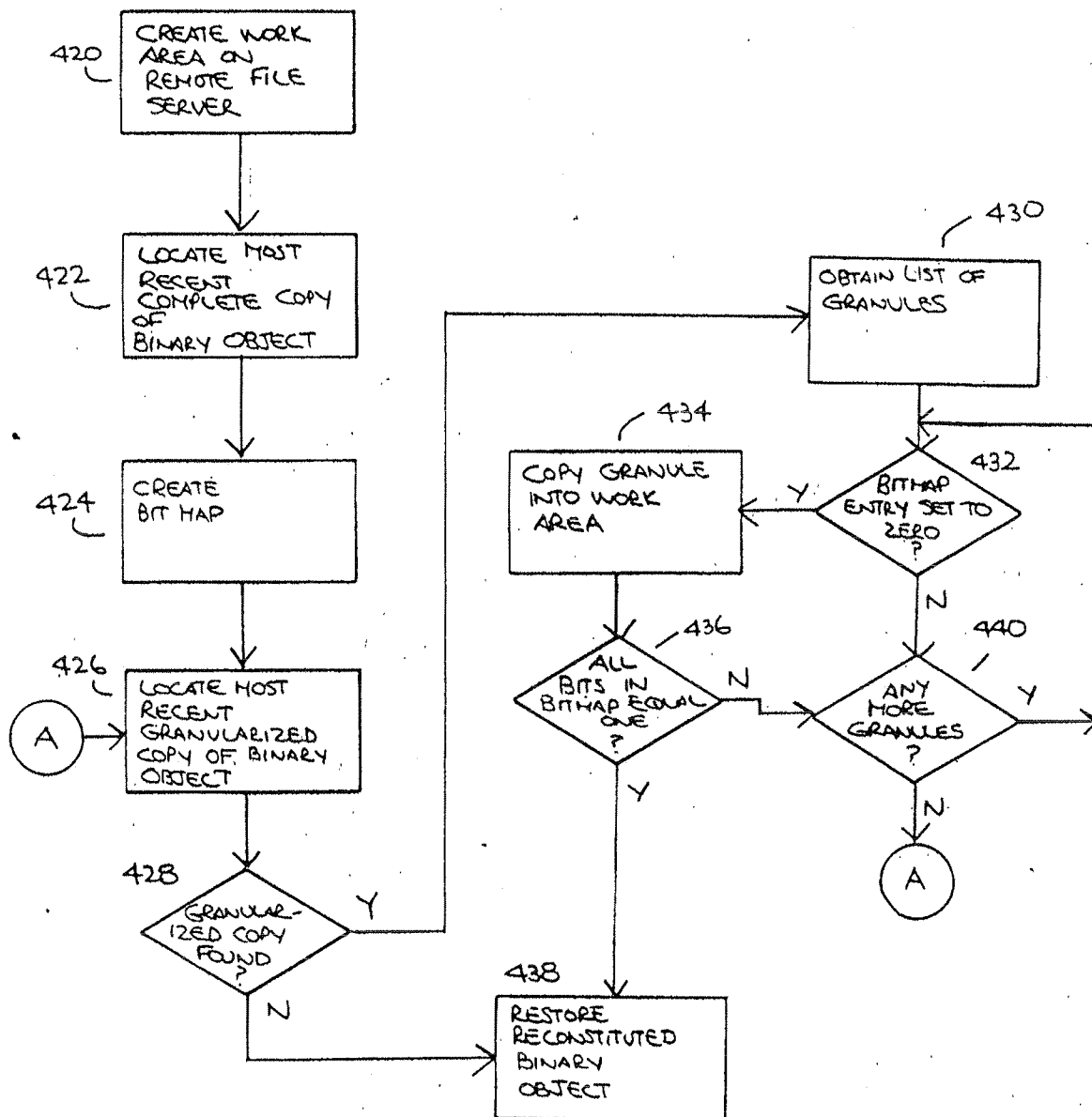
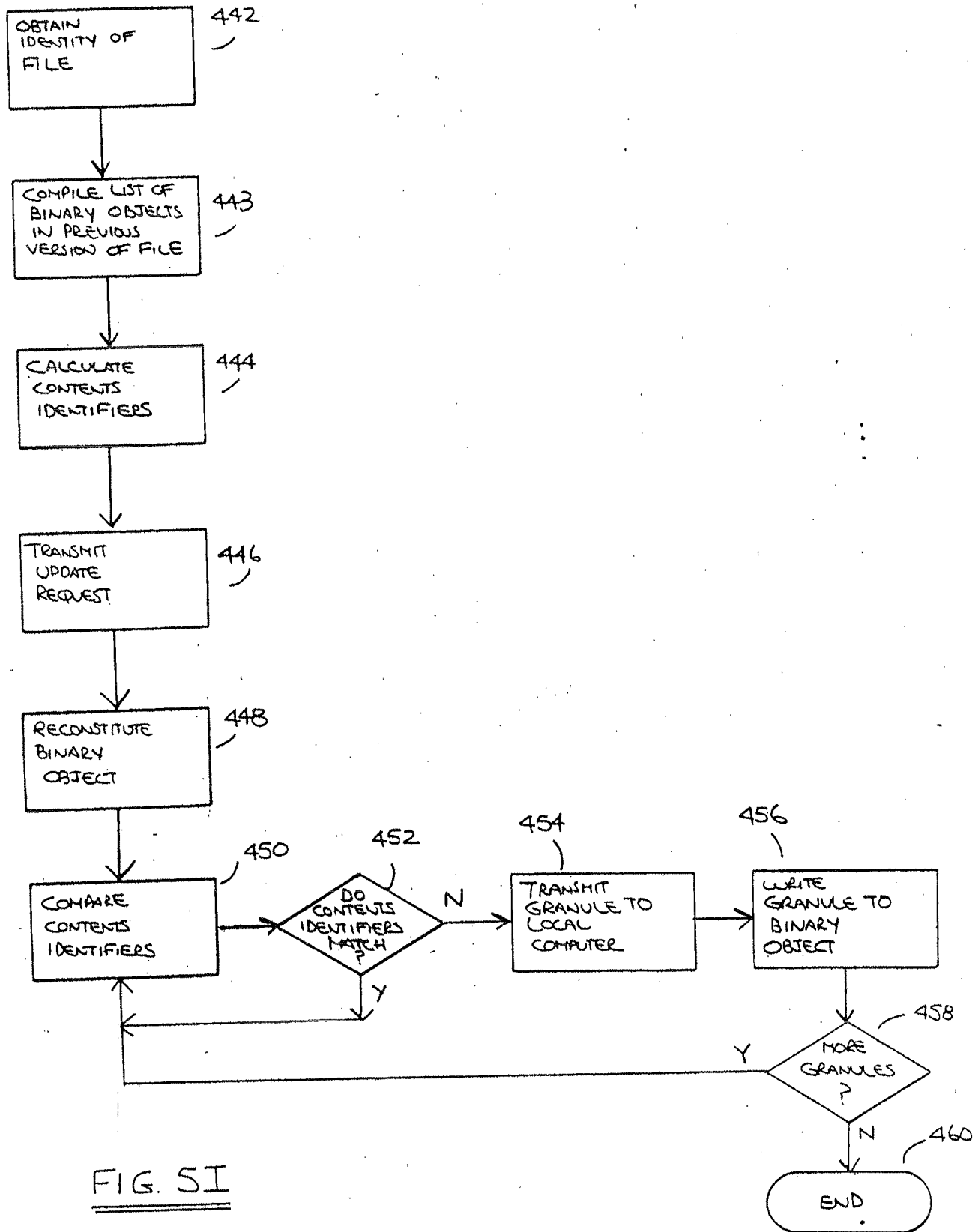


