

Filed on behalf of EMC Corporation and VMware, Inc.

By: Peter M. Dichiara, Reg. No. 38,005
David L. Cavanaugh, Reg. No. 36,476
WILMER CUTLER PICKERING HALE AND DORR LLP
peter.dichiara@wilmerhale.com
david.cavanaugh@wilmerhale.com
Tel.: 617-526-6466
Fax: 617-526-5000

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

EMC CORPORATION and VMWARE, INC.,
Petitioners

v.

Patent Owner of
U.S. Patent No. 6,415,280 to Farber et al.

IPR Case No. IPR2013-00083

PETITIONERS' DEMONSTRATIVES FOR ORAL ARGUMENT

Pursuant to 37 C.F.R. § 42.70(b) and the Board's Order (Paper 67),
Petitioners file the enclosed demonstratives for the oral hearing scheduled for
December 16, 2013. Petitioners intend to use any of the exhibits filed in this
proceeding as demonstratives. Petitioners also intend to use as a demonstrative the
physical copy the October 1993 Walnut Creek CD-ROM pictured in Ex. 1041.

Respectfully Submitted,

Dated: December 11, 2013

/Peter M. Dichiara/

Peter M. Dichiara
Registration No. 38,005
Cynthia Vreeland
Admitted *pro hac vice*
WILMER CUTLER PICKERING
HALE AND DORR LLP
60 State Street
Boston, Massachusetts 02109
peter.dichiara@wilmerhale.com
Tel.: 617-526-6466
Fax: 617-526-5000

CERTIFICATE OF SERVICE

I hereby certify that on December 11, 2013, I caused a true and correct copy of the following materials:

- Petitioners' Demonstratives for Oral Argument (Ex. 1083)
- Exhibit List

to be served via email on the following counsel of record for Patent Owner:

Joseph A. Rhoa, Lead Counsel
USPTO Reg. No. 37,515
NIXON & VANDERHYE P.C.
901 North Glebe Road, 11th Floor
Arlington, VA 22203-1808
jar@nixonvan.com
Tel.: 703-816-4043

Updeep S. Gill, Backup Counsel
USPTO Reg. No. 37,344
NIXON & VANDERHYE P.C.
901 North Glebe Road, 11th Floor
Arlington, VA 22203-1808
usg@nixonvan.com
Tel.: 703-816-4030

/Heather M. Petruzzi/
Heather M. Petruzzi
Reg. No. 71,270

Petitioners' Demonstrative Slides

Case IPR2013-00082 – U.S. Patent No. 5,978,791

Case IPR2013-00083 – U.S. Patent No. 6,415,280

Case IPR2013-00084 – U.S. Patent No. 7,945,544

Case IPR2013-00085 – U.S. Patent No. 7,945,539

Case IPR2013-00086 – U.S. Patent No. 7,949,662

Case IPR2013-00087 – U.S. Patent No. 8,001,096

December 16, 2013

Overview

- PersonalWeb Patents
- Woodhill, Langer, Kantor
- Disputed Claim Limitations
 - '791 (Woodhill)
 - '280 (Woodhill)
 - '539 (Langer, Woodhill, Kantor)
 - '096 (Kantor)
 - '662 (Kantor)
 - '544 (Kantor, Woodhill)

PersonalWeb Patents

- All of the patents relate to the use of “substantially unique identifiers” for file management functions
 - Files are divided into one or more data items
 - Data items are identified by “substantially unique identifiers” based on the data in the data item:

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

'791 patent at Col. 3, lines 29-35 (IPR'82, Ex. 1001)

PersonalWeb Patents

- True File registry stores True Name and corresponding information:

FIG. 4

140

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

The True File registry (TFR) 126 is a data store for listing actual data items which have True Names, both files 120 and segments 122. When such data items occur in the True File registry 126 they are known as True Files. True Files are identified in True File registry 126 by their True Names or identities. The table True File registry 126 also stores location, dependency, and migration information about True Files.

'791 patent at Col. 8, lines 28-33 (IPR'82, Ex. 1001)

'791 patent at Fig. 4 (IPR'82, Ex. 1001)

True File ID identity or disk location of the actual physical representation of the file or file segment. It is sufficient to use a filename in the registration directory of the

'791 patent at Col. 9, lines 63-66 (IPR'82, Ex. 1001)

PersonalWeb Patents

- All of the patents use the identifiers to perform basic file management functions, e.g.:
 - determining if a data item is present in the system
 - accessing a data item
 - eliminating unnecessary duplicate copies

PersonalWeb Patents

- According to the patents:

In **all** of the prior data processing systems the names or identifiers provided to identify data items (the data items being files, directories, records in the database, objects in object-oriented programming, locations in memory or on a physical device, or the like) are **always** defined relative to a specific context. For instance, the file identified by a par-

'791 patent at Col. 1, line 65 to Col. 2, line 3 (IPR'82, Ex. 1001)

In prior art systems for identifying data items there is **no** direct relationship between the data names and the data item.

'791 patent at Col. 2, lines 12-13 (IPR'82, Ex. 1001)

PersonalWeb Patents

This invention relates to data processing systems and, more particularly, to data processing systems wherein data items are identified by substantially unique identifiers which:

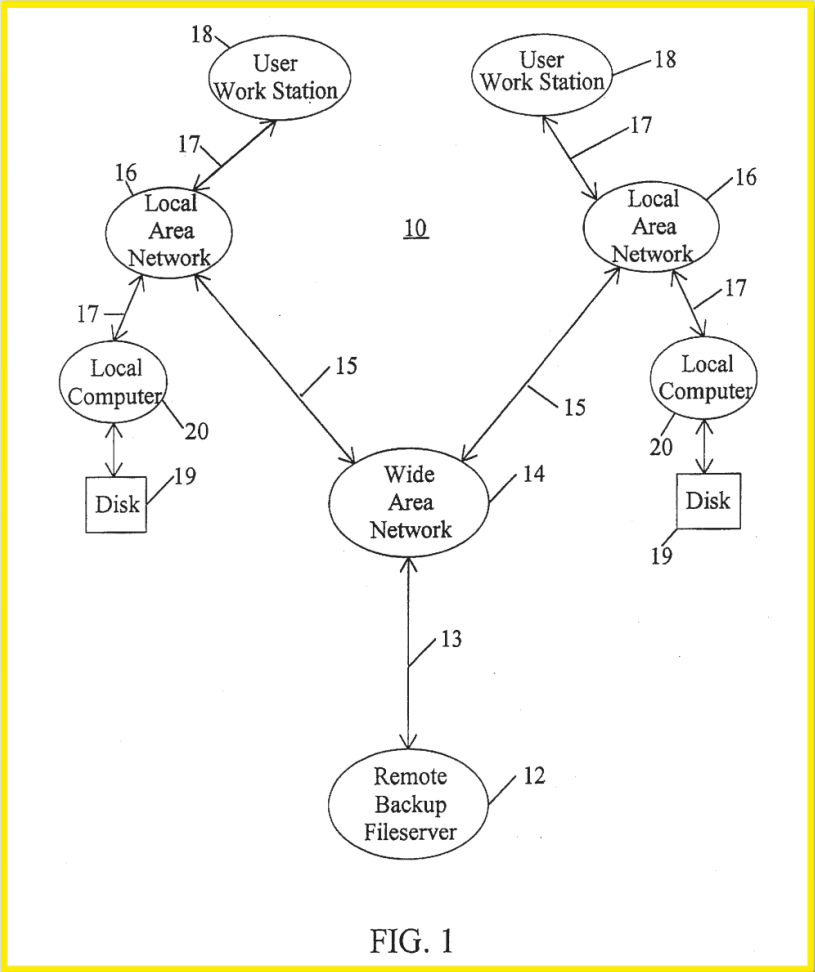
- (A) depend on and
- (B) are determined using:
 - (a) all of the data in the data items and
 - (b) only the data in the data items.

A notable and significant property of this invention is that, in any particular system, two identical data items in the system will have the same identifier.

'791 patent prosecution history, Amendment of August 29, 1997, at page 8 (IPR'82, Ex. 1028)

Overview of Woodhill, Langer and Kantor

Woodhill



Woodhill at Fig. 1 (IPR'82, Ex. 1005)

Woodhill

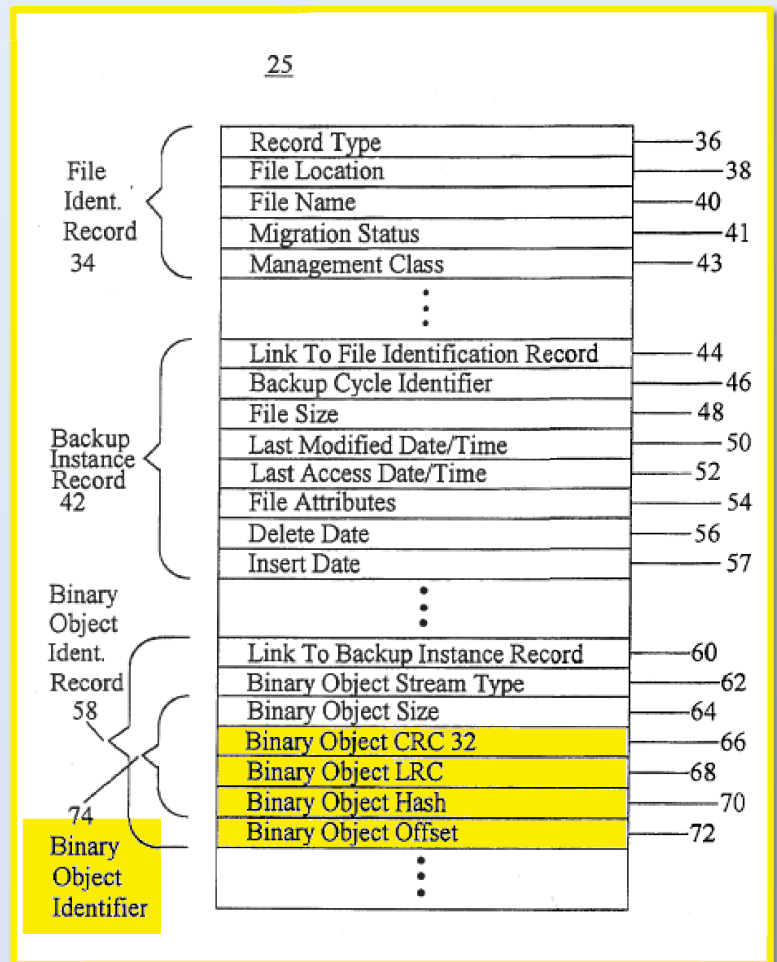
- Files are divided into one or more binary objects (BOBs)
- BOBs are identified by unique binary object identifiers (BOBIDs):

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

Woodhill at Col. 8, lines 58-62 (IPR'82, Ex. 1005)

Woodhill

- Woodhill's File Database stores BOBIDs and corresponding information:



Woodhill at Fig. 3 (IPR'82, Ex. 1005)

Woodhill

- Woodhill uses BOBIDs for basic file management functions, e.g.:
 - Identifying and accessing file data
 - Managing file backups
 - Comparing binary objects to recognize duplicates

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.