FIG. 5H





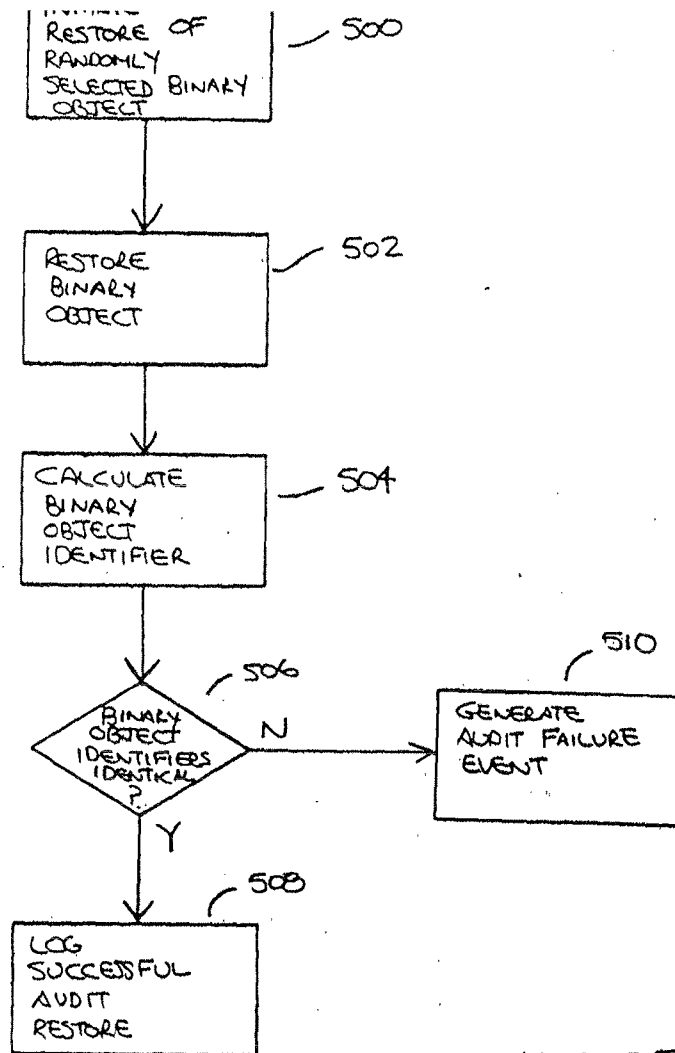


FIG. 5J

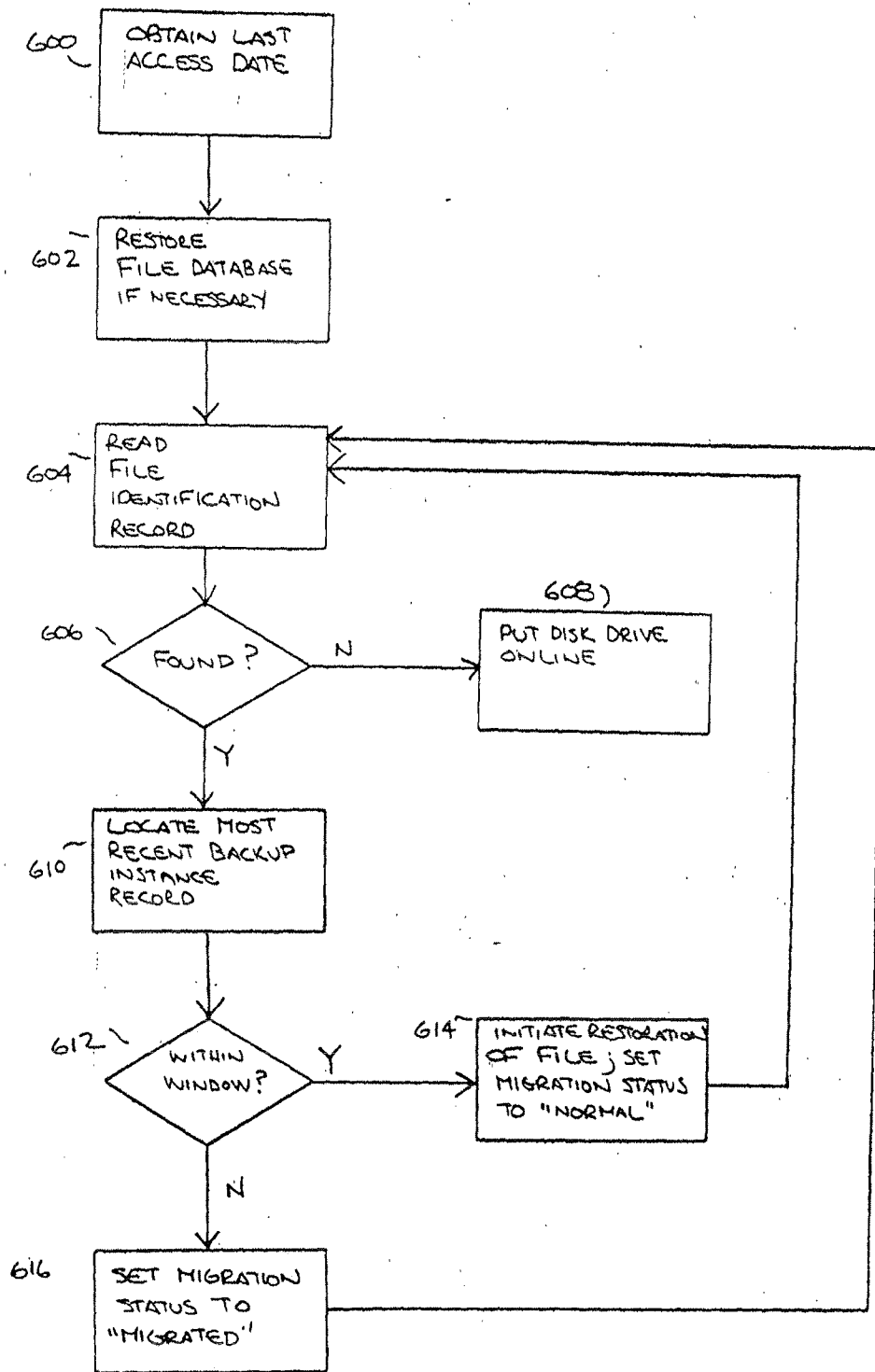


FIG. 5K

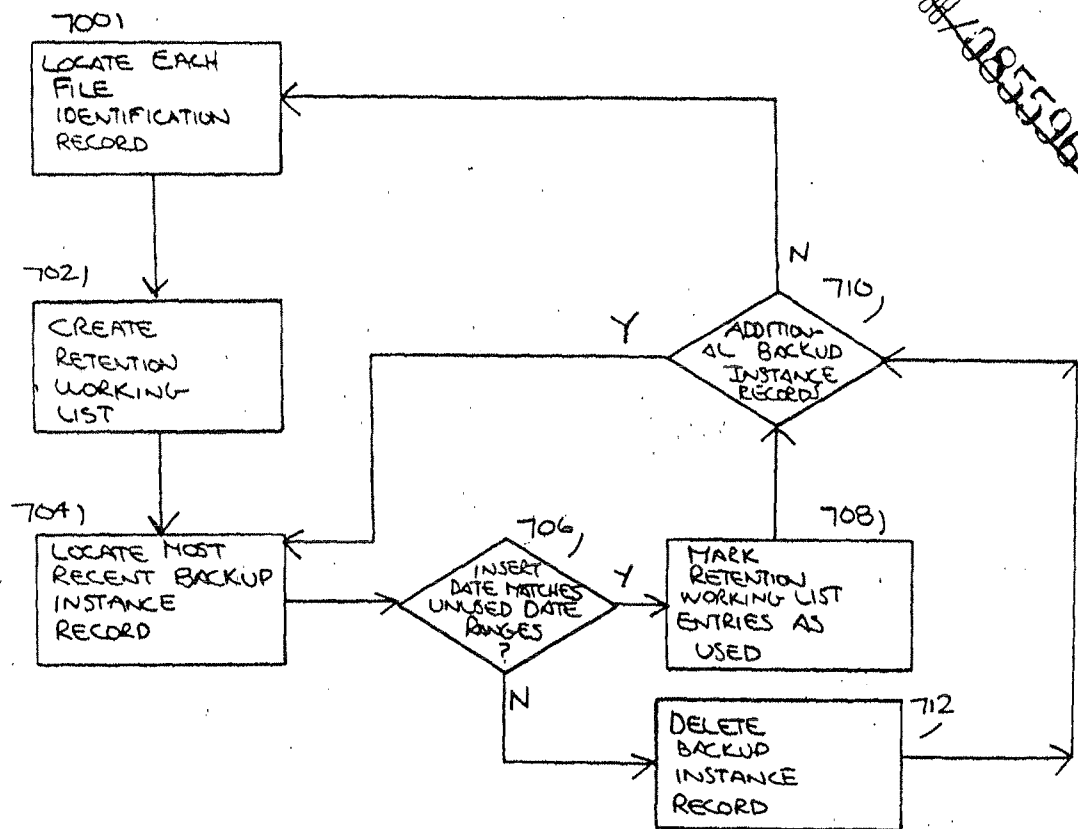


FIG. 5L

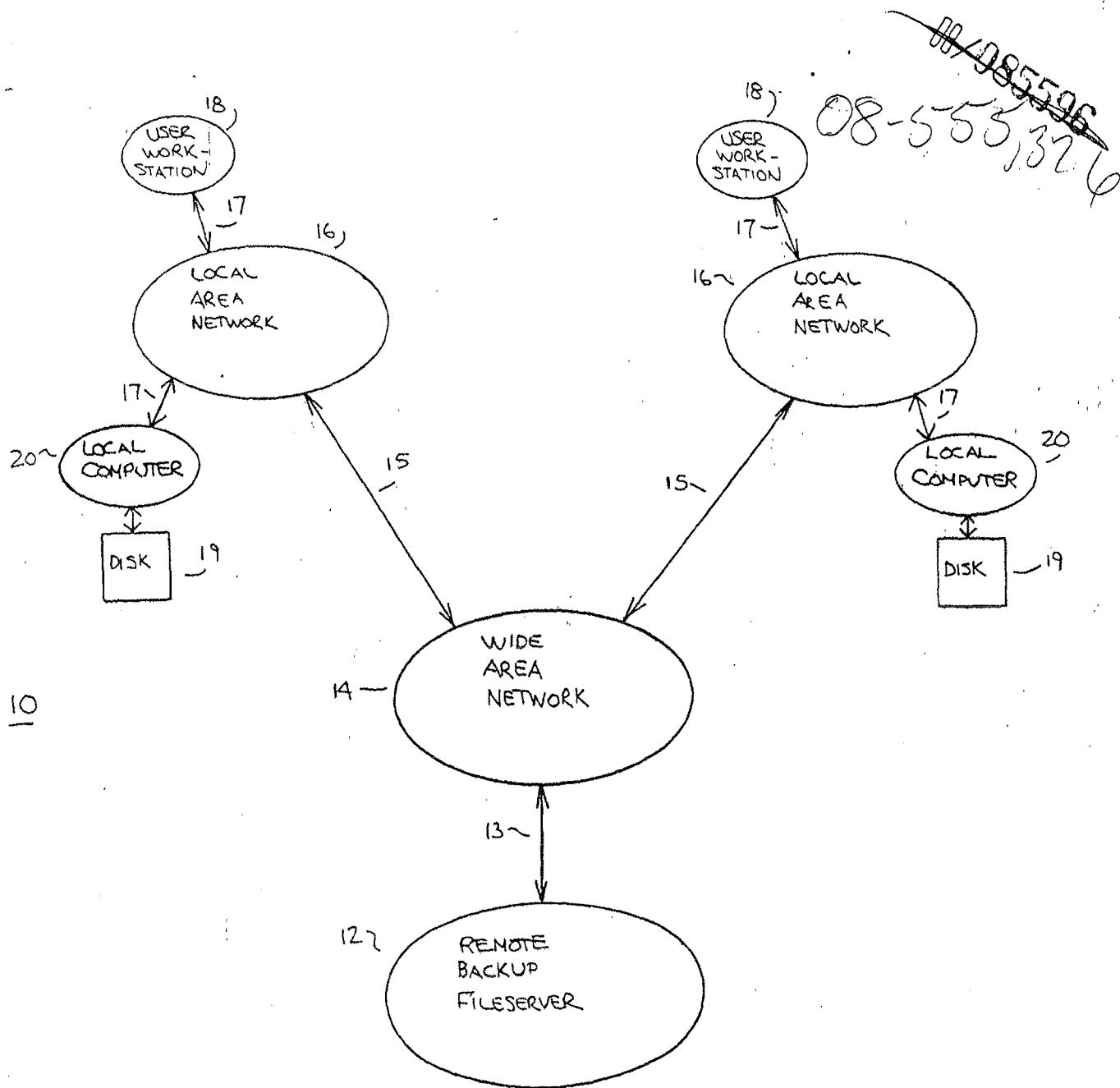


FIG. 1

19

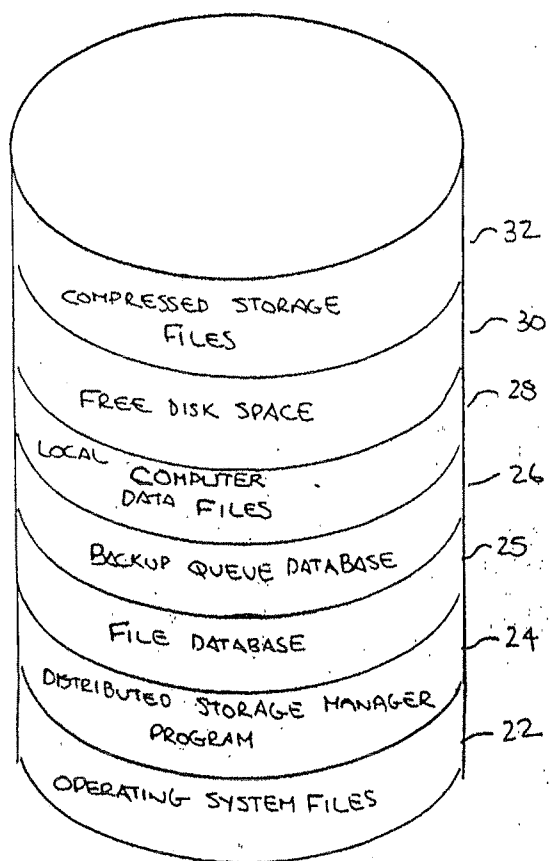
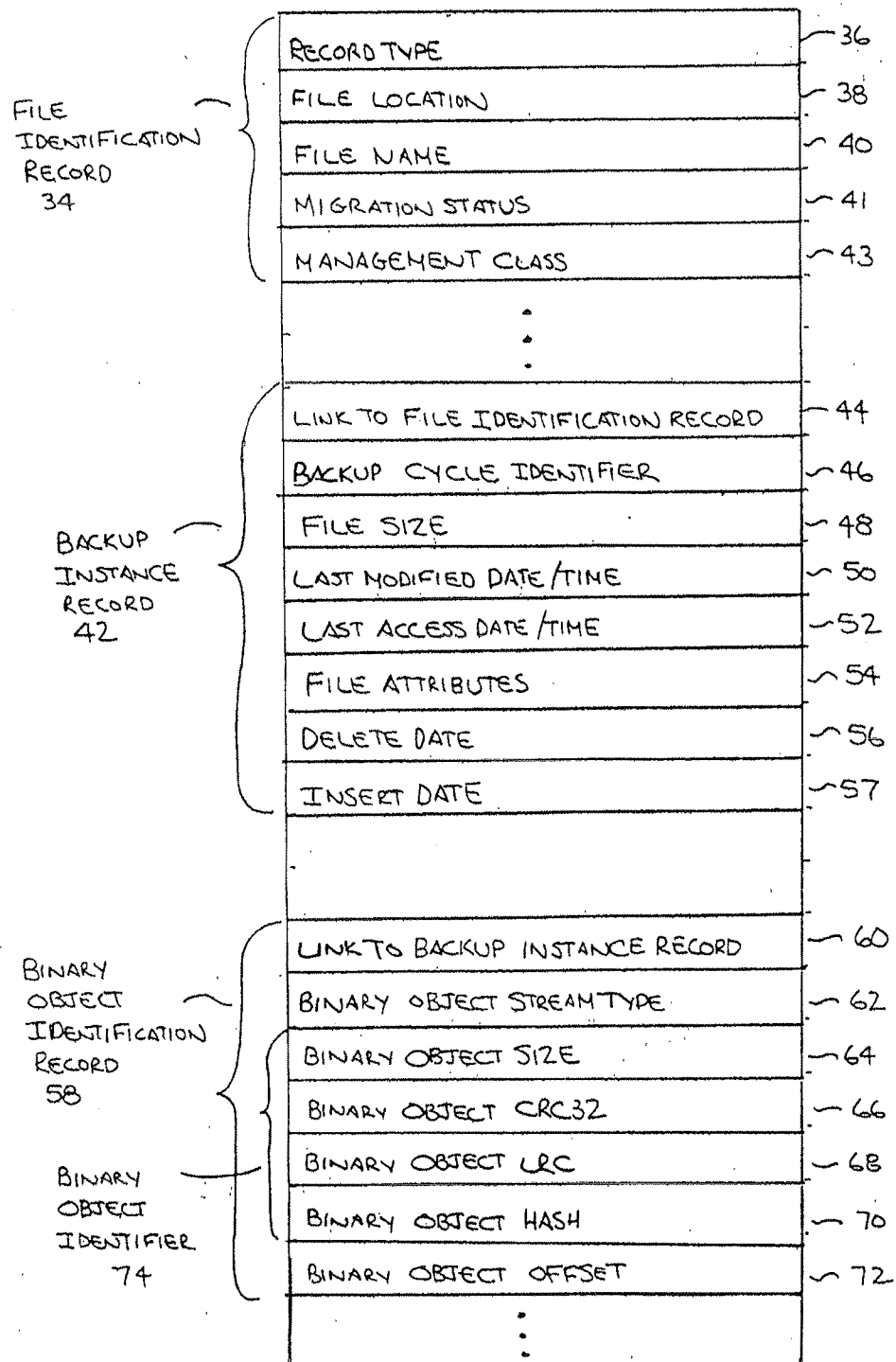


FIG. 2

08-558,346



25

FIG. 3

26

BACKUP  
QUEUE  
RECORD  
75

RECORD TYPE	76
FILE LOCATION	78
FILE NAME	80
FILE STATUS	82
FILE SIZE	84
LAST MODIFIED DATE/TIME	86
LAST ACCESS DATE/TIME	88
FILE ATTRIBUTES	90
FILE PRIORITY	92
...	

FIG. 4



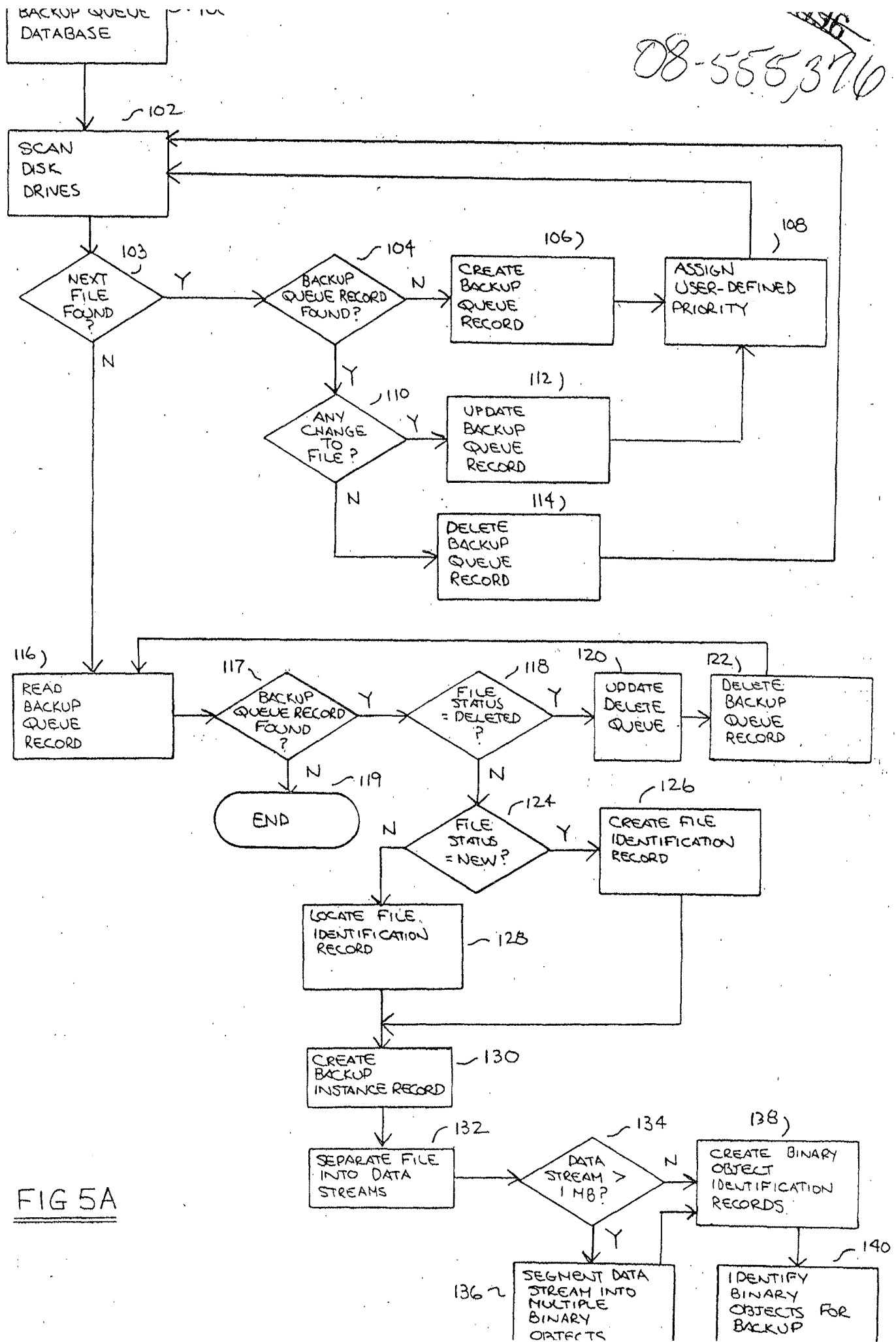


FIG 5A

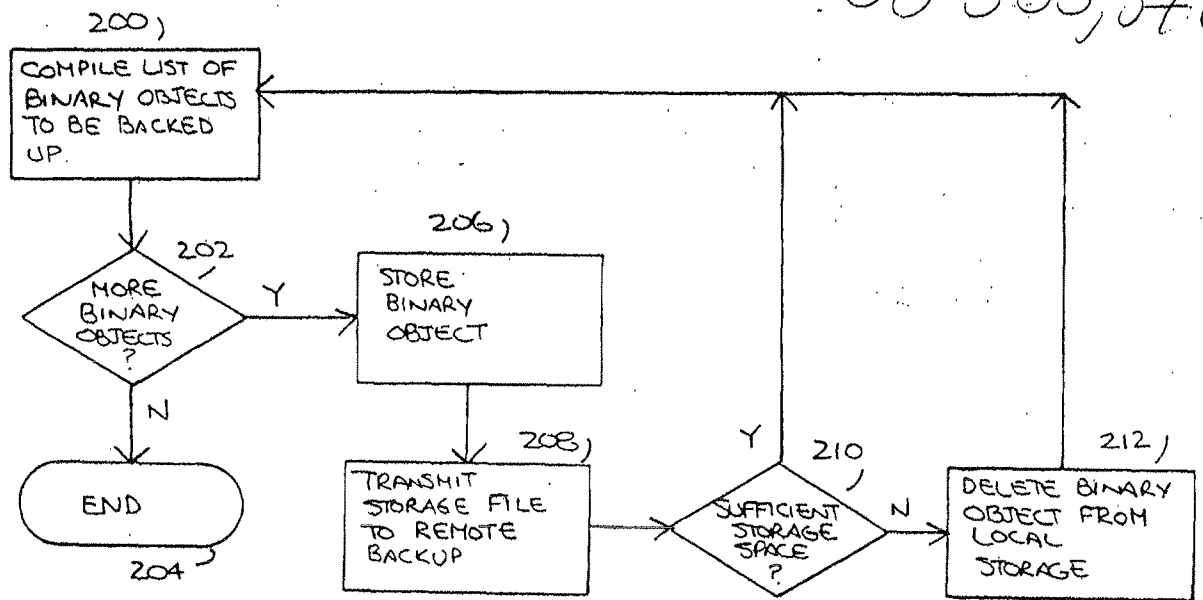
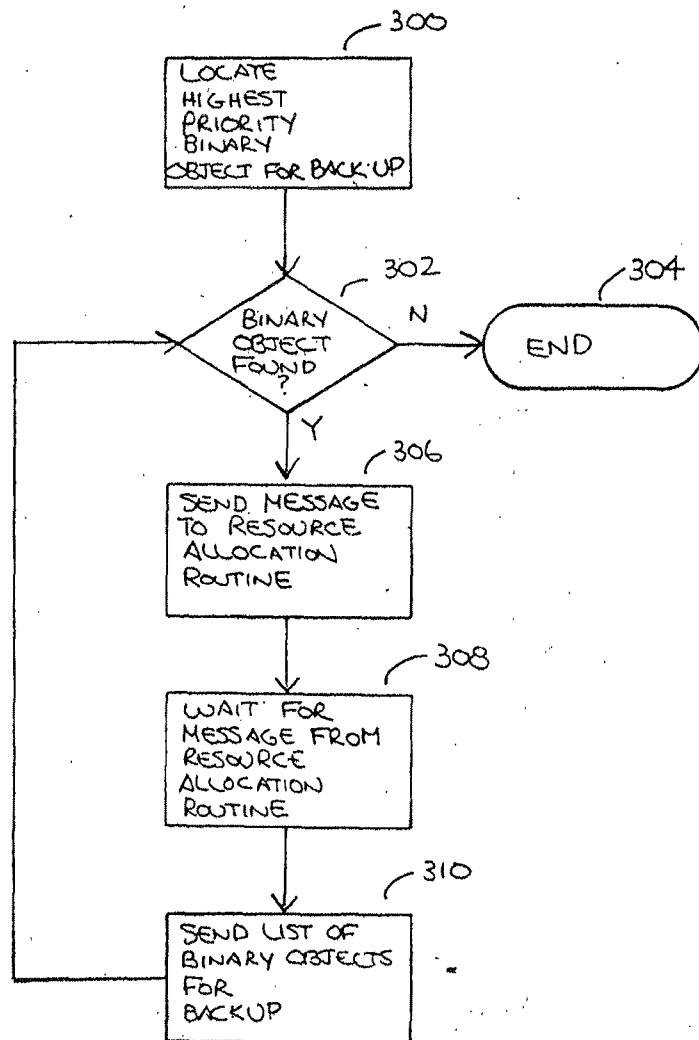
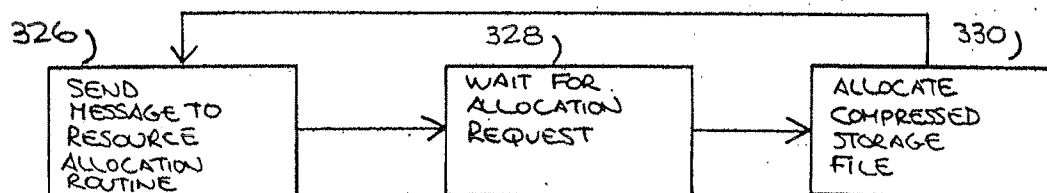
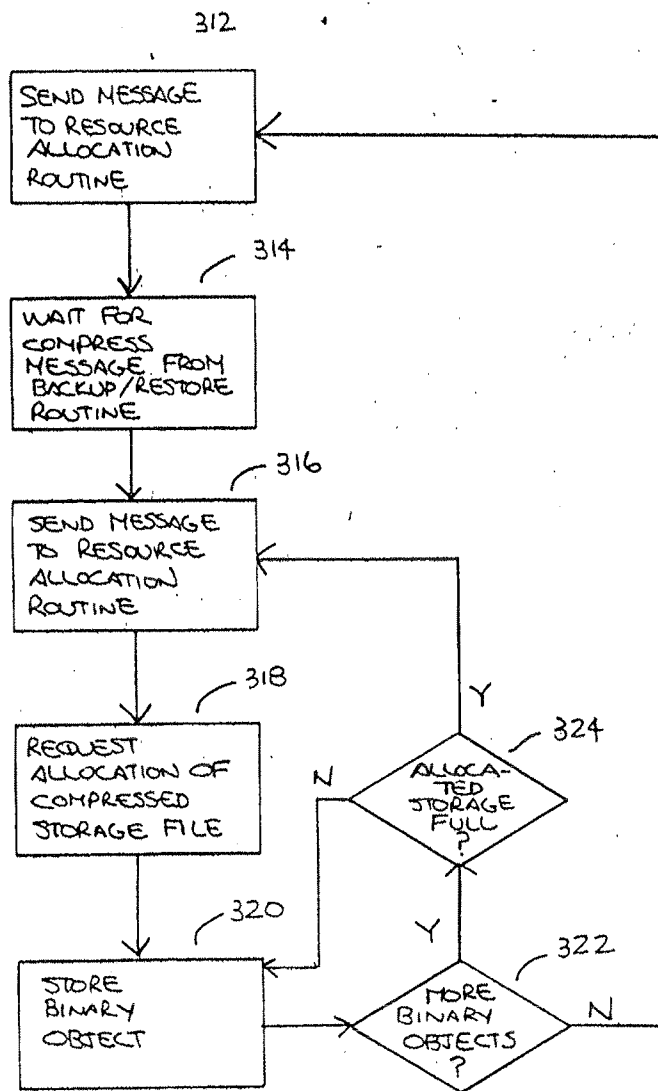