Woodhill at Col. 8, lines 62-65 (IPR'82, Ex. 1005)

Langer

http://groups.google.com/group?selm=1991Aug7.225159.786@newshost.anu.edu.au&oe=UTF-8&output=plain 304

From: cmf851@anu.oz.au (Albert Langer)
Newsgroups: alt.sources.d,comp.archives.admin
Subject: Re: dl/describe (File descriptions) posted to alt.sources
Message-ID: <1991Aug7.225159.786@newshost.anu.edu.au>
Date: 7 Aug 91 22:51:59 GMT
References: <1991Aug7.124457.6814@cew.viccol.edu.au>
<1991Aug7.131048.6817@cew.viccol.edu.au>
Sender: news@newshost.anu.edu.au
Followup-To: comp.archives.admin
Organization: Computer Services Centre, Australian National University,
Canberra, Australia.
Lines: 291

In article <1991Aug7.131048.6817@cew.viccol.edu.au>: timco@cew.viccol.edu.au
(Tim Cook) writes:

>I have just posted a new version of "dl/describe" (previously known as

From: cmf851@anu.oz.au (Albert Langer)
Newsgroups: alt.sources.d,comp.archives.admin
Subject: Re: dl/describe (File descriptions) posted to alt.sources
Message-ID: <1991Aug7.225159.786@newshost.anu.edu.au>
Date: 7 Aug 91 22:51:59 GMT

Langer at 1 (IPR'85, Ex. 1003)

permit eventual transition to complete MARC records from the OCLC
project or other forms of more detailed cataloging. Also long
filenames should be handled transparently. A simple display convention
could be that anything starting in the first column is a new filename
(or a continuation of a previous ridiculously long filename that did
not end before the end of the previous line). Anything that starts
with whitespace is a continuation of a description from the previous
line or if at the top, is a description of the directory.
(A similar convention is used in FILES.BBS files that play a similar
role on some BBSes).

>#ifdef MODEST
>
>I would explore all those looking for a system of setting and listing
>descriptive file comments to investigate dl/describe. I think it is a
>very good solution (I especially think the use of DBM files to store

http://groups.google.com/group?selm=1991Aug7...6540@newshost.anu.edu.au&oe=UTF-8&output=plain (1 of 6) (7/29/2003 5:18:17 PM)
EMCMVMW 1003

Langer

- Files are identified by MD5 codes

A simple method of defining a unique identifier that does NOT include a particular site identifier would be to use a hash function on the entire contents of the file. This can be generated locally without requiring a registration system and if long enough the chances of collision are negligible. I would suggest using a cryptographic hash function such as MD5 which generates a 16 byte result. The extra

Langer at 4 (IPR'85, Ex. 1003)

- Packages of files are identified as MD5 hash of hashes

Likewise the code for a tar or cpio or ZIP archive etc or a collection of shar files (with or without uuencoding etc) could be the code obtained by applying MD5 again to the concatenation of the codes of the extracted files, in numeric order. (This deliberately

Langer at 5 (IPR'85, Ex. 1003)

Langer

- Computers can query a central database (Archie) using unique MD5 identifiers to obtain location information

A simple ftp implementation would just hardlink every file available for ftp to a filename encoding of it's MD5 token. Users would then ftp the directory path and filename of the MD5 token and obtain the file. An archie or similar lookup could first determine which nearby systems have the file (though come to think of it, that database lookup may as well also provide the local directory and filename for it). For dial-up sites a mail-server request could be chained until it reached a site with directory access, and the files requested added to temporary

Langer at 4 (IPR'85, Ex. 1003)

and [34d]. For example, Langer discloses that a user computer can access a particular file data item by sending a query to a central database server, such as Archie or WAIS. (Langer at 3-4.) The query includes an MD5 hash for the requested file. (Langer at 3-4.) For example, Langer discloses that a file server

Clark Dec. at ¶ 29 (IPR'85, Ex. 1009)

Langer

- Langer uses MD5 identifiers for basic file management functions

A simple ftp implementation would just hardlink every file available for ftp to a filename encoding of it's MD5 token. Users would then ftp the directory path and filename of the MD5 token and obtain the file. An archie or similar lookup could first determine which nearby systems have the file (though come to think of it, that database lookup may as well also provide the local directory and filename for it). For dial-up sites a mail-server request could be chained until it reached a site with directory access, and the files requested added to temporary

Langer at 4 (IPR'85, Ex. 1003)

- PersonalWeb admits that Langer uses the MD5 identifiers to access files

Langer appears to disclose accessing a standalone file by employing an MD5 of the file contents. (Langer, 4.) However, Langer treats packages *differently*.

Patent Owner Resp. at 41 (IPR'85)

Kantor

FWKCS(TM) Contents_Signature System
Version 1.22
1993 August 10

(C) Copyright F

FWKCS(TM) Contents_Signature System

Version 1.22

1993 August 10

(C) Copyright Frederick W. Kantor 1988-1993. All rights reserved.

EMC 1004

Kantor at Title Page (IPR'85, Ex. 1004)

Kantor

- Files are identified by contents signatures

By constructing a "contents_signature" in which the 32_bit CRC and the uncompressed file_length are both part of a single structure, one can drive the purely statistical contribution of the accidental CRC_matches between files with different lengths all the way to zero

Kantor at 7 (IPR'85, Ex. 1004)

- Zipfiles are made up of inner files and identified by zipfile contents signatures

Zipfile_Contents_Signature ("zcs")
~~~~~

In the case of a zipfile, FWKCS also generates a special "zipfile contents signature" ("zcs"), in which all the files in the zipfile are treated in a special way as a whole. This is done by adding together all the 32\_bit CRC's for the files in the zipfile, modulo  $2^{32}$ , separately adding together their uncompressed file\_lengths modulo  $2^{32}$ , and then arranging the two resulting hexadecimal numbers as a single structure. This has the desirable property that

Kantor at 9 (IPR'85, Ex. 1004)

# Kantor

- Kantor CSLIST stores contents signatures and corresponding information

```

Col.1
|          | |          | |          | |          | |          | |
|          1          |2|          3          |4|          5          |6|          7          ||          8
|          | |          | |          | |          | |          | |
16_character_cs| |cs_owner | |in, or has| |where... ||, zipped_path...|
32bitCRC32bitLen FILENAME.EXT ZIPFILE.EXT [D:] [\PATH] [, INNERPATH]
CRCsum32LenSum32 ZIPFILE.EXT z cs [D:] [\PATH]

32bitCRC32bitLen FILENAME.EXTpZIPFILE.EXT [D:] [\PATH] [, INNERPATH]
CRCsum32LenSum32 ZIPFILE.EXT p z cs [D:] [\PATH]

32bitCRC32bitLen FILENAME.EXTaZIPFILE.EXT v[D:] [\PATH] [, INNERPATH]
CRCsum32LenSum32 ZIPFILE.EXT v z cs [D:] [\PATH]

32bitCRC32bitLen FILENAME.EXTuZIPFILE.EXT v[D:] [\PATH] [, INNERPATH]
CRCsum32LenSum32 ZIPFILE.EXT u z cs [D:] [\PATH]

32bitCRC32bitLen FILENAME.EXTaZIPFILE.EXT v[D:] [\PATH] [, INNERPATH]
32bitCRC32bitLen FILENAME.EXTpZIPFILE.EXT v[D:] [\PATH] [, INNERPATH]
CRCsum32LenSum32 ZIPFILE.EXT u z cs [D:] [\PATH]