FIG. 5B



08-555,376  
085596



08-555,376

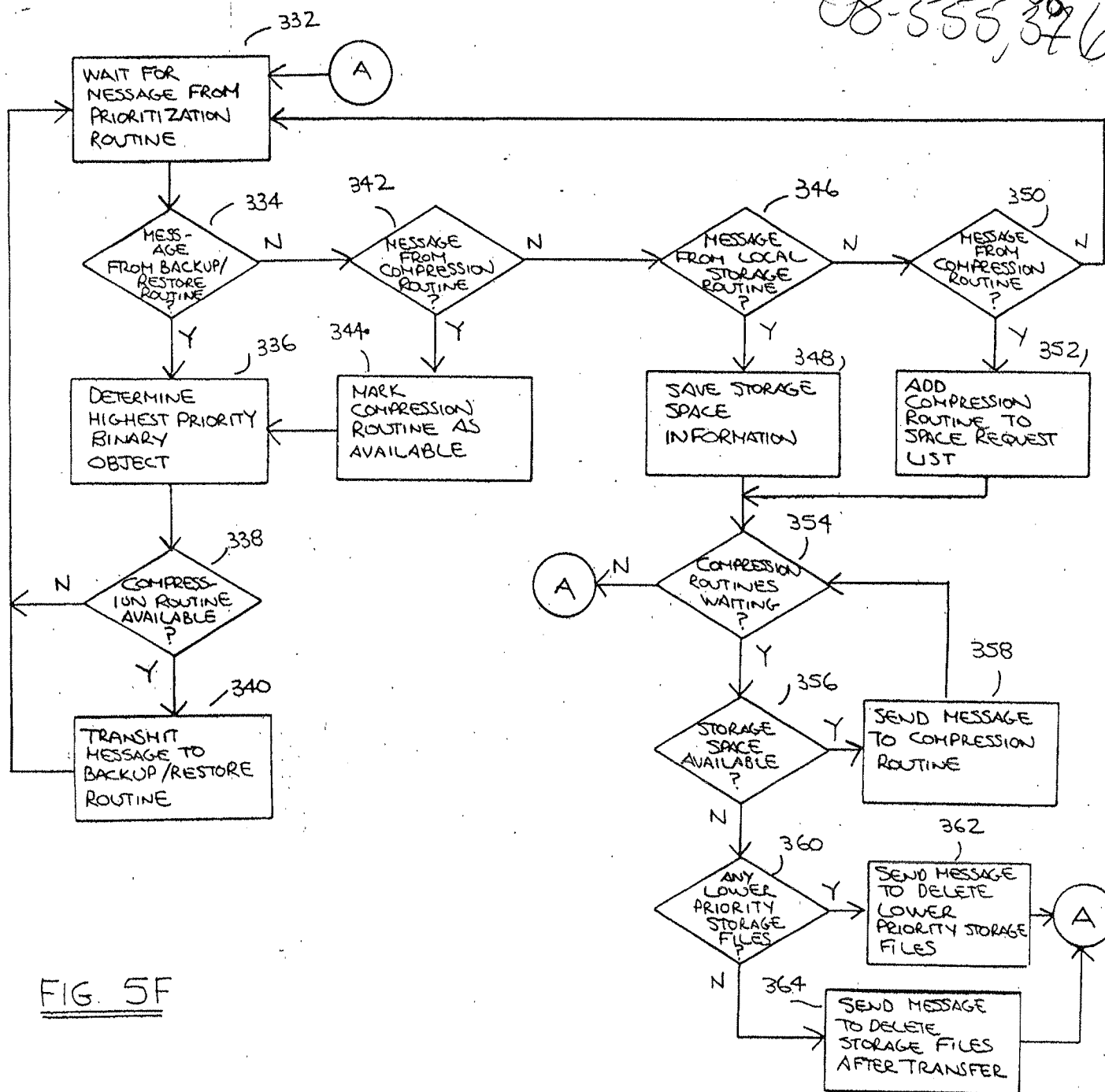


FIG. 5F

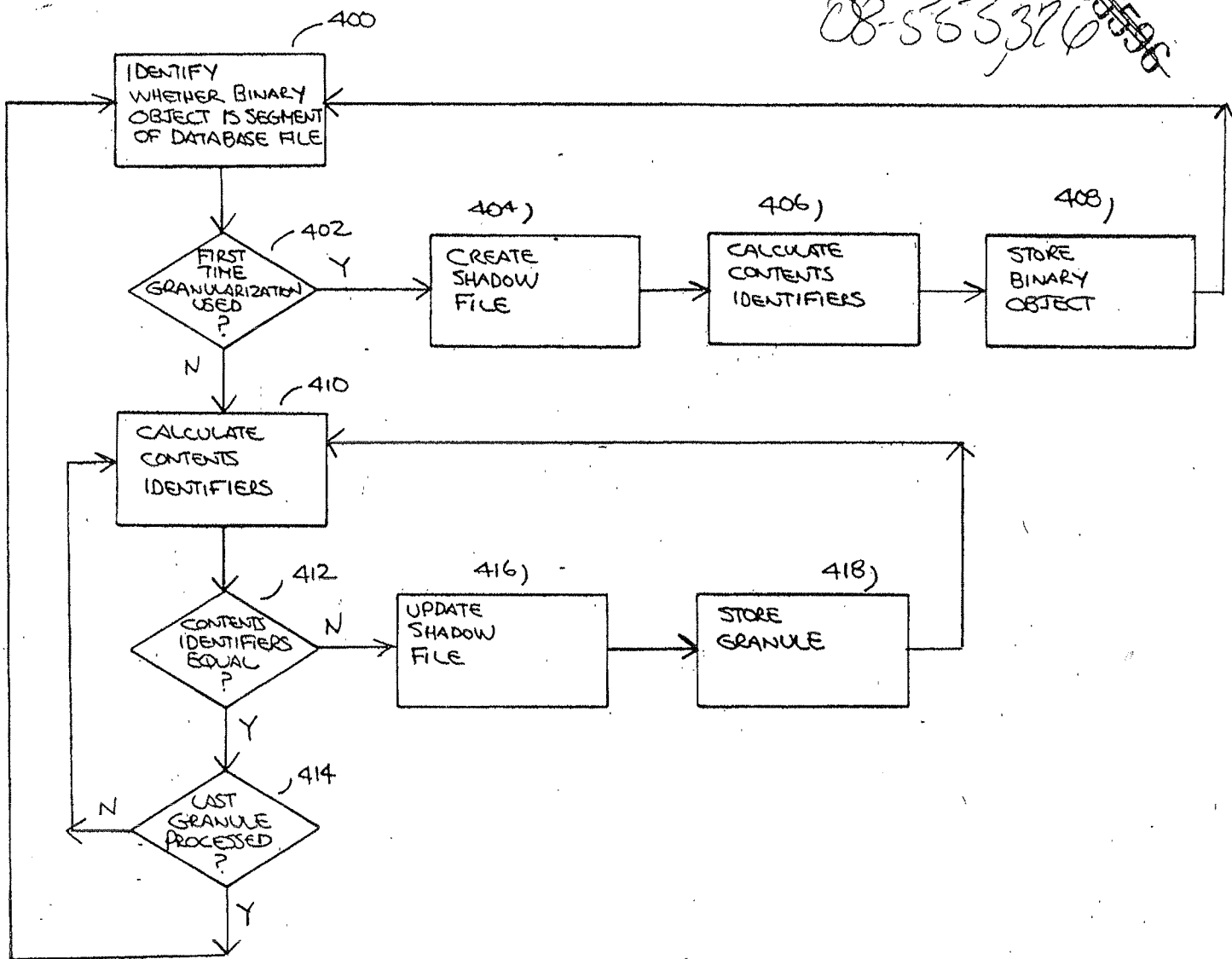


FIG. 5G

08-555,326  
~~08-555,326~~

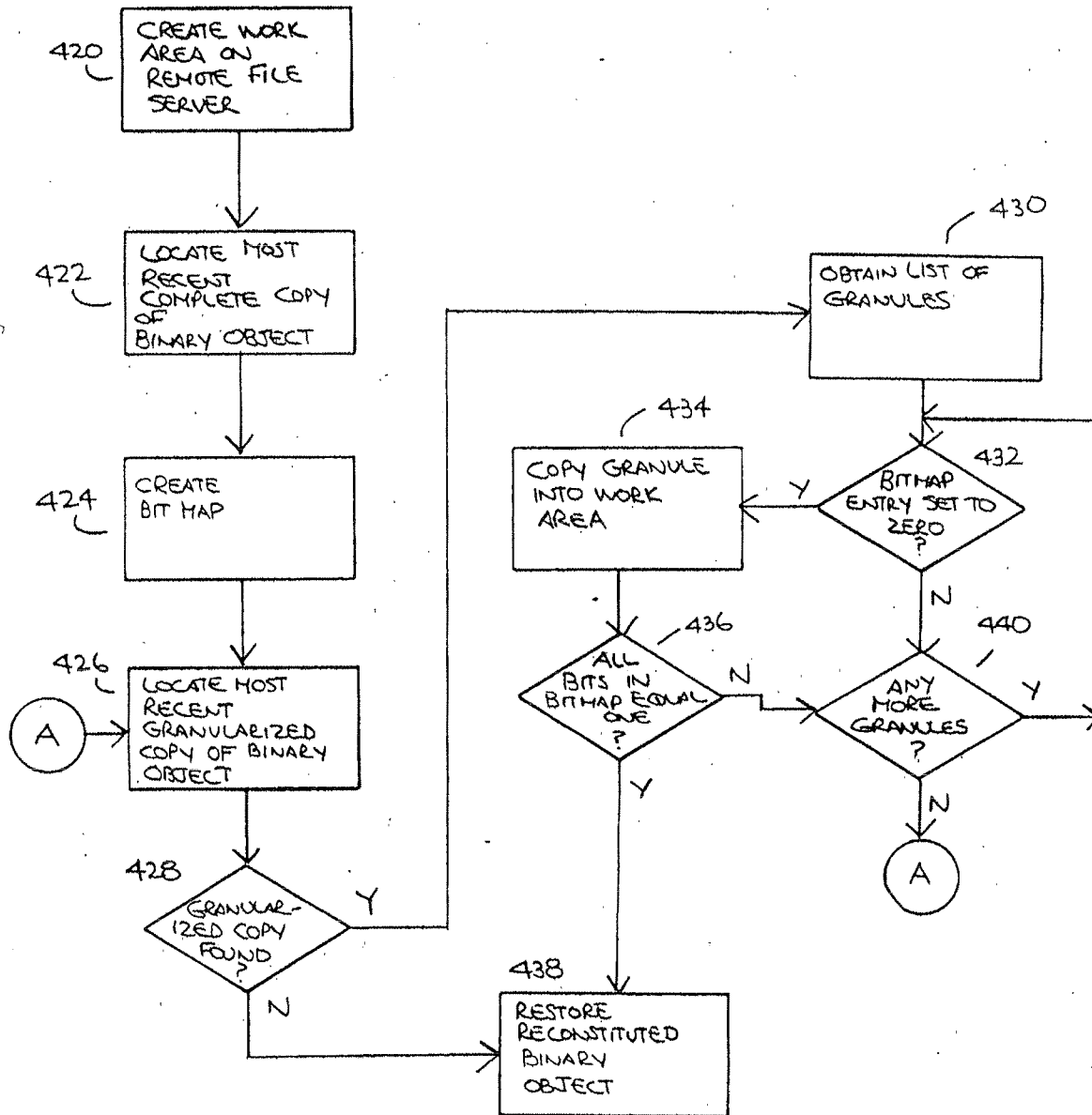


FIG. 5H

08-555370  
#385596

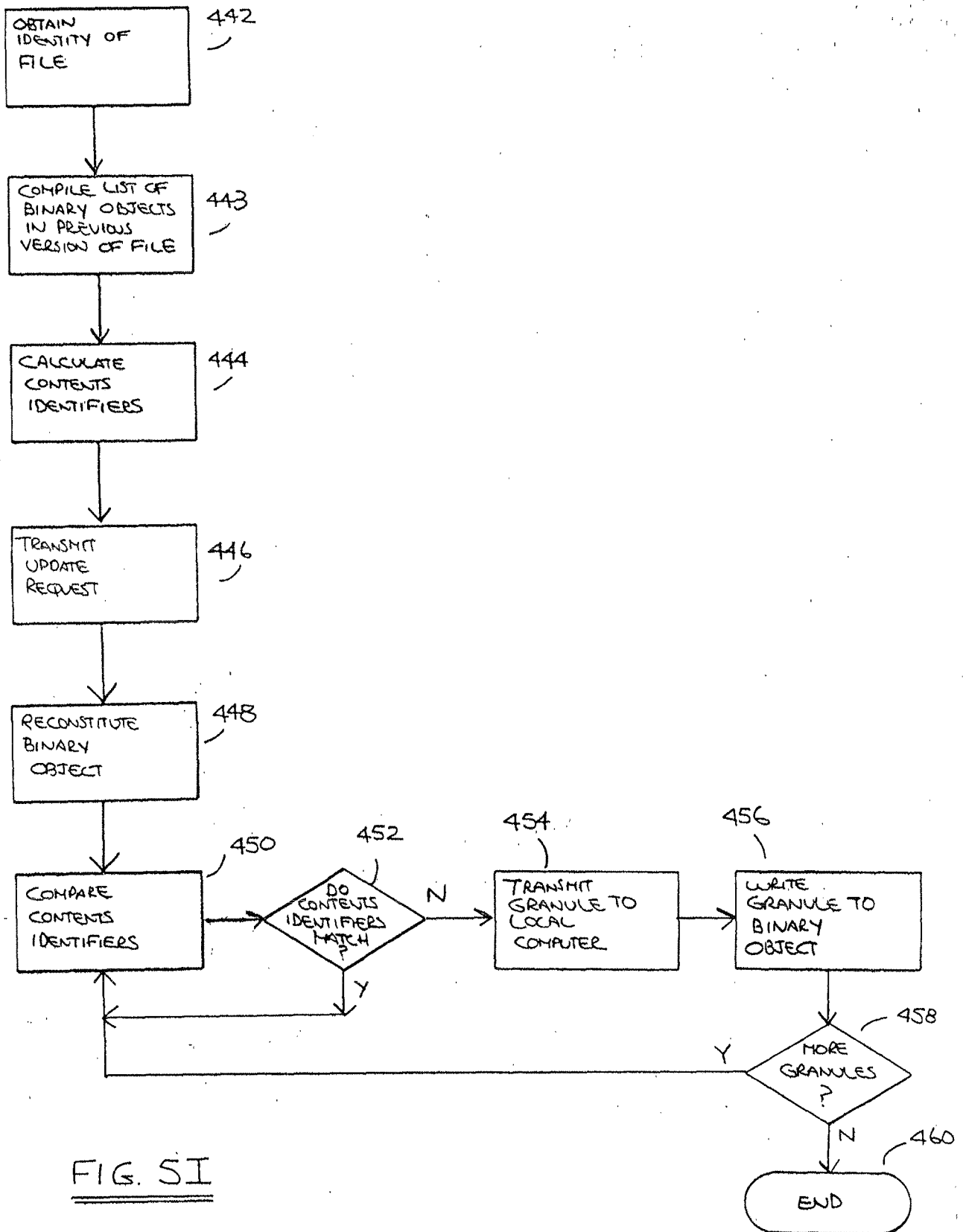


FIG. 5I

08-555,374

~~08-555,374~~

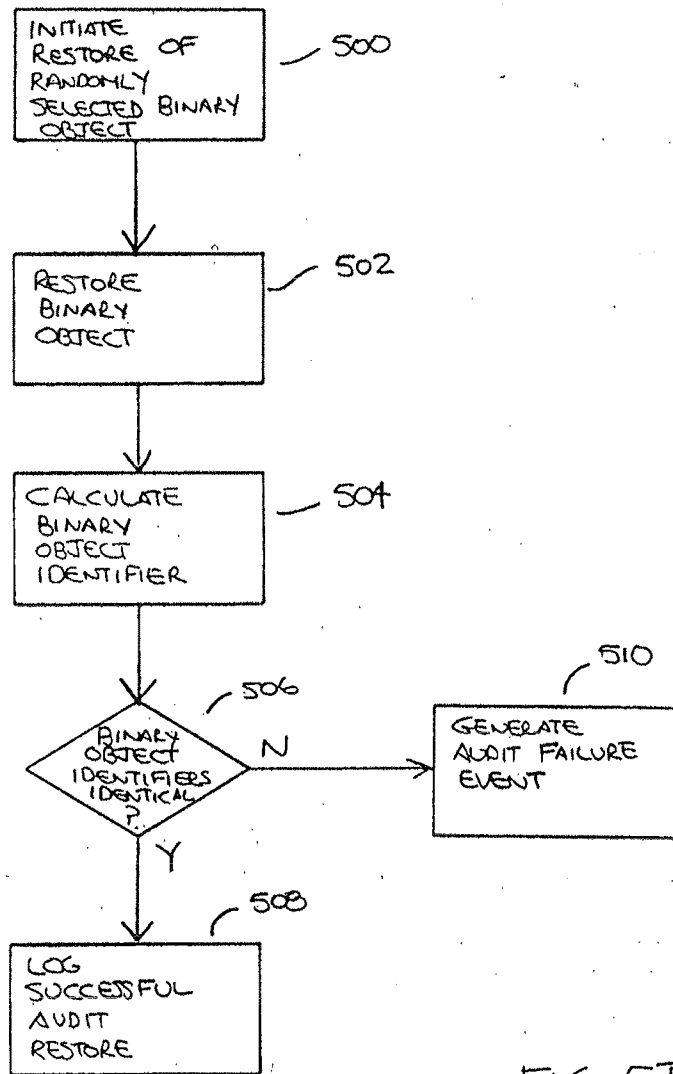


FIG. 5J



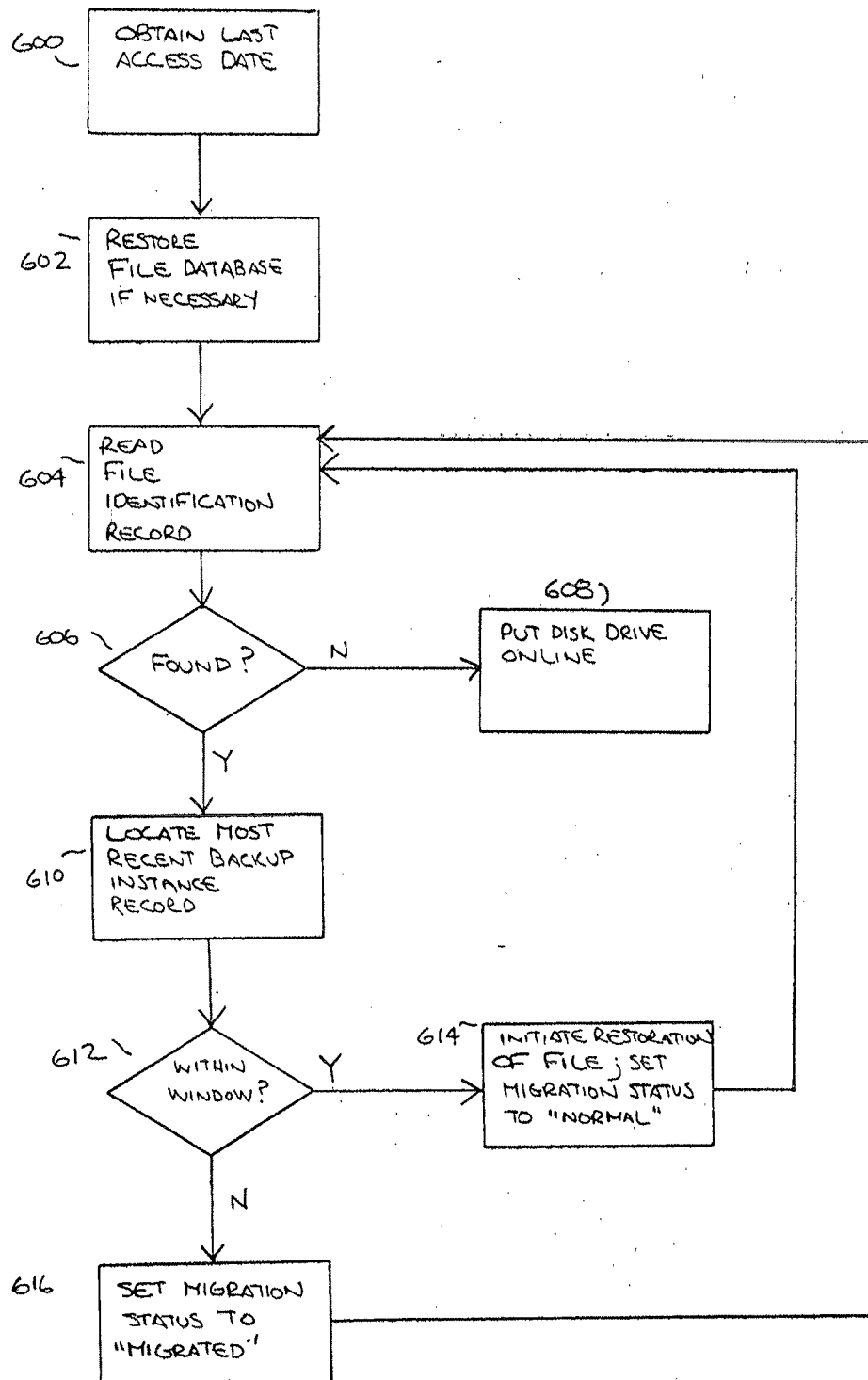


FIG. 5K

08.555376

~~08.555376~~

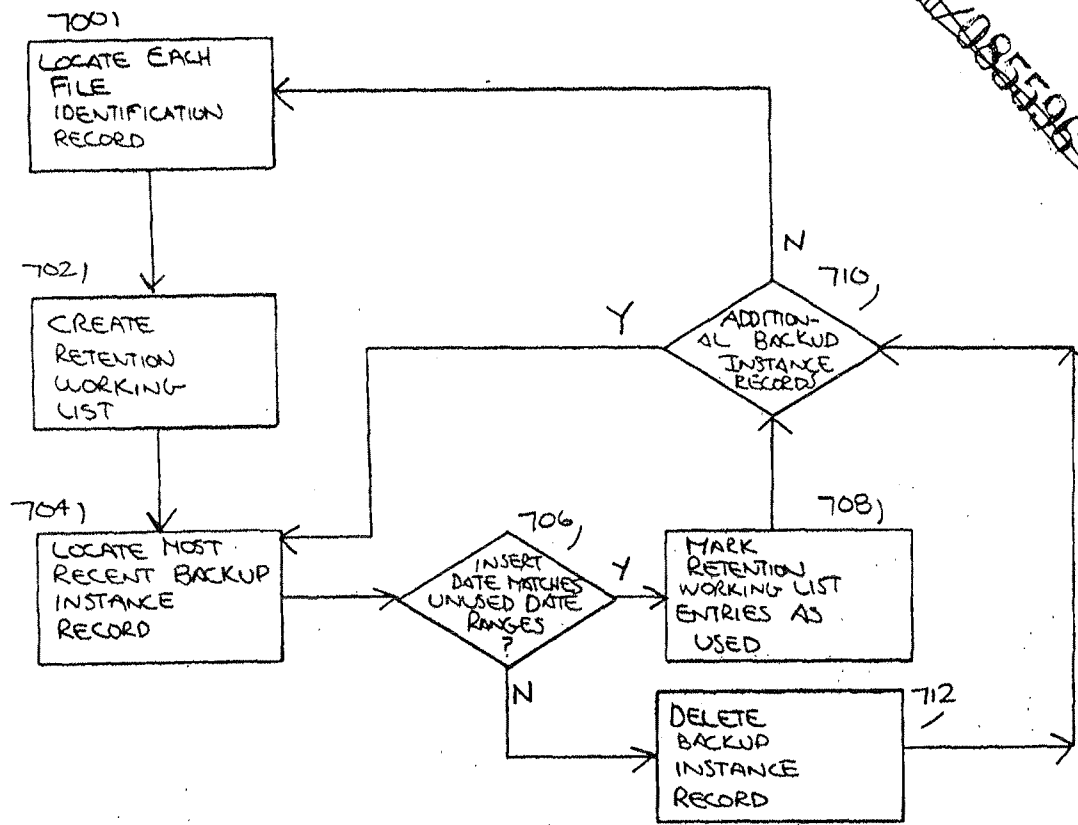


FIG. 5L



PATENT  
93085

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Art Unit 2307  
Examiner Homere

In re application of  
James R. Woodhill et al.

Serial No. 08/085,596

Filed July 1, 1993

:

:

:

:

:

SYSTEM AND METHOD FOR  
DISTRIBUTED STORAGE  
MANAGEMENT ON NETWORKED  
COMPUTER SYSTEMS

Group No. 2300

RECEIVED

GROUP 2300

ENTRIES  
BEGIN  
WITH  
A 2  
VIA  
A.M.C.  
1-5

AMENDMENT

Pittsburgh, Pennsylvania 15222

February 20, 1995

Hon. Commissioner of Patents  
and Trademarks

Washington, DC 20231

Sir:

In response to the Office Action dated 21 October 1994,  
please amend the above-identified application as follows:

In the title

Please amend the title of the above-identified application  
by adding at the end thereof --Using Binary Object Identifiers--.

In the claims

Please amend the claims as follows:

1. (Once Amended) A system for distributed management of the storage space and data on a networked computer system wherein the networked computer system includes at least two storage devices for storing data files comprised of one or more binary objects, said distributed storage management system comprising:  
means for selectively copying the binary objects stored on one of the storage devices to another of the storage devices;

a2

means for calculating a current value for a binary object identifier for selected binary objects stored on the storage devices wherein said calculation of said binary object identifier is based upon the actual data contents of the associated binary object[;], said calculated binary object identifier being used as the identifier for the selected binary object;

means for storing said current value of said binary object identifier as a previous value of said binary object identifier;

means for comparing said current value of said binary object identifier associated with a particular binary object to one or more previous values of said binary object identifier associated with that particular binary object; and

means for [commanding] controlling said means for selectively copying binary objects in response to said means for comparing.

Claim 7, line 5, cancel "commanding" and substitute therefor --controlling--.

Claim 8, line 2, cancel "commanding" and substitute therefor --controlling--.

Claim 8, line 3, cancel "commanding" and substitute therefor --instructing--.

9. (Once amended) A system for distributed management of the storage space and data on a networked computer system wherein the networked computer system includes at least two storage devices for storing data/files comprised of one or more binary objects, said distributed storage management system comprising:

means for segmenting the binary objects into granules of data;

means for selectively copying said granules stored on one of the storage devices to another of the storage devices;

means for calculating a current value for a granule identifier for selected granules stored on the storage devices wherein said calculation of said granule identifier is based upon the actual data contents of said associated granule[;], said calculated granule identifier being used as the identifier for the selected granule;

means for storing said current value of said granule identifier as a previous value of said granule identifier;

means for comparing said current value of said granule identifier associated with a particular granule to one or more previous values of said granule identifier associated with said particular granule; and

means for [commanding] controlling said means for selectively copying granules in response to said means for comparing.

Claim 13, line 16, cancel "commanding" and insert therefor --controlling--.

18. (Amended) A method for management of the storage space and data on a computer system wherein the computer system includes at least two storage areas for storing data files comprised of one or more binary objects, said method comprising the steps of:

selectively copying the binary objects stored in one of the storage areas to another of the storage areas;

calculating a current value for a binary object identifier for selected binary objects stored in the storage areas wherein said calculation of said binary object identifier is based upon the actual data contents of the associated binary object[;].

calculated binary object identifier being used as the identifier  
for the selected binary object;

storing said current value of said binary object identifier  
as a previous value of said binary object identifier;

comparing said current value of said binary object  
identifier associated with a particular binary object to one or  
more previous values of said binary object identifier associated  
with that particular binary object; and

controlling said step for selectively copying binary objects  
in response to said step for comparing.

#### Remarks

In response to paragraph 1 of the Office Action, the title of the invention has been amended to indicate that binary object identifiers are used in the management of storage space on a network computer system. It is applicants' position that the title, as amended, is descriptive of the invention. If the Examiner remains of the position that the amended title is not indicative of the invention, the Examiner is respectfully requested to suggest wording changes to the amended title.

In response to paragraph 2 of the Office Action, applicants acknowledge that the instant application has been filed with informal drawings. Formal drawings will be submitted when allowable subject matter is indicated.