32bitCRC32bitLen FILENAME.EXT f cs [D:] [\PATH]

```

(The above examples are grouped to show what happens when the file entry in location 5 appears in location 3. A special\_purpose format for contents\_signature cross\_reference is listed separately. See table of contents.)

Kantor at 52-53 (IPR'85, Ex. 1004)

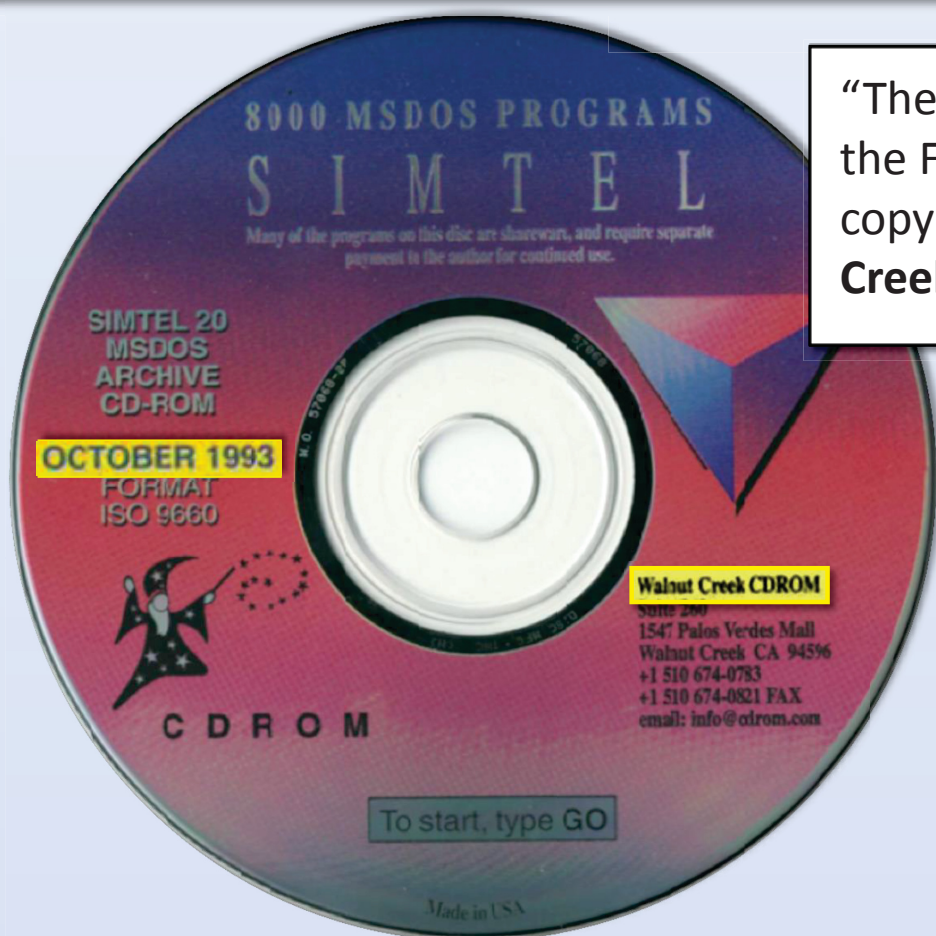
# Kantor

- Kantor uses contents signatures and zipfile contents signatures for basic file management functions, e.g., detecting and eliminating unwanted duplicates

By means of "contents\_signatures", FWKCS automatically compares all the files in a newly received standard zipfile ("standard" means that it is built according to the rules set down by Phil Katz, up to and including a full DOS tree of subdirectories, with no "gimmicks") with all the files in all the standard zipfiles on an entire electronic bulletin board system, working in real time, independent of filenames, file extensions, dates, compression ratios, comments, zipped paths, and order in which the files appear in the zipfile. It recognizes as redundant a zipfile made of pieces scattered in many different zipfiles on the system. It is used to prevent wasteful duplication, and to protect against unwanted files.

Kantor at Preface 2 (IPR'85, Ex. 1004)

# Kantor



“The Kantor Reference is part of the FWKCS122.ZIP file from my copy of the **October 1993 Walnut Creek CD-ROM**. . . .”

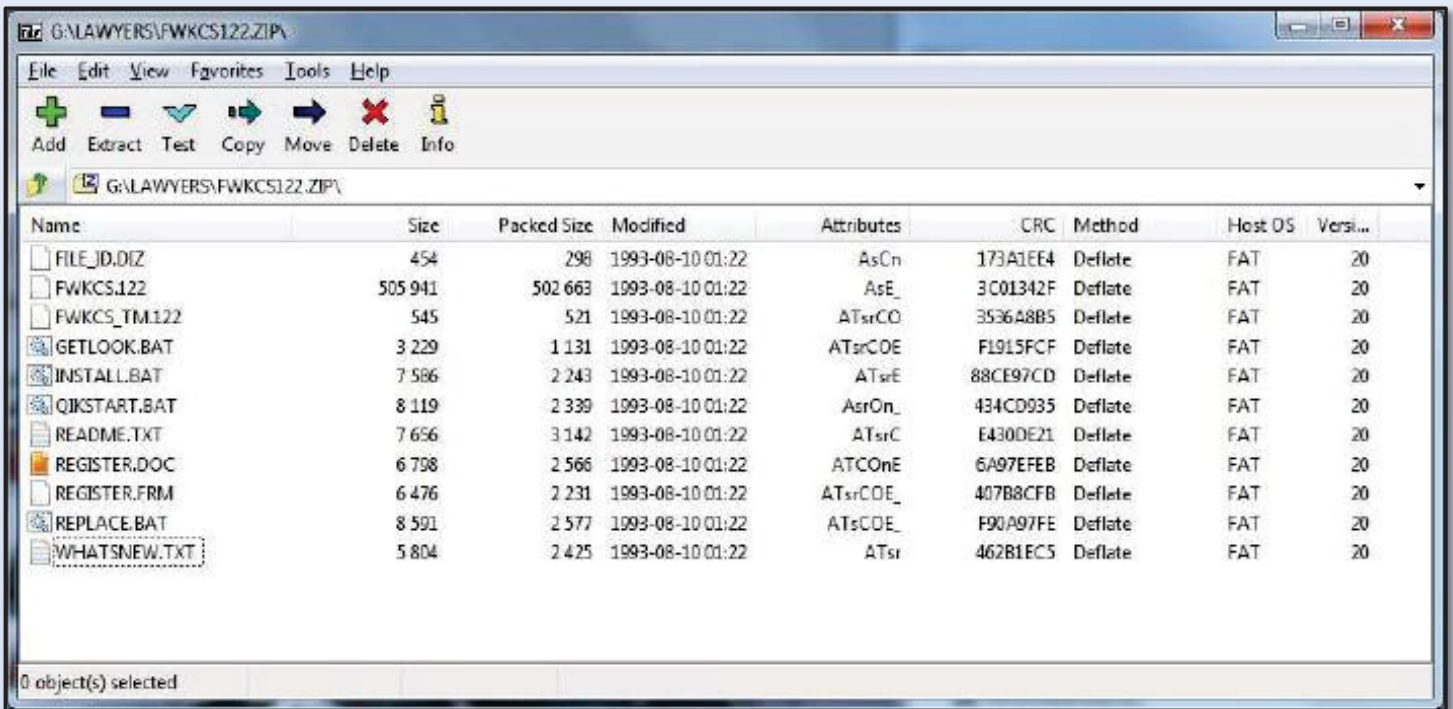
Declaration of Jason Sadofsky,  
Director of *The BBS Documentary*  
at ¶ 16 (IPR’85, Ex. 1081)

October 1993 Walnut Creek CD-ROM (IPR’85, Ex. 1052)



# Kantor

- Mr. Sadofsky's screenshot



Sadofsky Reply Dec. at ¶ 10 (IPR'85, Ex. 1091)

# Kantor

## New Version

-----  
To get a new version of FWKCS, download FWKCSnnn.ZIP from The Invention Factory BBS, where nnn is the new version number without a decimal point. These special downloads are available at no fee, from a 43\_line hunt\_up group of USR Dual Standard modems, at 2400-16800 bits/sec (including V32.bis):

212-274-8110 8N1

Kantor at 158-159 (IPR'85, Ex. 1004)

```
have made a program which accepts the price you want from
the file system you had, subtracts the amount you actually
from the amount you registered price.
To register, use the form REGISTER.FRM provided below.
Registration is by Mark Class P.O. Mail addressed to:
          Frederick W Kantor
          Founder
          Invention Factory
          158 & 159
          New York, NY 10013-1413
*Please note: register in person, and registered users only.
In all you need to use this software in a public, educational
setting, contact me for permission. For commercial use, contact
me for details. Please mark all checks. *The report may be
sent to me but need not go through the U.S. Mail.
*****
Support
The Invention Factory BBS system has been running since
January 1985. It is a 43_line hunt_up group of USR Dual Standard
modems, at 2400-16800 bits/sec (including V32.bis).
*****
Registration
Registration is by Mark Class P.O. Mail addressed to:
          Frederick W Kantor
          Founder
          Invention Factory
          158 & 159
          New York, NY 10013-1413
*****
Support
The Invention Factory BBS system has been running since
January 1985. It is a 43_line hunt_up group of USR Dual Standard
modems, at 2400-16800 bits/sec (including V32.bis).
*****
Registration
Registration is by Mark Class P.O. Mail addressed to:
          Frederick W Kantor
          Founder
          Invention Factory
          158 & 159
          New York, NY 10013-1413
*****
```

```
*****
Registration
Registration is by Mark Class P.O. Mail addressed to:
          Frederick W Kantor
          Founder
          Invention Factory
          158 & 159
          New York, NY 10013-1413
*****
Support
The Invention Factory BBS system has been running since
January 1985. It is a 43_line hunt_up group of USR Dual Standard
modems, at 2400-16800 bits/sec (including V32.bis).
*****
Registration
Registration is by Mark Class P.O. Mail addressed to:
          Frederick W Kantor
          Founder
          Invention Factory
          158 & 159
          New York, NY 10013-1413
*****
Support
The Invention Factory BBS system has been running since
January 1985. It is a 43_line hunt_up group of USR Dual Standard
modems, at 2400-16800 bits/sec (including V32.bis).
*****
Registration
Registration is by Mark Class P.O. Mail addressed to:
          Frederick W Kantor
          Founder
          Invention Factory
          158 & 159
          New York, NY 10013-1413
*****
```

# **Disputed Claim Limitations**

**'791 – Woodhill (IPR'82)**

# '791 Patent

- Challenged claims
  - MPF Claims 1-4, 29
  - Method claims 30-33, 41
- Claims focus on substantially unique identifiers to perform basic file management functions such as determine if a data item is present in the system or to access the data item
- Woodhill is primary reference

## Woodhill: “Identity means” (claim 1)

- PersonalWeb does not dispute Woodhill satisfies identity means as construed by PTAB
  - “Identity means . . . whereby two identical data items in the system will have the same identifier” (’791 patent at claim 1 (IPR’82, Ex. 1001))
- Instead, PersonalWeb contends PTAB’s construction is wrong
- But PersonalWeb’s expert has “no opinion” whether PersonalWeb’s construction is correct. (Dewar Dec. at ¶ 32 (IPR’82, Ex. 2013))
- PersonalWeb’s expert admits “all hash functions” have the property that “identical data items will have the same identifier” (Dewar Tr. At 60 (IPR’82, Ex. 1078))

# Woodhill: “Existence means for determining whether a particular data item is present in system” (claim 1)

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.

Woodhill at Col. 8, lines 62-65 (IPR'82, Ex. 1005)

The Binary Object Identifiers 74 calculated in step 138 are compared against their counterparts in the File Database 25

Woodhill at Col. 9, lines 14-16 (IPR'82, Ex. 1005)

The True Name of a file can be used to identify a file by contents, to confirm that a file matches its original contents, or to compare two files.

'791 Patent at Col. 15, lines 25-27 (IPR'82, Ex. 1001)

## Woodhill: “Determining whether a particular data item is present at a particular location” (claims 2,3)

Q. And the system knows that the modified first BOB exists on the local computer?

A. Yes.

Q. I mean, that's a simple case, there is no doubt about that?

A. Yes.

Q. It definitely exists there, right?

A. Right.

Q. And therefore, since it exists on the local computer, it exists in the system?

A. Yes.

Q. There is no doubt about that?

A. There is no doubt about that.

Dewar Tr. at 124-25 (IPR'82, Ex. 1078)



## Woodhill: “Determining whether a particular data item is present in the system” (claim 1)/ “at a particular location” (claims 2,3)

Q. And when Woodhill is processing that BOB, the comparison is going to determine that the BOBIDs are the same, right?

A. Right.

Q. And it is going to determine that the remote already has that BOB because the BOBIDs are equal, right?

A. Right.

Q. In this situation, the system knows that the second BOB exists on the local computer, right?

A. Right.

Q. And there is no doubt about that because that's where it is, right?

A. Right.

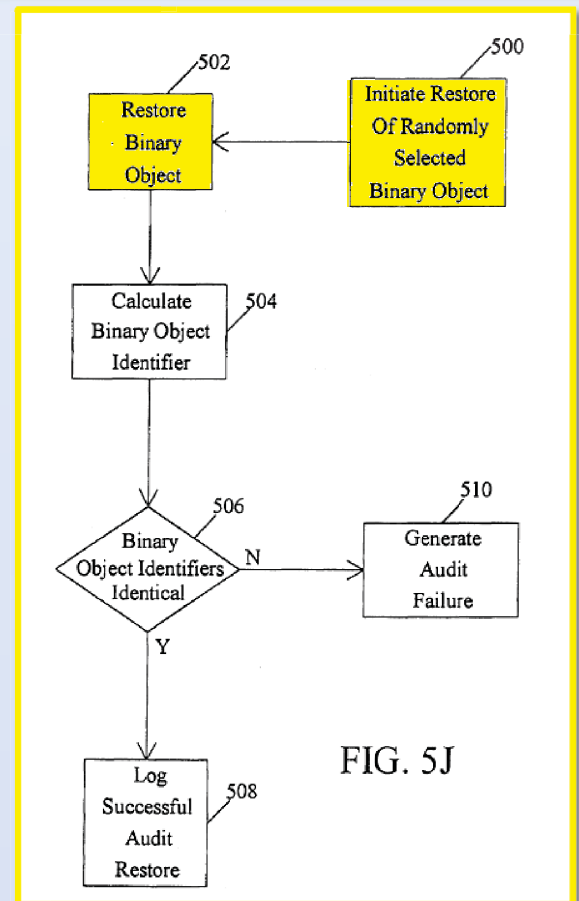
Q. And it also knows that that second BOB exists in the backup file server?

A. Right.

# Woodhill: “Access means” (claim 4)/ “accessing” a data item (claim 30)

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.

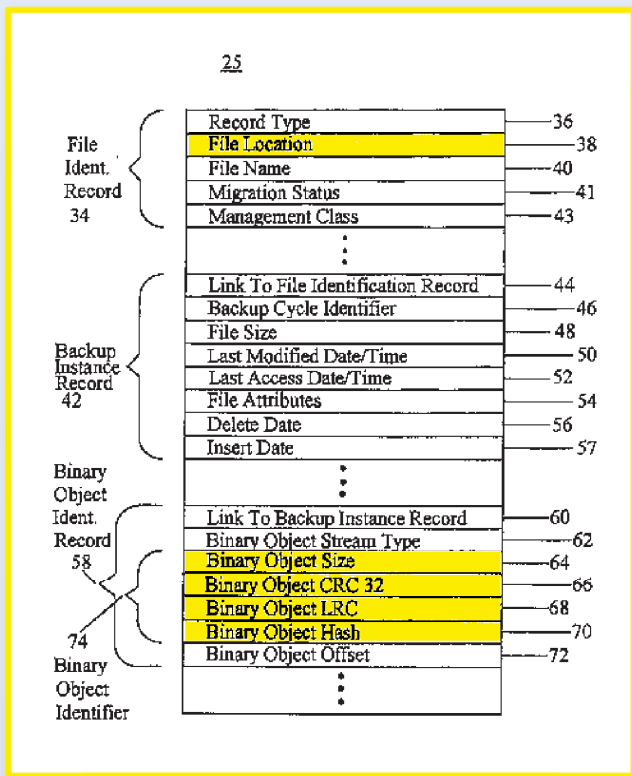
Woodhill at Col. 18, lines 17-19 (IPR'82, Ex. 1005)



Woodhill at Fig. 5J (IPR'82, Ex. 1005)

# Woodhill: File database (claims 1-4)/ “data associating means” (claim 4)

- Woodhill database:



Woodhill at Fig. 3 (IPR'82, Ex. 1005)

- '791 True File Registry:

shown in the True File registry record 140 in FIG. 4. The True File registry 126 consists of the database described in the table below as well as the actual True Files identified by the True File IDs below.

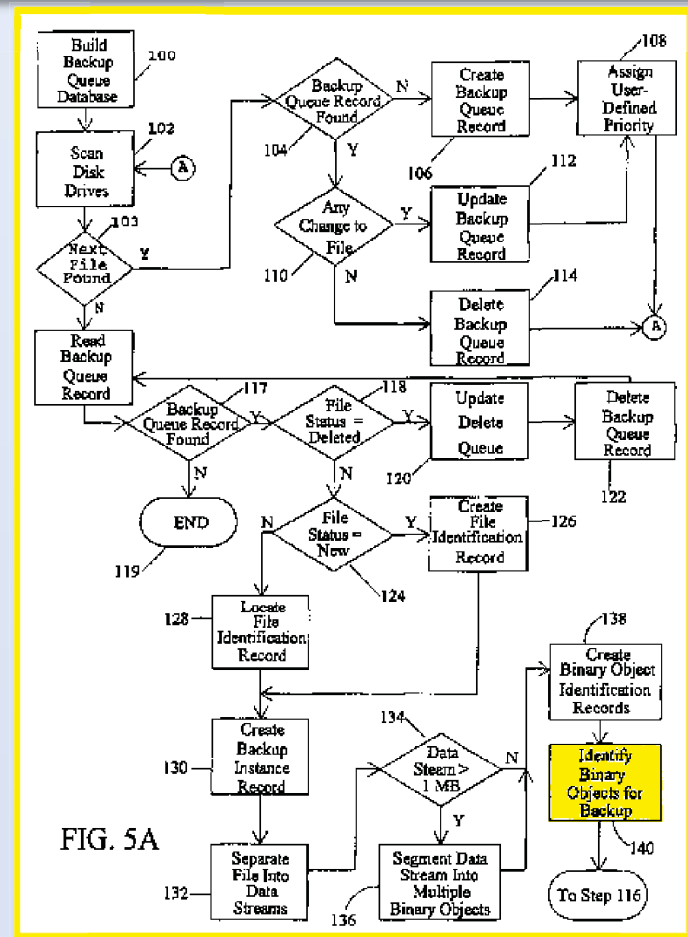
| Field        | Description                                                                                                                                                                                                                                                                                       |
|--------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| True Name    | computed True Name or identity of the file.                                                                                                                                                                                                                                                       |
|              | * * *                                                                                                                                                                                                                                                                                             |
| True File ID | identity or disk location of the actual physical representation of the file or file segment. It is sufficient to use a filename in the registration directory of the underlying operation system. The True File ID is absent if the actual file is not currently present at the current location. |

'791 patent at Col. 9, lines 37-67 (IPR'82, Ex. 1001)  
(portion omitted)

# Woodhill: “Providing” (claim 33)/ “fetching” data (claim 41)

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.

Woodhill at Col 9, lines 23-26 (IPR'82, Ex. 1005)



Woodhill at Fig. 5A (IPR'82, Ex. 1005)

## Woodhill: “Providing” (claim 33)/ “fetching” data (claim 41)

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

Woodhill at Col. 10, lines 32-34 (IPR’82, Ex. 1005)

- Q. There is no dispute that even by the time of Woodhill there was technology around that would determine that on the local computer, if your disk was fried or your file was corrupted, that it would say I can't get this file?
- A. Right.
- Q. And in that case, you would, using the Woodhill technology, you could get that file from the backup file server?
- A. Right.

Dewar Tr. at 139-40 (IPR’82, Ex. 1078)

**'280 – Woodhill (IPR'83)**

## '280 Patent

- Challenged claims
  - Method claims 36, 38
- Claims focus on using content-based identifiers to store, request and obtain a copy of a data file from a set of servers
- Primary Reference is Woodhill

# Woodhill: “Responsive to a client request for the data file ... causing the data file to be provided” (claims 36, 38)

| The '280 Patent Claims                                                                                                                                                     | U.S. Pat. No. 5,649,196 to Woodhill et al.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                                                                                                                                            | skill in the art exercising ordinary creativity.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| [36c] responsive to a client request for the data file, the request including a hash of the contents of the data file, causing the data file to be provided to the client. | <p>Woodhill discloses, responsive to a client request for the data file, the request including a hash of the contents of the data file, causing the data file to be provided to the client. For example, the data processing system of Woodhill allows restores (client request) of binary objects (data files) using their Binary Object Identifiers during the Backup/Restore Routine. Also, the data processing system of Woodhill performs periodic self-audits by initiating a restore of a randomly selected binary object, identified by its Binary Object Identification Record, which includes its Binary Object Identifier. A Binary Object Identifier includes a hash value calculated by a hash function of the contents of the binary object that it identifies. For example:</p> <p>“Since most restores of files on a local area network 16 consist of requests to restore the most recent backup version of a file, 30 the local copies of binary objects serve to handle very fast restores for most restore requests that occur on the local area network 16.” (Woodhill at col. 10, lines 27 – 32).</p> |

**to the client. For example, the data processing system of Woodhill allows restores (client request) of binary objects (data files) using their Binary Object Identifiers during the Backup/Restore Routine. Also, the data processing system of Woodhill**

\* \* \*

“Since most restores of files on a local area network 16 consist of requests to restore the most recent backup version of a file, 30 the local copies of binary objects serve to handle very fast restores for most restore requests that occur on the local area network 16.” (Woodhill at col. 10, lines 27 – 32).

Woodhill claim chart at 4 (IPR’83, Ex. 1032)





# Woodhill: “Responsive to a client request for the data file ... causing the data file to be provided” (claims 36, 38)

binary object identified by a Binary Object Identification Record 58.” A Binary Object Identification Record includes a “Binary Object Identifier 74 portion ... 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.” (Woodhill at col. 7, line 60 – col. 8, line 65; Ex. 1005.) In particular, the Binary Object Hash field 70 “is calculated against the contents of the binary object.” (Id.; Ex. 1005.) The Binary Object Hash field 70 is thus a value determined as a given function of the contents of the file. A person of ordinary skill in the art would understand that a local computer initiating a restore of a binary object is a client request.

27. Indeed, a person of ordinary skill in the art would find many similarities between Woodhill and the ‘280 patent. Furthermore, both Woodhill and the ‘280 patent are directed to file backup applications. The ‘280 patent notes that “[b]ackup of data items in a DP system employing the present invention can be done based on the True Names of the data items. By tracking backups using True Names, duplication in the backups is prevented.” (‘280 patent at col. 36, ll. 57-59; Ex. 1001.) Woodhill is similarly directed to a distributed backup system, and reduces duplicate copies of binary objects by tracking Binary Object Identifiers of previously backed up binary objects. (Woodhill at col. 1, ll. 11-17 and col. 8, ll. 66 – col. 9, ll. 27; Ex. 1005.) As well, both Woodhill and the ‘280 patent perform self-checks to ensure that backed up data remains secure and

-17-

unchanged. The ‘280 patent describes that “the system might store the True Names of all executable applications on the system and then periodically redetermine the True Names of each of these applications to ensure that they match the stored True Names.” (‘280 patent at col. 34, ll. 50-54; Ex. 1001.) Like the ‘280 patent, Woodhill “is able to perform self-audits on a periodic basis to ensure that the binary objects that have been backed up can be restored.” (Woodhill at col. 18, ll. 11-13; Ex. 1005.) Woodhill does so by restoring a randomly selected binary object and ensuring that a new Binary Object Identifier calculated for that restored binary object matches the Binary Object Identifier that was previously stored. (Id. at col. 18, ll. 13-30; Ex. 1005.)

Ground 4. Challenged Claims 36 and 38 are Unpatentable as Obvious in view of Woodhill.

28. As discussed above, Woodhill discloses a distributed storage management system for storing multiple backup copies of files and ultimately restoring those files. (Woodhill at col. 2, ll. 39-49; Ex. 1005.) A person of ordinary skill in the art at the time of the filing of the ‘280 patent would understand a distributed storage management system to disclose multiple servers.

29. Furthermore, to the extent multiple servers are not already disclosed in Woodhill, it would be well within the routine creativity of a person of ordinary skill in the art to add an additional remote backup file server, or servers, to the system disclosed in Woodhill. Indeed, Woodhill provides an express motivation

-18-

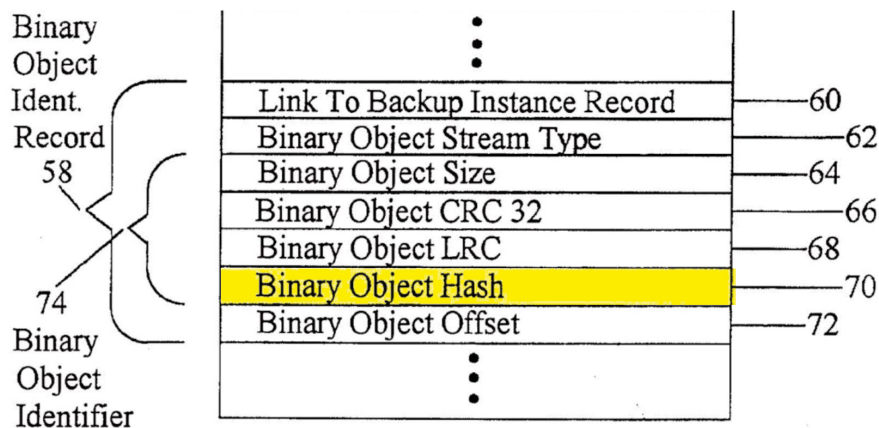