In paragraph 3 of the Office Action, claims 1, 9, and 13 stand rejected under 35 U.S.C. § 112 because the phrase "means for commanding . . . comparing" is alleged to be ambiguous. Claims 1, 9, and 13 have been amended to recite "means for controlling". The means for controlling controls the means for selectively copying in response to the means for comparing. As amended, it is believed that the 35 U.S.C. § 112 rejection is now overcome.

08/555376

PATENT APPLICATION SERIAL NO. \_\_\_\_\_

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

070 DF 12/18/95 08555376

1 101

750.00 CK



Attorney's Docket No. 93085CON

PATENT *Sub*

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE *21-96*

Anticipated Classification of this application:

Class \_\_\_\_\_ Subclass \_\_\_\_\_

Application No.: 08/ 085 596

PRIOR APPLICATION Examiner: Homere, J.

Art Unit: 2307

Box FWC  
Assistant Commissioner for Patents  
Washington, D.C. 20231

FILE WRAPPER CONTINUING APPLICATION (FWC) TRANSMITTAL  
(37 C.F.R. 1.62)

- WARNING:** This form cannot be used where the parent case may not be abandoned because the filing of a request under the FWC procedure "will be considered to be a request to expressly abandon the prior application as of the filing date granted to the continuing application." 37 C.F.R. 1.62(g).
- WARNING:** This procedure can only be used for a pending application prior to payment of the issue fee (37 C.F.R. 1.62(a)), except if the parent application was withdrawn under 37 C.F.R. 1.313(b)(5) "to permit consideration of an information disclosure statement under 1.97 in a continuing application." 37 C.F.R. 1.62(a).
- WARNING:** The filing of an application at the United States stage of an international application requires an oath or declaration. 37 C.F.R. 1.61(a)(4).
- WARNING:** The claims of this new application may be finally rejected in the first Office action where all claims of the new application are drawn to the same invention claimed in the earlier application and would have been properly finally rejected on the grounds or art of record in the next Office action if they had been entered in the earlier application. MPEP § 706.07(b).
- WARNING:** An application under 37 C.F.R. 1.62 is filed by making changes by amendment to the prior application, (37 C.F.R. 1.62(a)), and not by filing a new application. 37 C.F.R. 1.62(e).
- WARNING:** Filing under 37 C.F.R. 1.62 is permitted only if filed by the same or less than all the inventors named in the prior application. 37 C.F.R. 1.62(a).

CERTIFICATION UNDER 37 C.F.R. 1.10

I hereby certify that this FWC Transmittal and the documents referred to as attached therein are being deposited with the United States Postal Service on this date November 9, 1995, in an envelope as "Express Mail Post Office to Addressee," mailing Label Number EG 229434922US addressed to the: Assistant Commissioner for Patents, Washington, D.C. 20231.

Beth H. Retort

(type or print name of person mailing paper)

*Beth H. Retort*  
Signature of person mailing paper

**NOTE:** Each paper or fee filed by "Express Mail" must have the number of the "Express Mail" mailing label placed thereon prior to mailing. (37 C.F.R. 1.10(b)).

**WARNING:** Certificate of mailing (first class) or facsimile transmission procedures of 37 C.F.R. 1.8 cannot be used to obtain a date of mailing or transmission for this correspondence.



**WARNING:** File wrapper continuing procedure filings can only be based on a prior complete application as defined by § 1.51(a)(1), and not a prior complete provisional application as defined by § 1.51(a)(2).

This is a request for a filing under the file wrapper continuing application procedure (37 C.F.R. 1.62), for a

- ☒ continuation
- ☐ divisional
- ☐ continuation-in-part (for oath or declaration, see III below)

Attached is an amendment for added subject matter

- ☐ continuing application to permit consideration of an information disclosure statement under 37 C.F.R. 1.97.

**NOTE:** The filing date under 37 C.F.R. 1.62(a) is "... the date on which a request is filed for an application ... including identification of the application number and applicant's name of the prior application." The prior application under 37 C.F.R. 1.62(a) must be "... a prior complete application," as defined in 37 C.F.R. 1.51(a)(1).

#### **PARTICULARS OF PRIOR NONPROVISIONAL APPLICATION**

**WARNING:** File wrapper continuing procedure filings can only be based on a prior complete application as defined by § 1.51(a)(1), and not a prior complete provisional application as defined by § 1.51(a)(2). 37 C.F.R. 1.62(a).

- A. Application No. 08 / 085,596 filed July 1, 1993 date.
- B. Title (as originally filed SYSTEM & METHOD FOR DISTRIBUTED STORAGE  
and as last amended) MANAGEMENT ON NETWORKED COMPUTER SYSTEMS  
SYSTEM & METHOD FOR DISTRIBUTED STORAGE
- C. Name of applicant(s) (as originally filed and as last amended) and current  
MANAGEMENT ON NETWORKED COMPUTER SYSTEMS  
correspondence address of applicant(s)

1-00 1. FULL NAME OF INVENTOR	FAMILY NAME <u>Woodhill</u>	FIRST GIVEN NAME <u>James</u>	SECOND GIVEN NAME <u>R.</u>
RESIDENCE & CITIZENSHIP	CITY <u>Houston</u>	STATE OR FOREIGN COUNTRY Texas TX	COUNTRY OF CITIZENSHIP U.S.A.
POST OFFICE ADDRESS	POST OFFICE ADDRESS 1960 Haddon Avenue	CITY Houston	STATE & ZIP CODE/COUNTRY TX 77019
2-00 2. FULL NAME OF INVENTOR	FAMILY NAME <u>Woodhill</u>	FIRST GIVEN NAME <u>Louis</u>	SECOND GIVEN NAME <u>R.</u>
RESIDENCE & CITIZENSHIP	CITY <u>Richmond</u>	STATE OR FOREIGN COUNTRY Texas TX	COUNTRY OF CITIZENSHIP U.S.A.
POST OFFICE ADDRESS	POST OFFICE ADDRESS 2114 Pecan Trail Drive	CITY Richmond	STATE & ZIP CODE/COUNTRY Texas 77469
3-00 3. FULL NAME OF INVENTOR	FAMILY NAME <u>More, Jr.</u>	FIRST GIVEN NAME <u>William</u>	SECOND GIVEN NAME <u>Russell</u>
RESIDENCE & CITIZENSHIP	CITY <u>Houston</u>	STATE OR FOREIGN COUNTRY Texas TX	COUNTRY OF CITIZENSHIP U.S.A.
POST OFFICE ADDRESS	POST OFFICE ADDRESS 786 Thicket	CITY Houston	STATE & ZIP CODE/COUNTRY Texas 77079

☒ Continued on Added Page for Inventor's Data

4. FULL NAME OF INVENTOR

FAMILY NAME: Berlin

FIRST GIVEN NAME: Jay

SECOND GIVEN NAME: Harris

RESIDENCE & CITIZENSHIP

CITY: Houston

STATE OR FOREIGN COUNTRY: Texas

COUNTRY OF CITIZENSHIP: U.S.A.