and col. 8, l. 66 – col. 9, l. 27; Ex. 1005.) As well, both Woodhill and the ‘280 patent perform self-checks to ensure that backed up data remains secure and unchanged. The ‘280 patent describes that “the system might store the True Names of all executable applications on the system and then periodically redetermine the True Names of each of these applications to ensure that they match the stored True Names.” (‘280 patent at col. 34, ll. 50-54; Ex. 1001.) Like the ‘280 patent, Woodhill “is able to perform self-audits on a periodic basis to ensure that the binary objects that have been backed up can be restored.” (Woodhill at col. 18, ll. 11-13; Ex. 1005.) Woodhill does so by restoring a randomly selected

Clark Dec. at ¶ 27 (IPR’83, Ex. 1009)

# Woodhill: “Responsive to a client request for the data file ... causing the data file to be provided” (claims 36, 38)

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.

Woodhill at Col 18, lines 17-19 (IPR'83, Ex. 1005)



Woodhill at Fig. 3 (IPR'83, Ex. 1005) (portion omitted)

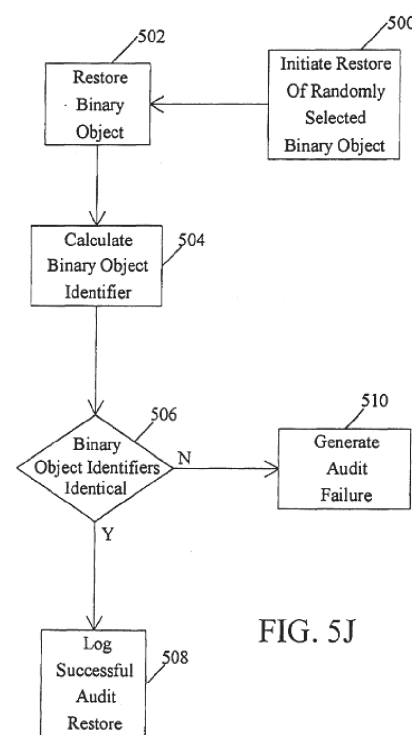


FIG. 5J

Woodhill at Fig. 5J (IPR'83, Ex. 1005)

**'539 – Langer, Woodhill,  
and Kantor (IPR'85)**

## '539 Patent

- Challenged claims
  - Method claims 10, 21, 34
- Claims focus on using substantially unique identifiers, based on the contents of the segments contained within a data item, to access the data item
- Three primary references:
  - Langer
  - Woodhill
  - Kantor

**Langer As Primary Reference**

## Langer: Claims 10, 21, 34

Langer discloses that *a user computer can access a particular file data item by sending a query to a central database server, such as Archie or WAIS. (Langer at 3–4.) The query includes an MD5 hash for the requested file. (Langer at 3–4.) For*

*token.” (Langer at 4.) In response, the central database server uses the MD5 hash to return the locations, such as FTP sites, that store a copy of the file corresponding to the identifier. (Langer at 4; Ex. 1003) (“An archie or similar lookup could*

Decision at 21-22 (quoting Clark Dec. at ¶ 29 (Ex. 1009)) (IPR’85)

## Langer: Claims 10, 21, 34

provide the local directory and filename for it).”) *In order to retrieve a particular file in an archive, a person of ordinary skill in the art would understand that the MD5 hash of the archive could be used to obtain the MD5 hashes that were computed for each of the files within the archive. The user computer can then access the particular file using the FTP protocol. (Langer at 4.) (“Users would then ftp the directory*

Decision at 21-22 (quoting Clark Dec. at ¶ 29 (Ex. 1009)) (IPR’85)



## Langer: “Obtaining ... segment identifiers” (claims 10, 21)

## “Providing at least said second data item” (claim 34)

### PACKAGES CONTAINING A DIRECTORY OR DIRECTORY TREE

A related problem is that essentially the same collection of information may be available as different .tar.Z or zoo or ZIP or shar files etc. This happens especially with files distributed through sources newsgroups and archived with different methods (or even with the same methods, but including the local headers, which are different). It will also happen where a local modification has been added to a package.

Ultimately these do have to be regarded as DIFFERENT files and any connections between them listed separately. Nevertheless a user may be wondering whether to ftp a package that has a new MD5 code to see if it contains new revisions and it would be nice to be able to tell the user without the need for collecting the entire package.

Langer at 5 (IPR'85, Ex. 1003)

**Langer: “Obtaining ... segment identifiers” (claims 10, 21)  
“Providing at least said second data item” (claim 34)**

Q. And then he continues, "...and it would be nice to be able to tell the user without the need for collecting the entire package," right?

A. Right.

Q. And so what he is saying is, he's trying to propose a way so that the user can find out what the new revisions are without having to get the whole package over?

A. Right.

Q. And there is no dispute about that?

A. No.

Dewar Tr. at 355-56 (IPR'85, Ex. 1088)

## Langer & Woodhill: “Dividing a ... data item” (claim 34)

31. As I discuss below, in ¶¶ 44-55, Woodhill discloses a technique for dividing data items into segments and an added length value to its identifier. A person of ordinary skill in the art, exercising ordinary creativity, would have been motivated to modify Langer’s system to use a segmentation technique and added length value, like the ones disclosed in Woodhill. Such combination of Langer

with Woodhill would be obvious to a person of ordinary skill in the art. The known device of Langer is a system of more efficiently handling data by reducing the probability

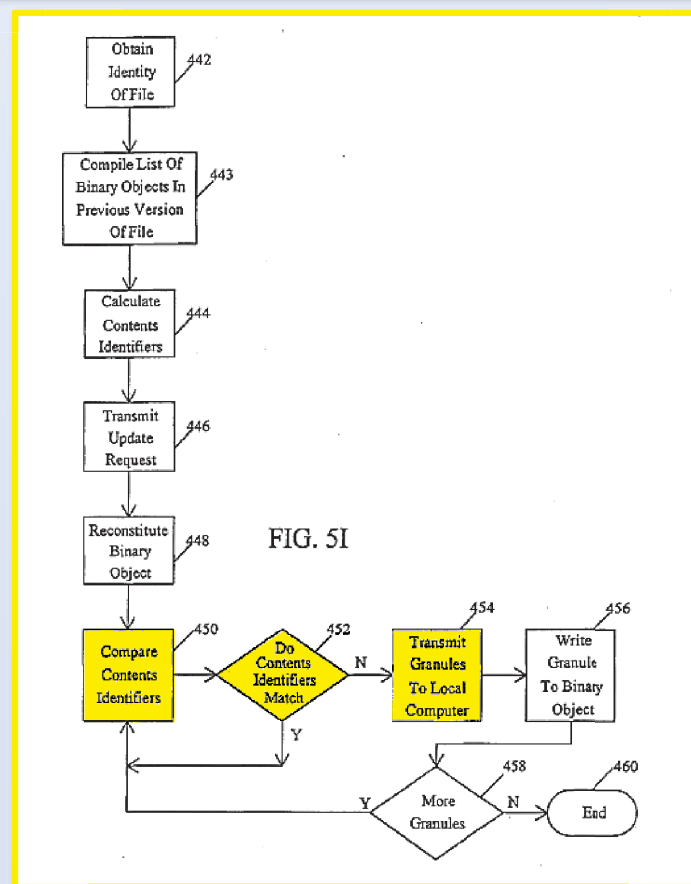
of data loss (Ex. 1005, 1:1-10; 2:1-10; 3:1-10; 4:1-10; 5:1-10; 6:1-10; 7:1-10; 8:1-10; 9:1-10; 10:1-10; 11:1-10; 12:1-10; 13:1-10; 14:1-10; 15:1-10; 16:1-10; 17:1-10; 18:1-10; 19:1-10; 20:1-10; 21:1-10; 22:1-10; 23:1-10; 24:1-10; 25:1-10; 26:1-10; 27:1-10; 28:1-10; 29:1-10; 30:1-10; 31:1-10; 32:1-10; 33:1-10; 34:1-10; 35:1-10; 36:1-10; 37:1-10; 38:1-10; 39:1-10; 40:1-10; 41:1-10; 42:1-10; 43:1-10; 44:1-10; 45:1-10; 46:1-10; 47:1-10; 48:1-10; 49:1-10; 50:1-10; 51:1-10; 52:1-10; 53:1-10; 54:1-10; 55:1-10; 56:1-10; 57:1-10; 58:1-10; 59:1-10; 60:1-10; 61:1-10; 62:1-10; 63:1-10; 64:1-10; 65:1-10; 66:1-10; 67:1-10; 68:1-10; 69:1-10; 70:1-10; 71:1-10; 72:1-10; 73:1-10; 74:1-10; 75:1-10; 76:1-10; 77:1-10; 78:1-10; 79:1-10; 80:1-10; 81:1-10; 82:1-10; 83:1-10; 84:1-10; 85:1-10; 86:1-10; 87:1-10; 88:1-10; 89:1-10; 90:1-10; 91:1-10; 92:1-10; 93:1-10; 94:1-10; 95:1-10; 96:1-10; 97:1-10; 98:1-10; 99:1-10; 100:1-10). (Pet. 41.) Indeed, Woodhill describes that if a data stream is larger than the maximum size (*e.g.*, 1 megabyte), the Distributed Storage Manager (“DSM”) program divides the data stream into multiple binary objects (Ex. 1005, 4:19-30), and for large files (*e.g.*, database files), the DSM program divides the large file into “granules” and creates a “shadow file” which contains the content identifiers for the “granules” (Ex. 1005, 14:62-65; 15:9-24). Dr. Clark testifies that a person

Clark Dec. at ¶ 19  
(IPR’85, Ex. 1009)

Decision at 24 (IPR’85)

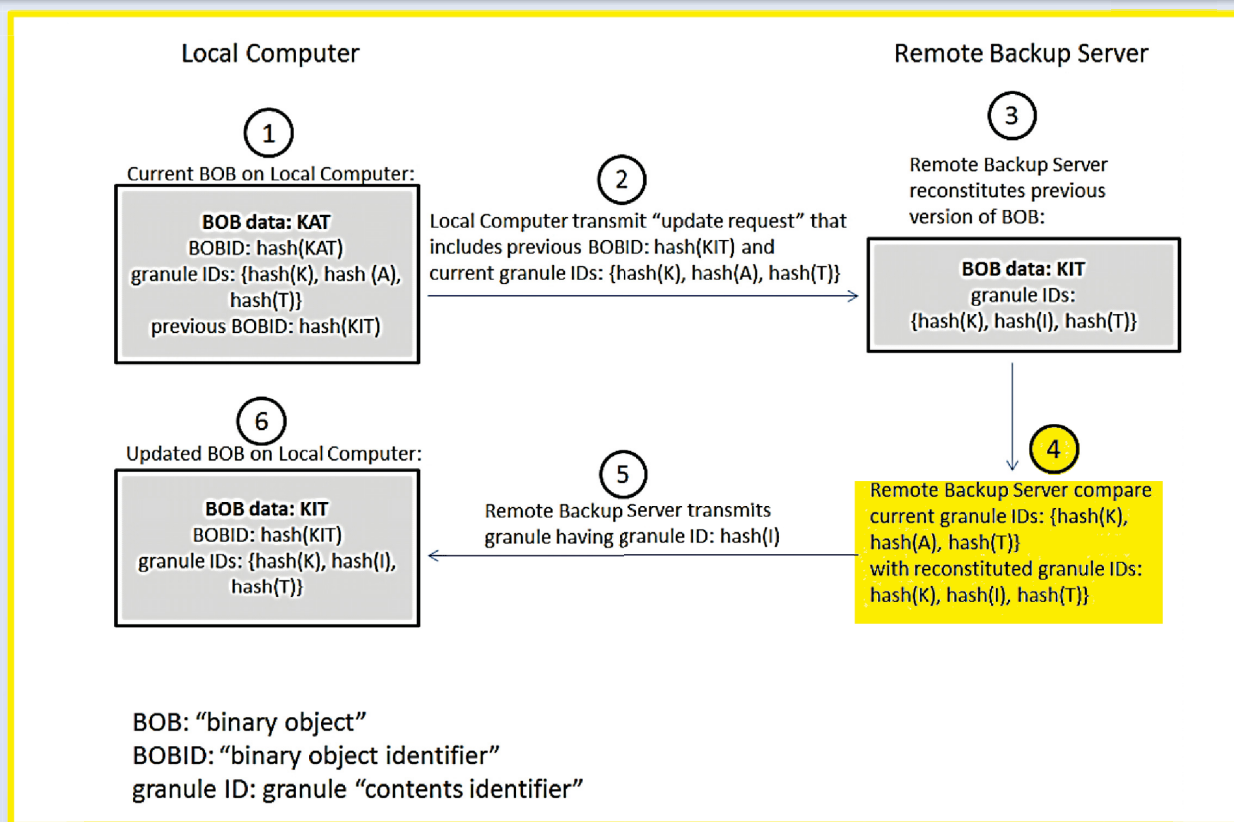
# **Woodhill As Primary Reference**

**Woodhill/Fischer: “Using at least one of said segment identifiers ... requesting at least one particular segment” (claims 10, 21)**



Woodhill at Fig. 5I  
(IPR'85, Ex. 1005)

# Woodhill/Fischer: “Using a first data identifier to obtain a plurality of segment identifiers” (claim 21)/ “using at least one of said segment identifiers ... requesting at least one particular segment” (claims 10,21)



Clark Dec. at ¶ 50 (IPR’85, Ex. 1009)

**Woodhill/Fischer: “Using at least one of said segment identifiers ... requesting at least one particular segment”  
(claims 10, 21)**

Q. When the remote backup file server notices that these identifiers don't match, it knows that the local computer is missing the corresponding granule as it exists in the work area?

A. Right.

Q. And then it transmits it in step 454?

A. Right.

Q. And in order to make that transmission, it has to request that that granule get transmitted from the remote back to the local?

A. Right.

Dewar Tr. at 197 (IPR'85, Ex. 1088)

## Woodhill/Fischer: “Said first identifier is based, at least in part, on a second given function of the plurality of segment identifiers” (claims 10, 21)

57. Fischer is a representative example of the use of a hash of hashes.

The use of a hash of hashes was well-known to a person of ordinary skill in the art at the time of the filing of the '539 patent, as further demonstrated by its use in

Browne, Langer, and Kantor. For example, Fischer discloses building an

aggregate “fileHash” value for a database file containing multiple records by

hashing the data of each record and then combining those values using a functions

such as “exclusive OR” or “arithmetic addition.” (Fischer at col. 7, line 66 – col.

8, line 31; Ex. 1036.) Both of these functions are th

Thus, by combining hashes of the data of each reco

Fischer hashes the hashes of the records to create th

hashes”).

Clark Dec. at ¶ 57  
(IPR'85, Ex. 1009)

### • Fischer disclosure:

**In accordance with the exemplary embodiment, a hash of the database File is computed as follows. The file hash is initially set equal to an initial value (such as 0). Thereafter, the hash routine indexes over all records in the database File using all the record identifiers  $K_i$  of the records  $R_i$ , 1 to N in the database file, computing this value:**

$$\text{fileHash} = F(\text{fileHash}, H(K_i \& R_i))$$

Fischer at Col. 7, line 65 to Col. 8, line 5 (IPR'85, Ex. 1036)



# **Kantor As Primary Reference**

## Kantor: “Data item comprising a plurality of segments” (claims 10, 21)

By means of "contents\_signatures", FWKCS automatically compares all the files in a newly received standard zipfile ("standard" means that it is built according to the rules set down by Phil Katz, up to and including a full DOS tree of subdirectories, with no "gimmicks") with all the files in all the standard zipfiles on an entire electronic bulletin board system, working in real time, independent of filenames, file

Kantor at Preface 2 (IPR'85, Ex. 1004)

**Kantor: “The segment identifier ... being based, at least in part, on a first given function of the data” (claims 10, 21)/ “determining ... segment identifiers” (34)**

By constructing a "contents\_signature" in which the 32\_bit CRC and the uncompressed file\_length are both part of a single structure, one can drive the purely statistical contribution of the accidental CRC\_matches between files with different lengths all the way to zero

Kantor at 7 (IPR'85, Ex. 1004)

f - make cs for file(s) in zip(s).

Make a "File contents signature" for (each) File in zipfile(s).

This looks inside the zipfile named on the command line (wildcards \* ? are permitted), or inside each file in the control list (wildcards can be used inside the control list); tests to see if the file is a zipfile (including both storage zipfiles and self-extracting zipfiles); and then uses information stored in the ZIP format to make a contents\_signature for each of the files in the zipfile; the zipfile

Kantor at 48 (IPR'85, Ex. 1004)

## Kantor: Obvious to modify read/download commands to identify files with contents signatures (claims 10, 21, 34)

41. Kantor renders obvious all portions of claims 10, 21 and 34. As

discussed above, Kantor discloses using the file contents-signature to identify a file by its contents. A person of ordinary skill in the art would find it obvious to modify the read and/or download command to the BBS used with the FWKCS system disclosed in Kantor to provide a more efficient and context-free means for accessing and sharing files, and would have been obvious to a person of ordinary skill in the art.

filename. I Indeed, Kantor provides an express motivation to do so. For example, Kantor discloses the Lookup operation, which is a request containing a file or zipfile contents-signature to determine where the corresponding file or zipfile is located on the BBS. (Kantor at 96-97; Ex. 1004).

For example, Kantor discloses the Lookup operation, which is a request containing a file or zipfile contents-signature to determine where the corresponding file or zipfile is located on the BBS.

(Kantor at 96-97; Ex. 1004). In response to a read and/or download command to the BBS used with the FWKCS system disclosed in Kantor, the BBS would obtain the predictable result of providing a more efficient and context-free means for accessing and sharing files, and would have been obvious to a person of ordinary skill in the art.

file contents-signature identifies. Furthermore, adding a contents-signature based (e.g. using a contents-signature or “zcs”) read and/or download command to the BBS used with the FWKCS system disclosed in Kantor would obtain the predictable result of providing a more efficient and context-free means for accessing and sharing files, and would have been obvious to a person of ordinary skill in the art.

Clark Dec. at ¶ 41 (IPR’85, Ex. 1009)

## Kantor: Obvious to modify read/download commands to identify files with contents signatures (claims 10, 21, 34)

i - process remote Inquiries requesting a contents\_signature search. With option i, a person can ask ahead to find out if material which he/she is thinking of uploading is already on a BBS.

Option i also provides potentially valuable cross\_search capabilities. For example, a user could use this to find other files on a BBS which contain material related to a file he/she has, by learning in which zipfiles the specific file appears, and then use the y form of the TEST function to obtain full sets of contents\_signatures for all the files in each of those zipfiles, and so on.

This remote\_Inquiries option helps the BBS sysop and the BBS users work together as a team.

Kantor at 96 (IPR'85, Ex. 1004)

## Kantor and Langer: “Dividing a particular data item into a plurality of segments” (claim 34)

```
ER/* ACCESZIP.BAT for accessioning zip or plain files , uses  
Phil Katz's PKUNZIP(R) to unzip zipfiles,  
can perform a variety of tests, and can  
rename files which do not pass the tests.  
This uses the options set in FWKCS.EXE  
macro [b].
```

Kantor at 174 (IPR'85, Ex. 1004)

## Kantor and Langer: “Determining a data item identifier for said data item...” (claim 34)

Zipfile\_Contents\_Signature ("zcs")

~~~~~

In the case of a zipfile, FWKCS also generates a special "zipfile contents signature" ("zcs"), in which all the files in the zipfile are treated in a special way as a whole. This is done by adding together all the 32_bit CRC's for the files in the zipfile, modulo 2^{32} , separately adding together their uncompressed file_lengths modulo 2^{32} , and then arranging the two resulting hexadecimal numbers as a single structure. This has the desirable property that

Kantor at 9 (IPR'85, Ex. 1004)

Kantor and Langer: “Determining a data item identifier for said data item...” (claim 34)

SUMMARY OF THE INVENTION

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

'539 patent at Col. 3, lines 55-58 (IPR'85, Ex. 1001)

structure, FWKCS has the special ability to make a "zipfile contents signature", ("zcs") which is independent of zipfile comment, file comments, zipfile name, zipfile date, order in which files are included in the zipfile, the names and dates of files in the zipfile, zipped path information, and file compression ratio. By comparing their zcs's, FWKCS can find,

Kantor at Preface 2 (IPR'85, Ex. 1004)

'096 – Kantor (IPR'87)