POST OFFICE ADDRESS

POST OFFICE ADDRESS: 9111 Landdowne Drive

CITY: Houston

STATE & ZIP CODE/COUNTRY: TX 77096

5. FULL NAME OF INVENTOR

FAMILY NAME:

FIRST GIVEN NAME:

SECOND GIVEN NAME:

RESIDENCE & CITIZENSHIP

CITY:

STATE OR FOREIGN COUNTRY:

COUNTRY OF CITIZENSHIP:

POST OFFICE ADDRESS

POST OFFICE ADDRESS:

CITY:

STATE & ZIP CODE/COUNTRY:

6. FULL NAME OF INVENTOR

FAMILY NAME:

FIRST GIVEN NAME:

SECOND GIVEN NAME:

RESIDENCE & CITIZENSHIP

CITY:

STATE OR FOREIGN COUNTRY:

COUNTRY OF CITIZENSHIP:

POST OFFICE ADDRESS

POST OFFICE ADDRESS:

CITY:

STATE & ZIP CODE/COUNTRY:

(FWC [4-2.1]—Added Page)

The above identified application, in which no payment of issue fee, abandonment of (other than where the above identified application was abandoned under 37 C.F.R. 1.313(b)(5) to permit consideration of an information disclosure statement under 37 C.F.R. 1.97), or termination of proceedings has occurred, is hereby expressly abandoned as of the filing date of this new application. Please use all the contents of the prior application file wrapper, including the drawings, as the basic papers for the new application.

It is understood that secrecy under 35 U.S.C 122 is hereby waived to the extent that if information or access is available to any one of the applications in the file wrapper of a 37 C.F.R. 1.62 application, be it either this application or a prior application in the same file wrapper, the PTO may provide similar information or access to all the other applications in the same file wrapper.

## II. Inventorship statement

*NOTE: "If the continuation, continuation-in-part, or divisional application is filed by less than all the inventors named in the prior application a statement must accompany the application when filed requesting deletion of the names of the person or persons who are not inventors of the invention being claimed in the continuation, continuation-in-part, or divisional application." 37 C.F.R. 1.62(a).*

*NOTE: "In the case of a continuation-in-part application which adds and claims additional disclosure by amendment, an oath or declaration as required by § 1.63 must be filed. In those situations where a new oath or declaration is required due to additional subject matter being claimed, additional inventors may be named in the continuing application. In a continuation or divisional application which discloses and claims only subject matter disclosed in a prior application, no additional oath or declaration is required and the application must name as inventors the same or less than all the inventors in the prior application." 37 C.F.R. 1.60(c).*

*(complete applicable item (a), (b) and/or (c) below)*

- (a) ☒ This application discloses and claims only subject matter disclosed in the prior application whose particulars are set out above and the inventor(s) in this application are
- ☒ the same.
- ☐ less than those named in the prior application. It is requested that the following inventor(s) identified above for the prior application be deleted:

---

*(type name(s) of inventor(s) to be deleted)*

- (b) ☐ This application discloses and claims additional disclosure by amendment and a new declaration or oath is being filed. With respect to the prior application whose particulars are set out above, the inventor(s) in this application are
- ☐ the same.
- ☐ Add the following additional inventor(s).

---

*(type name of inventor(s) to be added)*

- (c) The inventorship for all the claims in this application is
- ☒ the same.
- ☐ not the same. An explanation, including the ownership of the various claims at the time the last claimed invention was made, is submitted.

### III. Declaration or oath

#### A. Continuation or divisional

☒ None required.

#### B. Continuation-in-part

☐ Attached.

Executed by

*(check all applicable items)*

☐ inventor(s).

☐ legal representative of inventor(s). 37 C.F.R. 1.42 or 1.43.

☐ joint inventor or person showing a proprietary interest for inventor who refused to sign or cannot be reached. 37 C.F.R. 1.47;

☐ This is the petition required by 37 C.F.R. 1.47 and the statement required by 37 C.F.R. 1.47 is also attached.

*(See item VIII below for fee.)*

☐ Not attached.

☐ Application is made by a person authorized under 37 C.F.R. 1.41(c) on behalf of all of the above named applicant(s).

*(The declaration or oath, along with the surcharge required by 37 C.F.R. 1.16(e) can be filed subsequently.)*

☐ Attached is a showing that the filing is authorized.

*(Not required unless called into question. 37 C.F.R. 1.41(d))*

### IV. Identification of Claims for Further Prosecution

**WARNING:** "The claims of a new application may be finally rejected in the first Office action in those situations where (1) the new application is a continuing application of, or a substitute for, an earlier application, and (2) all the claims of the new application (a) are drawn to the same invention claimed in the earlier application, and (b) would have been properly finally rejected on the grounds of art of record in the next Office action if they had been entered in the earlier application." MPEP § 706.07(b).

☒ The fees to be charged are to be based on the number of claims remaining as a result of the:

☒ attached preliminary amendment.

☐ the unentered amendment filed under 37 C.F.R. 1.116 in the prior application, which is now repeated.

☐ the claims as on file in the prior application.

## V. Fee Calculation (37 C.F.R. 1.16)

NOTE: The filing fee for a continuation, continuation-in-part, or divisional application is based on the number of claims remaining in the application after entry of any preliminary amendment and entry of any amendments under 37 C.F.R. 1.116 unentered in the prior application which is requested to be entered in this FWC application. 37 C.F.R. 1.62.

CLAIMS FOR FEE CALCULATION				
Number Filed		Number Extra	Rate	Basic Fee 37 C.F.R. 1.16(a) \$750.00
Total Claims (37 C.F.R. 1.16(c))	4 - 20 =	0	×	\$ 22.00
Independent Claims (37 C.F.R. 1.16(b))	2 - 3 =	0	×	\$ 78.00
Multiple dependent claim(s), if any (37 C.F.R. 1.16(d))			+	\$250.00

☐ The fee for extra claims is not being paid at this time.

Filing fee calculation

\$ 750.00

NOTE: If the fees for extra claims are not paid on filing they must be paid or the claims cancelled by amendment, prior to the expiration of the time period set for response by the Patent and Trademark Office in any notice of fee deficiency. 37 C.F.R. 1.16(d).

## VI. Petition for Suspension of Prosecution for the Time Necessary to File an Amendment

NOTE: Where it is possible that the claims on file will give rise to a first action final for this continuation application and for some reason an amendment cannot be filed promptly (e.g., experimental data is being gathered), it may be desirable to file a petition for suspension of prosecution for the time necessary.

(check the next item, if applicable)

☐ There is provided herewith a Petition to Suspend Prosecution for the time Necessary to File an Amendment (New Application Filed Concurrently).

## VII. Small Entity Statement

☐ A verified statement that this is a filing by a small entity is attached.

**WARNING:** "Status as a small entity in one application or patent does not affect any other application or patent, including applications or patents which are directly or indirectly dependent upon the application or patent in which the status has been established. A nonprovisional application claiming benefit under 35 U.S.C. 119(e), 120, 121 or 365(c) of a prior application may rely on a verified statement filed in the prior application if the nonprovisional application includes a reference to a verified statement in the prior application or includes a copy of the verified statement filed in the prior application if status as a small entity is still proper and desired." 37 C.F.R. § 1.28(a).

(complete the following, if applicable)

- ☐ Status as a small entity was claimed in prior application  
\_\_\_\_\_ / \_\_\_\_\_, filed on \_\_\_\_\_, from which benefit is being  
claimed for this application under:

35 U.S.C. ☐ 120,  
☐ 121,  
☐ 365(c),

and which status as a small entity is still proper and desired.

- ☐ A copy of the verified statement in the prior application is included.

Reduced filing fee calculation (50% of above) \$\_\_\_\_\_

NOTE: 37 C.F.R. 1.28(a) states "Status as a small entity must be specifically established by a verified statement filed in each application or patent in which the status is available and desired, except those applications filed under § 1.60 or § 1.62 of this part where the status as a small entity has been established in a parent application and is still proper."

The last sentence of 37 C.F.R. 1.28(a) states: "Applications filed under § 1.60 or § 1.62 of this part must include a reference to a verified statement in a parent application if status as a small entity is still proper and desired."

Any excess of the full fee paid will be refunded if a verified statement and a refund request are filed within 2 months of the date of timely payment of a full fee then the excess fee paid will be refunded on request. 37 C.F.R. 1.28(a).

#### VIII. Fee Payment Being Made at This Time

##### Not attached

- ☐ No filing fee is submitted.  
(This and the surcharge required by 37 C.F.R. 1.16(e) can be paid subsequently.)

##### Attached

- ☒ filing fee \$ 750.00
- ☐ recording assignment  
(\$40.00; 37 C.F.R. 1.21(h)).  
For payment of fee see item XIV below. \$ \_\_\_\_\_
- ☐ petition fee for filing by other than all the  
inventors or person not the inventor where  
inventor refused to sign or cannot be reached  
(\$130.00; 37 C.F.R. 1.47 and 1.17(h)) \$ \_\_\_\_\_
- ☐ processing and retention fee  
(\$130.00; 37 C.F.R. 1.53(d) and 1.21(l)) \$ \_\_\_\_\_

NOTE: 37 C.F.R. 1.21(l) establishes a fee for processing and retaining any application that is abandoned for failing to complete the application pursuant to 37 C.F.R. 1.53(d) and this, as well as the changes to 37 C.F.R. 1.53 and 1.78, indicate that in order to obtain the benefit of a prior U.S. application, either the basic filing fee must be timely paid or the processing and retention fee in § 1.21(l) must be paid within 1 year from the notification under § 1.53(d).

Total fees enclosed \$ 750.00

## IX. Method of Payment of Fees

- ☒ Attached is check in the amount of \$ 750.00
- ☐ Charge Account No. \_\_\_\_\_ in the amount of \$ \_\_\_\_\_
- ☐ A duplicate of this request is attached.

NOTE: Fees should be itemized in such a manner that it is clear for which purpose the fees are paid. 37 C.F.R. 1.22(b).

## X. Authorization to Charge Additional Fees

WARNING: If no fee payment is made at this time, this item should not be completed.

WARNING: Accurately count claims, especially multiple dependent claims, to avoid unexpected high charges if extra claim charges are authorized.

- ☒ The Commissioner is hereby authorized to charge the following additional fees that may be required by this paper and during the entire pendency of this application to Account No. 11-1110:
- ☒ 37 C.F.R. 1.16(a), (f) or (g) (filing fees)
- ☒ 37 C.F.R. 1.16(b), (c) and (d) (presentation of extra claims)

NOTE: Because additional fees for excess or multiple dependent claims not paid on filing or on later presentation must only be paid or these claims cancelled by amendment prior to the expiration of the time period set for response by the PTO in any notice of fee deficiency (37 C.F.R. 1.16(d)), it might be best not to authorize the PTO to charge additional claim fees, except possibly when dealing with amendments after final action.

- ☒ 37 C.F.R. 1.16(e) (surcharge for filing the basic filing fee and/or declaration on a date later than the filing date of the application)
- ☒ 37 C.F.R. 1.17 (application processing fees)

WARNING: While 37 C.F.R. 1.17(a), (b), (c) and (d) deal with extensions of time under § 1.136(a), this authorization should be made only with the knowledge that: "Submission of the appropriate extension fee under 37 C.F.R. 1.136(a) is to no avail unless a request or petition for extension is filed" (Emphasis added). Notice of November 5, 1985 (1060 O.G. 27).

- ☐ 37 C.F.R. 1.18 (issue fee at or before mailing of Notice of Allowance, pursuant to 37 C.F.R. 1.311(b))

NOTE: Where an authorization to charge the issue fee to a deposit account has been filed before the mailing of a Notice of Allowance, the issue fee will be automatically charged to the deposit account at the time of mailing the Notice of Allowance. 37 C.F.R. 1.311(b).

37 C.F.R. 1.28(b) states: (a) notification of change of status must be made even if the fee is paid as "other than a small entity" and (b) no notification is required if the change is to another small entity. Notification of any change of status resulting in loss of entitlement to small entity status must be filed in the application prior to, or at the time of, paying the issue fee. 37 C.F.R. 1.28(b).

## XI. Instructions as to Overpayment

- ☐ Credit Account No. \_\_\_\_\_
- ☒ Refund



**XII. Priority—35 U.S.C. 119(a)-(d)**

- ☐ Priority of Application No. \_\_\_\_\_ / \_\_\_\_\_ filed on \_\_\_\_\_  
in \_\_\_\_\_ (country) is claimed under 35 U.S.C. 119.
- ☐ The certified copy has been filed on \_\_\_\_\_ in prior U.S. applica-  
tion Serial No. 0 / \_\_\_\_\_, which prior application was filed on  
\_\_\_\_\_.
- ☐ Certified copy will follow.

**XIII. Relate Back**

**WARNING:** If an application claims the benefit of the filing date of an earlier filed application under 35 U.S.C. 120, 121 or 365(c), the 20-year term of that application will be based upon the filing date of the earliest U.S. application that the application makes reference to under 35 U.S.C. 120, 121 or 365(c). (35 U.S.C. 154(a)(2) does not take into account, for the determination of the patent term, any application on which priority is claimed under 35 U.S.C. 119, 365(a) or 365(b).) For a c-i-p application, applicant should review whether any claim in the patent that will issue is supported by an earlier application and, if not, the applicant should consider canceling the reference to the earlier filed application. The term of a patent is not based on a claim-by-claim approach. See Notice of April 14, 1995, 60 Fed. Reg. 20,194, at 20,205.

**NOTE:** "Any nonprovisional application claiming the benefit of one or more prior filed copending nonprovisional applications or international applications designating the United States of America must contain or be amended to contain in the first sentence of the specification following the title a reference to each such prior application, identifying it by application number (consisting of the series code and serial number) or international application number and international filing date and indicating the relationship of the applications. Cross-references to other related applications may be made when appropriate. (See § 1.14(b))." 37 C.F.R. § 1.78(2).

(complete the following, if applicable)

- ☒ Amend the specification by inserting, before the first line, the sentence:

**A. 35 U.S.C. 119(e)**

**NOTE:** "Any nonprovisional application claiming the benefit of one or more prior filed copending provisional applications must contain or be amended to contain in the first sentence of the specification following the title a reference to each such prior provisional application, identifying it as a provisional application, and including the provisional application number (consisting of series code and serial number)." 37 C.F.R. § 1.78(a)(4).

**WARNING:** While this application under 37 C.F.R. § 1.62 cannot be a file wrapper continuation application of a provisional application, the nonprovisional application giving rise to this FWC filing could claim the benefit of a provisional application.

- ☐ "This application claims the benefit of U.S. Provisional Application(s) No(s).:

**APPLICATION NO(S).:**

**FILING DATE**

_____ / _____	_____
_____ / _____	_____
_____ / _____	_____ "

**B. 35 U.S.C. 120, 121 and 365(c)**

NOTE: "Any nonprovisional application claiming the benefit of one or more prior filed copending nonprovisional applications or international applications designating the United States of America must contain or be amended to contain in the first sentence of the specification following the title a reference to each such prior application, identifying it by application number (consisting of the series code and serial number) or international application number and international filing date and indicating the relationship of the applications. Cross-references to other related applications may be made when appropriate. (See § 1.14(b))." 37 C.F.R. § 1.78(2).