'096 Patent

- Challenged claims
 - Method claims 1, 2, 81 and 83
- Claims focus on replicating data items and using content-based identifiers to access data
- Kantor is the primary reference
 - Satyanarayanan II is the secondary reference for data replication

Kantor: “A method operable in a file system” (claims 1, 2, 81, 83)

which are sent to the users. FWKCS is in use on electronic bulletin board systems with multiple servers and up to more than 80 nodes, and runs also on a single personal computer for personal use. It is also used for handling a mix including

Kantor at Preface 2 (IPR'87, Ex. 1004)

Kantor: “Data item consisting of a sequence of non-overlapping parts” (claims 1, 2, 81, 83)

By means of "contents_signatures", FWKCS automatically compares all the files in a newly received standard zipfile ("standard" means that it is built according to the rules set down by Phil Katz, up to and including a full DOS tree of subdirectories, with no "gimmicks") with all the files in all the standard zipfiles on an entire electronic bulletin board system, working in real time, independent of filenames, file

Kantor at Preface 2 (IPR'87, Ex. 1004)

Kantor: “Digital part identifiers ... determined based at least in part on a first function comprising a hash function” (claims 1, 2, 81, 83)

By constructing a "contents_signature" in which the 32_bit CRC and the uncompressed file_length are both part of a single structure, one can drive the purely statistical contribution of the accidental CRC_matches between files with different lengths all the way to zero

Kantor at 7 (IPR'87, Ex. 1004)

Kantor: “A digital data item identifier ... based ... on the contents of the data item” (claims 1, 2, 81, 83)

```
Zipfile_Contents_Signature ("zcs")
```

```
~~~~~
```

In the case of a zipfile, FWKCS also generates a special "zipfile contents signature" ("zcs"), in which all the files in the zipfile are treated in a special way as a whole. This is done by adding together all the 32_bit CRC's for the files in the zipfile, modulo 2^{32} , separately adding together their uncompressed file_lengths modulo 2^{32} , and then arranging the two resulting hexadecimal numbers as a single structure. This has the desirable property that

Kantor at 9 (IPR'87, Ex. 1004)

Kantor: “A digital data item identifier ... based ... on the contents of the data item” (claims 1, 2, 81, 83)

SUMMARY OF THE INVENTION

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

'096 patent at Col. 3, lines 55-58 (IPR'87, Ex. 1001)

structure, FWKCS has the special ability to make a "zipfile contents signature", ("zcs") which is independent of zipfile comment, file comments, zipfile name, zipfile date, order in which files are included in the zipfile, the names and dates of files in the zipfile, zipped path information, and file compression ratio. By comparing their zcs's, FWKCS can find,

Kantor at Preface 2 (IPR'87, Ex. 1004)

Kantor: Obvious to modify read/download commands to identify files with contents signatures (claims 1, 2, 81, 83)

83. A person of ordinary skill in the art would have found it obvious to modify the BBS commands, including the download and/or read commands, so the commands would accept contents-signatures or zipfile contents-signatures to identify the files or zipfiles on which to operate. One of the benefits would be facilitate

integrity checking because a user could specify the file of interest based on its content instead of an arbitrary filename, and thus improve accuracy. Kantor shows that such a modification would be easy to implement. For example, FWKCS used contents-signatures as input for certain user commands, such as the “Lookup” feature (see *id.* at 97 and 173; Ex. 1004.), and it would have been straightforward to allow download and read commands to identify a file by a contents-signature in a similar way.

83. A person of ordinary skill in the art would have found it obvious to modify the BBS commands, including the download and/or read commands, so the commands would accept contents-signatures or zipfile contents-signatures to identify the files or zipfiles on which to operate. One of the benefits would be facilitate integrity checking because a user could specify the file of interest based on its content instead of an arbitrary filename, and thus improve accuracy. Kantor shows that such a modification would be easy to implement. For example, FWKCS used contents-signatures as input for certain user commands, such as the “Lookup” feature (*see id.* at 97 and 173; Ex. 1004.), and it would have been straightforward to allow download and read commands to identify a file by a contents-signature in a similar way.

Clark Dec. at ¶ 83 (IPR’87, Ex. 1009)

Kantor: Obvious to modify read/download commands to identify files with contents signatures (claims 1, 2, 81, 83)

i - process remote Inquiries requesting a contents_signature search. With option i, a person can ask ahead to find out if material which he/she is thinking of uploading is already on a BBS.

Option i also provides potentially valuable cross_search capabilities. For example, a user could use this to find other files on a BBS which contain material related to a file he/she has, by learning in which zipfiles the specific file appears, and then use the y form of the TEST function to obtain full sets of contents_signatures for all the files in each of those zipfiles, and so on.

This remote_Inquiries option helps the BBS sysop and the BBS users work together as a team.

Kantor at 96 (IPR'87, Ex. 1004)

Kantor And Satya II Are An Obvious Combination (claims 1, 2, 81, 83)

IEEE TRANSACTIONS ON COMPUTERS, VOL. 39, NO. 4, APRIL 1990

447

Coda: A Highly Available File System for a Distributed Workstation Environment

MAHADEV SATYANARAYANAN, MEMBER, IEEE, JAMES J. KISTLER, PUNEET KUMAR,
MARIA E. OKASAKI, ELLEN H. SIEGEL, AND DAVID C. STEERE

III. SERVER REPLICATION

The unit of replication in Coda is a *volume*, a set of files and directories located on one server and forming a partial subtree of the shared name space.⁴ Each file and directory in Coda has a unique low-level file *identifier* (FID), a component of which identifies the parent volume. All replicas of an object have the same FID.

The set of servers with replicas of a volume constitute its *volume storage group* (VSG). The degree of replication and the identity of the replication sites are specified when a volume is created and are stored in a *volume replication database* that is present at every server. Although these parameters can

Satya II at 450 (IPR'87, Ex. 1028)

'662 – Kantor (IPR'86)

'662 Patent

- Challenged claim
 - Method claim 30
- Claims focus on the concept of using content-based identifiers to delete data items
- Kantor is the primary reference
 - Satyanarayanan II is the secondary reference for data replication

Kantor: “Obtaining a particular digital data item identifier ... being obtained in response to an attempt to delete said particular data item...” (claim 30)

1) MULTIS

014FF56D	158AC	LAWN2.ZIP	z	CS	J:\RBBS
014FF56D	158ACd	LAWN200.ZIP	z	CS	M:\RBBS
0249E9AB	486D	EQIPVIEW.ZIP	z	CS	H:\RBBS
0249E9AB	486Dd	VIS.ZIP	z	CS	K:\RBBS
02FF6937	522E7d	CHEMIC20.ZIP	z	CS	E:\RBBS
02FF6937	522E7	CHEMICAL.ZIP	z	CS	E:\RBBS

Kantor at 189 (IPR'86, Ex. 1004)

Kantor: “Obtaining a particular digital data item identifier ... being obtained in response to an attempt to delete said particular data item...” (claim 30)

2) Exclude

excludes (zip)file if cs_match with x_flag found in column 17 of CSLIST.SRT or CSLIST1.SRT, for any contents_signature which matches the contents_signature of a non_zip file, or which matches the contents_signature of any file in a zipfile.

To mark material for exclusion from your BBS, you can use EXCLUDE, which is provided in FWKCS122.ZIP (see /a7.3, below).

re marking material for exclusion, see EXCLUDE /*<enter>.

Kantor at 81 (IPR'86, Ex. 1004)

Kantor: “Obtaining a particular digital data item identifier ... being obtained in response to an attempt to delete said particular data item...” (claim 30)

Q. We walked through at least two examples, both the duplicate and the **exclude**, where it is automatically deleted, right?

A. Yes, it seems to suggest in some cases there is an automatic deletion.

Dewar Tr. At 287-89 (IPR'86, Ex. 1079)

Kantor: "Updating a record in said list to reflect deletion of said particular data item" (claim 30)

MULTIS list

```
-----  
014FF56D 158AC LAWN2.ZIP z cs J:\RBBS  
014FF56D 158ACdLAWN200.ZIP z cs M:\RBBS  
-----  
0249E9AB 486D EQIPVIEW.ZIP z cs H:\RBBS  
0249E9AB 486DdVIS.ZIP z cs K:\RBBS  
-----  
02FF6937 522E7dCHEMIC20.ZIP z cs E:\RBBS  
02FF6937 522E7 CHEMICAL.ZIP z cs E:\RBBS
```

Kantor at 189
(IPR'86, Ex. 1004)

MULTIS deleted.log

Then, at solely your own risk, you could run

```
basica fwkcl7d (or use a different BASIC with your system)
```

to delete all of those marked files, and to save those lines (without the "d") in a file named DELETED.LOG. For example, here are the corresponding 3 lines copied from the beginning of the IFBB DELETED.LOG (see discussion, below):