- ☒ "This application is a
- ☒ continuation
  - ☐ divisional
  - ☐ continuation-in-part

12/30/96  
JRH  
of copending application(s)

☒ Serial Number 0 8/ 085,596 filed on July 1, 1993, now abandoned.

☐ International Application \_\_\_\_\_ filed on \_\_\_\_\_ and  
which designated the U.S."

NOTE: The proper reference to a prior filed PCT application that entered the U.S. national phase is the U.S. serial number and the filing date of the PCT application that designated the U.S.

**XIV. Assignment**

- ☒ The prior application is assigned of record to \_\_\_\_\_  
Legent Corporation
- ☐ An assignment of the invention to \_\_\_\_\_

is attached. A separate ☐ "COVER SHEET FOR ASSIGNMENT (DOCUMENT)  
ACCOMPANYING NEW PATENT APPLICATION" or ☐ FORM PTO 1595 is also  
attached.

NOTE: "If an assignment is submitted with a new application, send two separate letters—one for the application  
and one for the assignment." Notice of May 4, 1990 (1114 O.G. 77-78).

**XV. Power of Attorney**

The power of attorney in the prior application is to

Christopher H. Gebhardt

33,130

Attorney

Reg. No.

- a. ☒ The power appears in the original papers in the prior application.
- b. ☐ The power does not appear in the original papers, but was filed on \_\_\_\_\_.
- c. ☐ A new power has been executed and is attached.
- d. ☒ Address all future communications to:

(item d may only be completed by applicant, or attorney or agent of record.)

Edward L. Pencoske

29,688

Name

Reg. No.

Kirkpatrick & Lockhart LLP

Address

1500 Oliver Building

(412) 355-8645

Tel. No.

Pittsburgh, PA 15222

#### **XVI. Maintenance of Copendency of Prior Application**

*(this item must be completed and the necessary papers filed in the prior application, if the period set in the prior application has run.)*

- ☒ A petition, fee and response has been filed to extend the term in the pending prior application until November 11, 1995.

NOTE: The PTO finds it useful if a copy of the petition filed in the prior application extending the term for response is filed with the papers constituting the filing of the continuation application. Notice of November 5, 1985 (1060 O.G. 27).

- ☒ A copy of the petition for extension of time in the prior application is attached.

#### **XVII. Conditional Petitions for Extension of Time in Prior Application**

*(complete this item and file conditional petition in prior application, if previous item is not applicable)*

- ☐ A conditional petition for extension of time is being filed in the pending prior application

NOTE: The PTO finds it useful if a copy of the petition filed in the prior application extending the term for response is filed with the papers constituting the filing of the continuation application. Notice of November 5, 1985 (1060 O.G. 27).

- ☐ A copy of the conditional petition for extension of time in the prior application is attached.

#### **XVIII. Abandonment of Prior Application**

- ☒ Please abandon the prior application at a time while the prior application is pending or when the petition for extension of time or to revive in that application is granted and when this application is granted a filing date so as to make this application copending with said prior application. At the same time, please add the words "now abandoned" to the amendment to the specification set forth in XIII above.

NOTE: According to the Notice of May 13, 1983 (103 TMOG 6-7), the filing of a continuation or continuation-in-part application is a proper response with respect to a petition for extension of time or a petition to revive and should include the express abandonment of the prior application conditioned upon the granting of the petition and the granting of a filing date to the continuing application.

NOTE: "A registered attorney or agent acting under the provisions of § 1.34(a), or of record, may also expressly abandon a prior application as of the filing date granted to a continuing application when filing such a continuing application." 37 C.F.R. 1.138.

#### **XIX. Information Disclosure Statement**

- ☐ Submitted herewith is an Information Disclosure Statement.

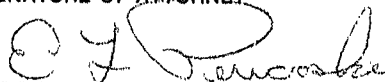
Assignment recorded in PTO on October 21, 1993

Reel 6737 Frame 0936

- ☐ Plus ADDED PAGE FOR INVENTOR'S DATA FOR FWC FILING  
☐ Plus ASSIGNMENT (DOCUMENT) COVER LETTER ACCOMPANY-  
ING NEW PATENT APPLICATION

Reg. No. 29,688

SIGNATURE OF ATTORNEY



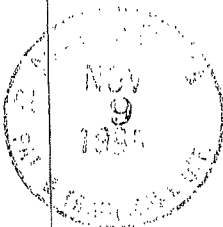
Edward L. Pencoske

(type or print name of attorney)

Tel. No.: (412) 355-8645

Kirkpatrick & Lockhart LLP

P.O. Address 1500 Oliver Building  
Pittsburgh, PA 15222



PATENT  
93085CON

2-1-96

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Art Unit  
Examiner :

In re application of : SYSTEM AND METHOD FOR  
DISTRIBUTED MANAGEMENT ON  
James R. Woodhill et al. : NETWORKED COMPUTER SYSTEMS

Serial No. :

Filed: November 9, 1995 : Group

PRELIMINARY AMENDMENT

Pittsburgh, Pennsylvania 15222

November 9, 1995

Hon. Commissioner of Patents  
and Trademarks

Washington, DC 20231

Sir:

Preliminary to the examination of the continuation  
application filed herewith, please amend that application as  
follows.


In the Claims

Please cancel claims 2 - 17.

REMARKS

This Preliminary Amendment is filed to reduce the number of claims for purposes of calculating the filing fee. Applicants intend to file a second preliminary amendment to either amend or replace claims 1 and 18 - 20.

Respectfully submitted,

A handwritten signature in dark ink, appearing to read "E. L. Pencoske", written over a horizontal line.

Edward L. Pencoske  
Registration No. 29,688

KIRKPATRICK & LOCKHART  
1500 Oliver Building  
Pittsburgh, PA 15222

(412) 355-8645



Attorney's Docket No. 93085CON

**PATENT**

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

In re application of: James R. Woodhill et al.

Serial No.: 08 / 555,376 Group No.:

Filed: November 9, 1995 Examiner:

For: SYSTEM AND METHOD FOR DISTRIBUTED MANAGEMENT ON  
NETWORKED COMPUTER SYSTEMS

Commissioner of Patents and Trademarks

Washington, D.C. 20231

**AMENDMENT TRANSMITTAL**

1. Transmitted herewith is an amendment for this application.

**STATUS**

2. Applicant is

- ☐ a small entity. A verified statement:  
☐ is attached.  
☐ was already filed.  
☒ other than a small entity.

RECEIVED

JAN 24 1996

RECEIVED 2300

FEB 20 1996

GROUP 2300

**CERTIFICATE OF MAILING/TRANSMISSION (37 CFR 1.8a)**

I hereby certify that this correspondence is, on the date shown below, being:

**MAILING**

☒ deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to the Commissioner of Patents and Trademarks, Washington, D.C. 20231

**FACSIMILE**

☐ transmitted by facsimile to the Patent and Trademark Office

Date: 5 JAN 1996

Signature

Edward L. Pencoske

(type or print name of person certifying)

(Amendment Transmittal [9-19]—page 1 of 4)

### EXTENSION OF TERM

NOTE: "Extension of Time in Patent Cases (Supplement Amendments) — If a timely and complete response has been filed after a Non-Final Office Action, an extension of time is not required to permit filing and/or entry of an additional amendment after expiration of the shortened statutory period.

*If a timely response has been filed after a Final Office Action, an extension of time is required to permit filing and/or entry of a Notice of Appeal or filing and/or entry of an additional amendment after expiration of the shortened statutory period unless the timely-filed response placed the application in condition for allowance. Of course, if a Notice of Appeal has been filed within the shortened statutory period, the period has ceased to run." Notice of December 10, 1985 (1061 O.G. 34-35).*

NOTE: See 37 CFR 1.645 for extensions of time in interference proceedings and 37 CFR 1.550(c) for extensions of time in reexamination proceedings.

3. The proceedings herein are for a patent application and the provisions of 37 CFR 1.136 apply

(complete (a) or (b) as applicable)

- (a) ☐ Applicant petitions for an extension of time under 37 CFR 1.136 (fees: 37 CFR 1.17(a)-(d) for the total number of months checked below:

Extension (months)	Fee for other than small entity	Fee for small entity
<input type="checkbox"/> one month	\$ 110.00	\$ 55.00
<input type="checkbox"/> two months	\$ 370.00	\$185.00
<input type="checkbox"/> three months	\$ 870.00	\$435.00
<input type="checkbox"/> four months	\$1,360.00	\$680.00

Fee \$\_\_\_\_\_

If an additional extension of time is required please consider this a petition therefor.

(check and complete the next item, if applicable)

- ☐ An extension for \_\_\_\_\_ months has already been secured and the fee paid therefor of \$\_\_\_\_\_ is deducted from the total fee due for the total months of extension now requested.

Extension fee due with this request \$\_\_\_\_\_

OR

- (b) ☒ Applicant believes that no extension of term is required. However, this conditional petition is being made to provide for the possibility that applicant has inadvertently overlooked the need for a petition for extension of time.



### FEE FOR CLAIMS

4. The fee for claims (37 CFR 1.16(b)-(d)) has been calculated as shown below:

	(Col. 1)		(Col. 2)		(Col. 3)	SMALL ENTITY	OTHER THAN A SMALL ENTITY
	CLAIMS REMAINING AFTER AMENDMENT		HIGHEST NO PREVIOUSLY PAID FOR		PRESENT EXTRA	ADDIT. RATE FEE	OR RATE ADDIT. FEE
TOTAL	18	MINUS	20	=	0	x11 = \$ 0	x22 = \$ 0
INDEP.	2	MINUS	3	=	0	x38 = \$ 0	x76 = \$ 0
<input type="checkbox"/> FIRST PRESENTATION OF MULTIPLE DEP. CLAIM						+120 = \$	+240 = \$
						TOTAL ADDIT. FEE \$	OR TOTAL 0 ADDIT. FEE \$

\* If the entry in Col. 1 is less than entry in Col. 2, write "0" in Col. 3.

\*\* If the "Highest No. Previously Paid for" IN THIS SPACE is less than 20, enter "20".

\*\*\* If the "Highest No. Previously Paid For" IN THIS SPACE is less than 3, enter "3".

The "Highest No. Previously Paid For" (Total or indep.) is the highest number found in the appropriate box in Col. 1 of a prior amendment or the number of claims originally filed.

**WARNING:** "After final rejection or action (§ 1.113) amendments may be made cancelling claims or complying with any requirement of form which has been made." 37 CFR § 1.116(a) (emphasis added).

(complete (c) or (d) as applicable)

(c) ☒ No additional fee for claims is required.

**OR**

(d) ☐ Total additional fee for claims required \$\_\_\_\_\_.

### FEE PAYMENT

5. ☐ Attached is a check in the sum of \$\_\_\_\_\_.

☐ Charge Account No. \_\_\_\_\_ the sum of \$\_\_\_\_\_.

A duplicate of this transmittal is attached.

### FEE DEFICIENCY

NOTE: If there is a fee deficiency and there is no authorization to charge an account, additional fees are necessary to cover the additional time consumed in making up the original deficiency. If the maximum, six-month period has expired before the deficiency is noted and corrected, the application is held abandoned. In those instances where authorization to charge is included, processing delays are encountered in returning the papers to the PTO Finance Branch in order to apply these charges prior to action on the cases. Authorization to charge the deposit account for any fee deficiency should be checked. See the Notice of April 7, 1986, (1065 O.G. 31-33).

6. ☒ If any additional extension and/or fee is required, charge Account No. 11-1110.

### AND/OR

- ☒ If any additional fee for claims is required, charge  
Account No. 11-1110.



SIGNATURE OF ATTORNEY

Reg. No.: 29,688

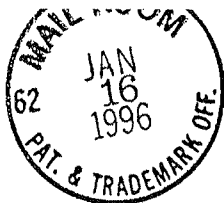
Edward L. Pencoske

type or print name of attorney

Tel. No.: (412) 355-8645

Kirkpatrick & Lockhart LLP

P.O. Address 1500 Oliver Building  
Pittsburgh, PA 15222



COPY "D"

PATENT  
93085CON

2301  
10-30-96  
IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

RECEIVED

FEB 20 1996

Art Unit  
Examiner

GROUP 2300

In re application of

: SYSTEM AND METHOD FOR  
DISTRIBUTED MANAGEMENT ON  
NETWORKED COMPUTER SYSTEMS

James R. Woodhill et al.

Serial No. 08/555,376

Filed: November 9, 1995

: Group

RECEIVED

JAN 24 1996

SECOND PRELIMINARY AMENDMENT

GROUP 2300

Pittsburgh, Pennsylvania 15222

January 5, 1996

Hon. Commissioner of Patents  
and Trademarks

Washington, DC 20231

Sir:

Preliminary to the examination of the application filed  
November 9, 1995, please amend that application as follows.

In the Claims

Please cancel claims 1 and 18 - 20, which are all of the  
claims remaining in the application, and substitute the  
following new claims.

--21. A system for distributed management of the storage  
space and data on a networked computer system wherein the  
networked computer system includes at least two storage devices  
for storing data files, said distributed storage management  
system comprising:

means for selectively copying data files stored on one of  
the storage devices to another of the storage devices;

means for dividing each data file into one or more binary objects of a predetermined size;

means for calculating a current value for a binary object identifier for each binary object within a file, said calculation of said binary object identifier being based upon the actual data contents of the associated binary object, said calculated binary object identifier being saved as the name of the associated binary object;

means for comparing said current name of a particular binary object to one or more previous names of said binary object;

means for storing said current name of said binary object identifier; and

means for controlling said means for selectively copying binary objects in response to said means for comparing.

22. The distributed storage management system of claim 21 wherein said means for calculating said current name includes means for calculating a current name comprised of at least two independently determined values.

23. The distributed storage management system of claim 21 wherein said means for calculating said current name includes means for calculating a 128-bit binary value comprised of four 32-bit fields.

24. The distributed storage management system of claim 23 wherein said four 32-bit fields include a binary object identifier size field, a Cyclical Redundancy Check number field calculated against the contents of the binary object, a Longitudinal Redundancy Check number field calculated against the contents of the binary object, and a binary object hash.

number field calculated against the contents of the binary object.

25. The distributed storage management system of claim 21 further including means for auditing the performance of said distributed storage management system, said means for auditing including:

second means for controlling said means for selectively copying binary objects to recopy a previously copied binary object;

means for recalculating said binary object identifier for the recopied binary object, said recalculated binary object identifier being saved as the name of the associated binary object;

means for comparing said recalculated name to a previous name of said binary object; and

means for reporting a failure if said recalculated name is not identical to said previous name of said binary object.

26. The distributed storage management system of claim 21 wherein said means for controlling said means for selectively copying includes means for instructing, in response to said means for comparing, said means for selectively copying to copy a particular binary object only if its current name is not identical to a previous name for that particular binary object.