```
014FF56D 158AC LAWN200.ZIP z cs M:\RBBS  
0249E9AB 486D VIS.ZIP z cs K:\RBBS  
02FF6937 522E7 CHEMIC20.ZIP z cs E:\RBBS
```

Kantor at 190
(IPR'86, Ex. 1004)

Kantor: “Updating a record in said list to reflect deletion of said particular data item” (claim 30)

● Exclude upload log

Here is a partial example of what the log may look like:

date	time	comment	new/total	file length (decimal)	filename.ext	ContentSignature type
				* * *		
19921018	0953:36.28	Duplicat	0/	2	272 DUPTEST.ZIP	z97085119 4E
19921018	0953:56.62	Accessnd	3/	4	510 1NOTNEWZ.ZIP	zDF06C9D4 5B
				* * *		
19921018	0958:45.09	Excluded	0/	1	29 XCPTEST.TXT	f8EFF2BA4 1D

What is done with the file in the various cases depends on options you have set up; option "s" saves rejected files, otherwise rejects are typically either **deleted** or sequestered.

Date and time are in the format YearMoDy HrMn:Sc.nn

which is

Year Month Day Hours(24) Minutes : Seconds.Hundredths

This can easily be sorted in **chronological order**. If you have more than one uploads log, you can simply copy them together into one file and sort the combined file to put them in order.

Kantor at 101 (IPR'86, Ex. 1004)

Kantor: “Updating a record in said list to reflect deletion of said particular data item” (claim 30)

Q. So the upload log file is going to give you both the type of operation, right, accessioned or **excluded**, right?

A. Right.

Q. The file name, the content signature, the date it happened, date and time it happened, correct?

A. Yes, that's what I read, yeah.

Dewar Tr. At 287-89 (IPR'86, Ex. 1079)

Kantor and Satya II Are An Obvious Combination (claim 30)

IEEE TRANSACTIONS ON COMPUTERS, VOL. 39, NO. 4, APRIL 1990

447

Coda: A Highly Available File System for a Distributed Workstation Environment

MAHADEV SATYANARAYANAN, MEMBER, IEEE, JAMES J. KISTLER, PUNEET KUMAR,
MARIA E. OKASAKI, ELLEN H. SIEGEL, AND DAVID C. STEERE

III. SERVER REPLICATION

The unit of replication in Coda is a *volume*, a set of files and directories located on one server and forming a partial subtree of the shared name space.⁴ Each file and directory in Coda has a unique low-level file *identifier* (FID), a component of which identifies the parent volume. All replicas of an object have the same FID.

The set of servers with replicas of a volume constitute its *volume storage group* (VSG). The degree of replication and the identity of the replication sites are specified when a volume is created and are stored in a *volume replication database* that is present at every server. Although these parameters can

Satya II at 450 (IPR'86, Ex. 1026)

Kantor and Satya II Are An Obvious Combination (claim 30)

47. As discussed above, Kantor discloses commands for marking and recording the deletion of zipfiles and regular, non-zip files. Kantor further discloses that FWKCS is designed to run on a BBS multiple servers, but Kantor did not specifically disclose that copies of a given file would be stored on multiple servers. However, the Coda system discloses this technique. (See Satyanarayanan II at 447 and 450; Ex 1026.) A person of ordinary skill in the art at the time of the filing of the '662 patent, exercising ordinary creativity, would have found it obvious to combine Kantor with the Coda system so as to benefit from the

reliability

delete file

modified

FWKCS c

would ha

Coda syst

with the C

result of p

would ha

II at 447 and 450; Ex 1026.) A person of ordinary skill in the art at the time of the filing of the '662 patent, exercising ordinary creativity, would have found it obvious to combine Kantor with the Coda system so as to benefit from the reliability of the Coda replicated file system while still being able to identify and delete files using Kantor's contents-signatures. Subsequently, FWKCS as

Grounds of Invalidity for Challenged Claim 30 based on Woodhill as a Primary Reference

Clark Dec. at ¶ 47 (IPR'86, Ex. 1009)

'544 – Kantor, Woodhill (IPR'84)

'544 Patent

- Challenged claim 1
- Claim focuses on comparing two data items comprising parts based on their content-based identifiers.
- Primary references
 - Kantor
 - Woodhill

Kantor As Primary Reference

Kantor and Woodhill: “Data item comprising a plurality of parts” (claim 1)

By means of "contents_signatures", FWKCS automatically compares all the files in a newly received standard zipfile ("standard" means that it is built according to the rules set down by Phil Katz, up to and including a full DOS tree of subdirectories, with no "gimmicks") with all the files in all the standard zipfiles on an entire electronic bulletin board system, working in real time, independent of filenames, file

Kantor at Preface 2 (IPR'84, Ex. 1004)

Kantor and Woodhill: “Applying a first function to each part of said plurality of parts to obtain a corresponding part value” (claim 1)

By constructing a "contents_signature" in which the 32_bit CRC and the uncompressed file_length are both part of a single structure, one can drive the purely statistical contribution of the accidental CRC_matches between files with different lengths all the way to zero

Kantor at 7 (IPR'84, Ex. 1004)

f - make cs for file(s) in zip(s).

Make a "File contents signature" for (each) File in zipfile(s).

This looks inside the zipfile named on the command line (wildcards * ? are permitted), or inside each file in the control list (wildcards can be used inside the control list); tests to see if the file is a zipfile (including both storage zipfiles and self-extracting zipfiles); and then uses information stored in the ZIP format to make a contents signature for each of the files in the zipfile; the zipfile

Kantor at 48 (IPR'84, Ex. 1004)

Kantor and Woodhill: Obvious To Combine (claim 1)

26. It would be well within routine creativity of a person of ordinary skill in the art to combine the teachings of Kantor and Woodhill. Both are concerned with uniquely identifying files for reliable file access (Kantor at 10-11; Ex. 1004 and Woodhill at col. 9, ll. 30-44; Ex. 1005.) Although the zipfiles and files in Kantor are clearly themselves “data items” and “parts of data items,” a person of ordinary skill in the art would have found it obvious to modify Kantor to include even smaller “data items” and “parts of data items” (i.e., smaller sequences of bits), like those disclosed in Woodhill. It is a well-

known technique useful, for example, in dividing a file into parts. In particular, Woodhill discloses “binary objects,” and further discloses “granules”. (Woodhill at col. 15, ll. 4-8; Ex. 1005.) Woodhill would have found it obvious to modify Kantor to include even smaller “data items” and “parts of data items” (i.e., smaller sequences of bits), like those disclosed in Woodhill. It is a well-known technique to divide a file into parts. This can be done by dividing a file into parts, which are then transmitted over a network (Woodhill at col. 15, ll 4-8; Ex. 1005.)

themselves “data items” and “parts of data items,” a person of ordinary skill in the art would have found it obvious to modify Kantor to include even smaller “data items” and “parts of data items” (i.e., smaller sequences of bits), like those disclosed in Woodhill. It is a well-known technique to divide a file into parts. This can be

Clark Dec. at ¶ 26 (IPR’84, Ex. 1009)

Woodhill As Primary Reference

Woodhill: “Obtaining a first value for the first data item, said first value obtained by applying a second function to the part values” (claim 1)

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

Woodhill at Col. 5, lines 61-63 (IPR’84, Ex. 1005)

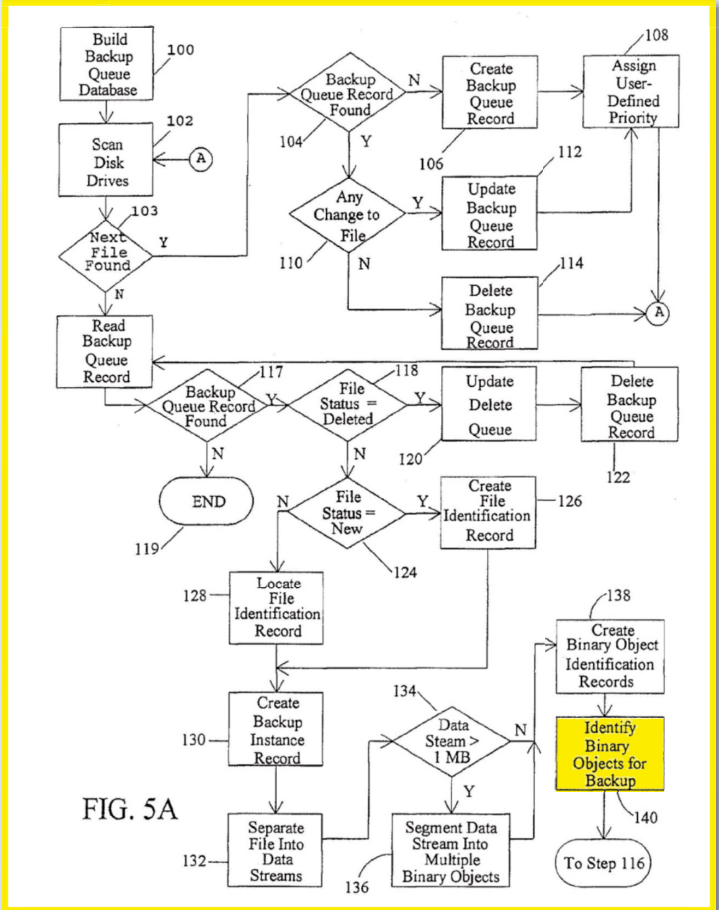
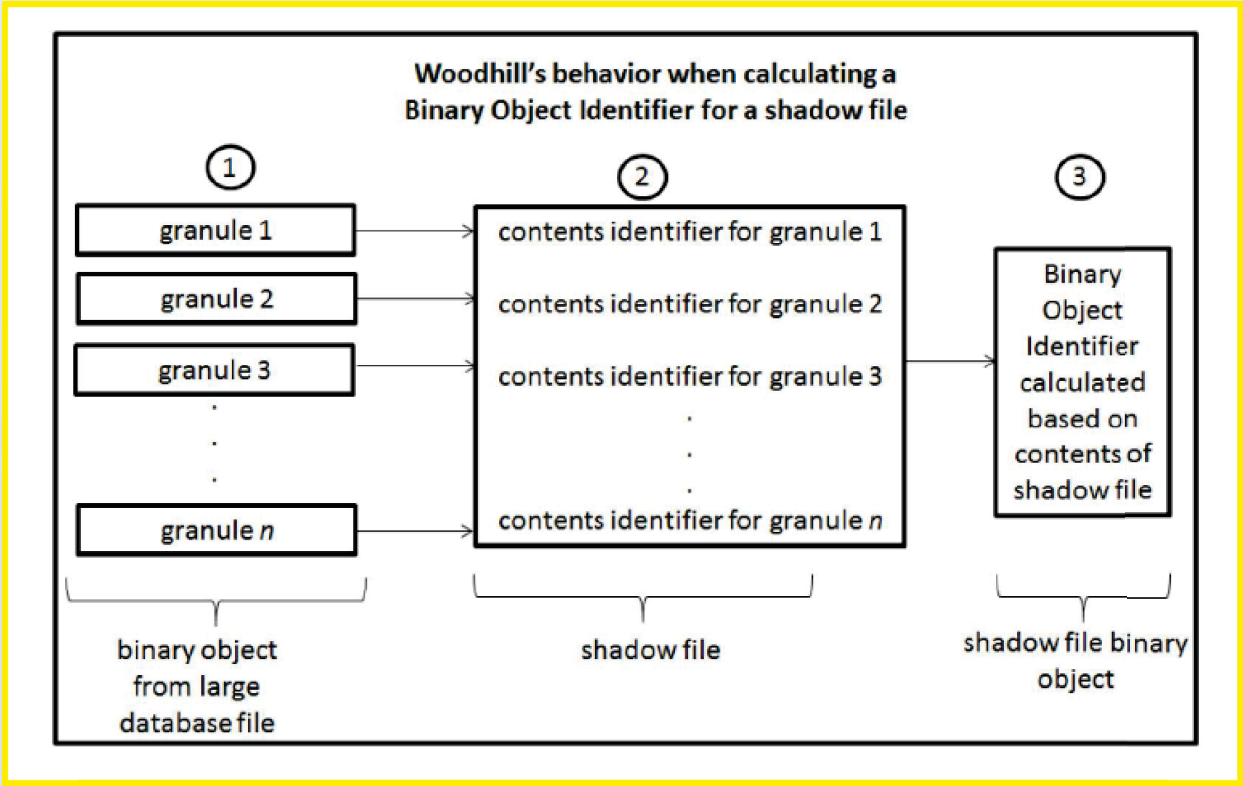


FIG. 5A

Woodhill at Fig. 5A (IPR’84, Ex. 1005)

**Woodhill: “Obtaining a first value for the first data item, said first value obtained by applying a second function to the part values”
(claim 1)**

- Backing up a shadow file:



Clark Reply Dec. at ¶ 17 (IPR'84, Ex. 1088)

Summary

- '791 claims unpatentable over Woodhill
- '280 claims unpatentable over Woodhill
- '539 claims unpatentable over Langer, Kantor, Woodhill
- '096 claims unpatentable over Kantor
- '662 claims unpatentable over Kantor
- '544 claims unpatentable over Kantor, Woodhill