27. The distributed storage management system of claim 21 additionally comprising means for segmenting the binary objects into granules of data, and wherein said granules of data are processed in the same manner as said binary objects.

28. The distributed storage management system of claim 27 further including means for reconstructing a binary object from a most recent complete copy of the binary object, said means for reconstructing including:

means for copying said granules copied by said means for selectively copying granules to said most recent complete copy of the binary object in order from most-recently copied granule to least-recently copied granule; and

means for generating a bitmap for controlling said means for copying said copied granules.

29. The distributed storage management system of claim 28 wherein said means for calculating said current name for said granule includes means for calculating a 32-bit Cyclical Redundancy Check number calculated against the contents of said granule and means for calculating a 32-bit binary object hash number calculated against the contents of said granule.

30. The distributed storage management system of claim 28 further including means for restoring a current version of a binary object to a previous version of that binary object, said means for restoring including:

means for calculating said name for each of said granules in the current version of the binary object;

means for comparing said calculated name to a previous name for each of said granules in the current version of the binary object; and

means, responsive to said means for comparing said names, for replacing those granules in the current version of the binary object for which said names are not identical.

31. The distributed storage management system of claim 21 additionally comprising;

means for indicating which of said copied binary objects must be copied to a particular storage device in the event of a failure of that storage device before that storage device is considered to be operable by the local computer with which that storage device is in communication; and

wherein said means for controlling said means for selectively copying binary objects is further responsive to said means for indicating.

32. The distributed storage management system of claim 31 wherein said means for indicating includes means for specifying a last access date such that only binary objects that have been accessed by the networked computer system on or after said last access date must be copied to a particular storage device before that storage device is considered to be operable.

33. The distributed storage management system of claim 21 additionally comprising:

means for maintaining a file retention list wherein said file retention list includes a file retention pattern for each binary object copied by said means for selectively copying binary objects;

means for determining which of the binary objects copied by said means for selectively copying binary objects match each of said file retention patterns; and

means for deleting the binary objects from the storage devices in response to said means for determining.

34. The distributed storage management system of claim 33 wherein said file retention pattern includes daily, weekly, monthly, quarterly and yearly retention patterns.

35. The distributed storage management system of claim 21 wherein said networked computer system includes a remote backup file server, and wherein said means for selectively copying copies the binary objects stored on one of the storage devices to another of the storage devices or to the remote backup file server, said distributed storage management system additionally comprising:

means for employing user-defined priorities to determine which binary objects are to be copied to another storage device and to determine a queuing sequence for copying binary objects to the remote backup file server.

36. A method for management of the storage space and data on a computer system wherein the computer system includes at least two storage areas for storing data files, said method comprising the steps of:

dividing each data file into one or more binary objects of a predetermined size;

calculating a current value for a binary object identifier for each binary object within a file, said calculation of said binary object identifier being based upon the actual data contents of the associated binary object, said calculated binary object identifier being saved as the name of the associated binary object;

comparing said current name of said binary object to one or more previous names of said binary object;

storing said current name of said binary object as a previous name of said binary object; and



selectively copying binary objects in response to said comparing step.

37. The method of claim 36 wherein said step of calculating said current name for said binary object includes the step of calculating a current name for a binary object comprised of at least two independently determined values.

38. The method of claim 36 wherein said step of calculating said current name for said binary object includes the step of calculating a current name for a binary object utilizing a 128-bit binary value comprised of four 32-bit fields and wherein said four 32-bit fields include a binary object identifier size field, a Cyclical Redundancy Check number field calculated against the contents of the binary object, a Longitudinal Redundancy Check number field calculated against the contents of the binary object, and a binary object hash number field calculated against the contents of the binary object.--

#### REMARKS

This Second Preliminary Amendment is filed to submit new claims for examination.

In the parent application, of which the instant application is a continuation, an Office Action was issued on 11 May 1995 rejecting claims 1 - 20 under 35 U.S.C. § 103 as being unpatentable over an article to Daniel Barbara et al. ("the Barbara article"). It is the Examiner's position, as set forth on page 3 of the Office Action in the parent application, that the Barbara article discloses "a means for storing". As the Examiner noted, the Barbara article did not specifically calculate the check sums to be used as identifiers for selected

## IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Art Unit  
Examiner

147  
Route  
10-29-96

In re application of : SYSTEM AND METHOD FOR  
DISTRIBUTED MANAGEMENT ON  
James R. Woodhill et al. : NETWORKED COMPUTER SYSTEMS  
Serial No. 555,376 :  
Filed: November 9, 1995 : Group

SECOND PRELIMINARY AMENDMENT

Pittsburgh, Pennsylvania 15222

January 5, 1996

Hon. Commissioner of Patents  
and Trademarks

Washington, DC 20231

RECEIVED

OCT 24/1996

GROUP 2800

Sir:

Preliminary to the examination of the application filed  
November 9, 1995, please amend that application as follows.

In the Claims

Please cancel claims 1 and 18 - 20, which are all of the  
claims remaining in the application, and substitute the  
following new claims.

1. A system for distributed management of the storage  
space and data on a networked computer system wherein the  
networked computer system includes at least two storage devices  
for storing data files, said distributed storage management  
system comprising:

means for selectively copying data files stored on one of  
the storage devices to another of the storage devices;

means for calculating a current value for a binary object identifier for each binary object within a file, said calculation of said binary object identifier being based upon the actual data contents of the associated binary object, said calculated binary object identifier being saved as the name of the associated binary object;

means for comparing said current name of a particular binary object to one or more previous names of said binary object;

means for storing said current name of said binary object identifier; and

means for controlling said means for selectively copying binary objects in response to said means for comparing.

2/  
22. The distributed storage management system of claim 21 wherein said means for calculating said current name includes means for calculating a current name comprised of at least two independently determined values.

3/  
23. The distributed storage management system of claim 21 wherein said means for calculating said current name includes means for calculating a 128-bit binary value comprised of four 32-bit fields.

4/  
24. The distributed storage management system of claim 23 wherein said four 32-bit fields include a binary object identifier size field, a Cyclical Redundancy Check number field calculated against the contents of the binary object, a Longitudinal Redundancy Check number field calculated against the contents of the binary object, and a binary object hash

<sup>5</sup>  
~~25~~. The distributed storage management system of claim ~~21~~<sup>1</sup> further including means for auditing the performance of said distributed storage management system, said means for auditing including:

second means for controlling said means for selectively copying binary objects to recopy a previously copied binary object;

means for recalculating said binary object identifier for the recopied binary object, said recalculated binary object identifier being saved as the name of the associated binary object;

means for comparing said recalculated name to a previous name of said binary object; and

means for reporting a failure if said recalculated name is not identical to said previous name of said binary object.

<sup>6</sup>  
D<sup>1</sup> ~~26~~. The distributed storage management system of claim ~~21~~<sup>1</sup> wherein said means for controlling said means for selectively copying includes means for instructing, in response to said means for comparing, said means for selectively copying to copy a particular binary object only if its current name is not identical to a previous name for that particular binary object.

<sup>7</sup>  
~~27~~. The distributed storage management system of claim ~~21~~<sup>1</sup> additionally comprising means for segmenting the binary objects into granules of data, and wherein said granules of data are processed in the same manner as said binary objects.

a most recent complete copy of the binary object, said means for reconstructing including:

means for copying said granules copied by said means for selectively copying granules to said most recent complete copy of the binary object in order from most-recently copied granule to least-recently copied granule; and

means for generating a bitmap for controlling said means for copying said copied granules.

<sup>9</sup>~~29~~. The distributed storage management system of claim <sup>8</sup>~~28~~ wherein said means for calculating said current name for said granule includes means for calculating a 32-bit Cyclical Redundancy Check number calculated against the contents of said granule and means for calculating a 32-bit binary object hash number calculated against the contents of said granule.

<sup>10</sup>~~30~~. The distributed storage management system of claim <sup>8</sup>~~28~~ further including means for restoring a current version of a binary object to a previous version of that binary object, said means for restoring including:

means for calculating said name for each of said granules in the current version of the binary object;

means for comparing said calculated name to a previous name for each of said granules in the current version of the binary object; and

means, responsive to said means for comparing said names, for replacing those granules in the current version of the binary object for which said names are not identical.

means for indicating which of said copied binary objects must be copied to a particular storage device in the event of a failure of that storage device before that storage device is considered to be operable by the local computer with which that storage device is in communication; and

wherein said means for controlling said means for selectively copying binary objects is further responsive to said means for indicating.

12/ 32. The distributed storage management system of claim 31 wherein said means for indicating includes means for specifying a last access date such that only binary objects that have been accessed by the networked computer system on or after said last access date must be copied to a particular storage device before that storage device is considered to be operable.

7- 13/ 33. The distributed storage management system of claim 31 additionally comprising:

means for maintaining a file retention list wherein said file retention list includes a file retention pattern for each binary object copied by said means for selectively copying binary objects;

means for determining which of the binary objects copied by said means for selectively copying binary objects match each of said file retention patterns; and

means for deleting the binary objects from the storage devices in response to said means for determining.

monthly, quarterly and yearly retention patterns.

15/35. The distributed storage management system of claim 21 wherein said networked computer system includes a remote backup file server, and wherein said means for selectively copying copies the binary objects stored on one of the storage devices to another of the storage devices or to the remote backup file server, said distributed storage management system additionally comprising:

means for employing user-defined priorities to determine which binary objects are to be copied to another storage device and to determine a queuing sequence for copying binary objects to the remote backup file server.

16/36. A method for management of the storage space and data on a computer system wherein the computer system includes at least two storage areas for storing data files, said method comprising the steps of:

dividing each data file into one or more binary objects of a predetermined size;

calculating a current value for a binary object identifier for each binary object within a file, said calculation of said binary object identifier being based upon the actual data contents of the associated binary object, said calculated binary object identifier being saved as the name of the associated binary object;

comparing said current name of said binary object to one or more previous names of said binary object;

storing said current name of said binary object as a previous name of said binary object; and

comparing step.

<sup>17</sup>  
37. The method of claim <sup>16</sup>36 wherein said step of calculating said current name for said binary object includes the step of calculating a current name for a binary object comprised of at least two independently determined values.

<sup>18</sup>  
38. The method of claim <sup>16</sup>36 wherein said step of calculating said current name for said binary object includes the step of calculating a current name for a binary object utilizing a 128-bit binary value comprised of four 32-bit fields and wherein said four 32-bit fields include a binary object identifier size field, a Cyclical Redundancy Check number field calculated against the contents of the binary object, a Longitudinal Redundancy Check number field calculated against the contents of the binary object, and a binary object hash number field calculated against the contents of the binary object.--

#### REMARKS

This Second Preliminary Amendment is filed to submit new claims for examination.

In the parent application, of which the instant application is a continuation, an Office Action was issued on 11 May 1995 rejecting claims 1 - 20 under 35 U.S.C. § 103 as being unpatentable over an article to Daniel Barbara et al. ("the Barbara article"). It is the Examiner's position, as set forth on page 3 of the Office Action in the parent application, that the Barbara article discloses "a means for storing". As the Examiner noted, the Barbara article did not specifically calculate the check sums to be used as identifiers for selected



SERIAL NUMBER	FILING DATE	FIRST NAMED APPLICANT	ATTORNEY DOCKET NO.
---------------	-------------	-----------------------	---------------------

08/555,376 11/09/95 WOODHILL

J	93085CON
EXAMINER	

E3M1/0102

EDWARD L. PENCOSKE  
KIRKPATRICK & LOCKHART  
1500 OLIVER BUILDING  
PITTSBURGH PA 15222

HOMER, J	PAPER NUMBER
----------	--------------

2307

DATE MAILED: 01/02/97

### NOTICE OF ALLOWABILITY

#### PART I.

1. ☒ This communication is responsive to the amendment of 10/29/96
2. ☒ All the claims being allowable, PROSECUTION ON THE MERITS IS (OR REMAINS) CLOSED in this application. If not included herewith (or previously mailed), a Notice Of Allowance And Issue Fee Due or other appropriate communication will be sent in due course.
3. ☒ The allowed claims are 21-38, now renumbered as 1-18
4. ☐ The drawings filed on \_\_\_\_\_ are acceptable.
5. ☐ Acknowledgment is made of the claim for priority under 35 U.S.C. 119. The certified copy has [ ] been received. [ ] not been received. [ ] been filed in parent application Serial No. \_\_\_\_\_, filed on \_\_\_\_\_.
6. ☐ Note the attached Examiner's Amendment.
7. ☐ Note the attached Examiner Interview Summary Record, PTOL-413.
8. ☒ Note the attached Examiner's Statement of Reasons for Allowance.
9. ☐ Note the attached NOTICE OF REFERENCES CITED, PTO-892.
10. ☐ Note the attached INFORMATION DISCLOSURE CITATION, PTO-1449.

#### PART II.

A SHORTENED STATUTORY PERIOD FOR RESPONSE to comply with the requirements noted below is set to EXPIRE THREE MONTHS FROM THE "DATE MAILED" indicated on this form. Failure to timely comply will result in the ABANDONMENT of this application. Extensions of time may be obtained under the provisions of 37 CFR 1.136(a).

1. ☐ Note the attached EXAMINER'S AMENDMENT or NOTICE OF INFORMAL APPLICATION, PTO-152, which discloses that the oath or declaration is deficient. A SUBSTITUTE OATH OR DECLARATION IS REQUIRED.
2. ☒ APPLICANT MUST MAKE THE DRAWING CHANGES INDICATED BELOW IN THE MANNER SET FORTH ON THE REVERSE SIDE OF THIS PAPER.
  - a. ☒ Drawing informalities are indicated on the NOTICE RE PATENT DRAWINGS, PTO-948, attached hereto or to Paper No. 4. CORRECTION IS REQUIRED
  - b. ☐ The proposed drawing correction filed on \_\_\_\_\_ has been approved by the examiner. CORRECTION IS REQUIRED.
  - c. ☐ Approved drawing corrections are described by the examiner in the attached EXAMINER'S AMENDMENT. CORRECTION IS REQUIRED.
  - d. ☒ Formal drawings are now REQUIRED.

Any response to this letter should include in the upper right hand corner, the following information from the NOTICE OF ALLOWANCE AND ISSUE FEE DUE: ISSUE BATCH NUMBER, DATE OF THE NOTICE OF ALLOWANCE, AND SERIAL NUMBER.

#### Attachments:

- ☐ Examiner's Amendment
- ☐ Examiner Interview Summary Record, PTOL-413
- ☒ Reasons for Allowance
- ☐ Notice of References Cited, PTO-892
- ☐ Information Disclosure Citation, PTO-1449

- ☐ Notice of Informal Application, PTO-152
- ☐ Notice re Patent Drawings, PTO-948
- ☐ Listing of Bonded Draftsmen
- ☐ Other