Title: OBJECT ORIENTED FAULT TOLERANCE

Abstract: An electronic storage system (100), such as a file system, may include a storage access routine (108) to store data objects. Data objects (112) stored in the system (100) have an entry in an object index that may be maintained by the storage access routine (108). The object index includes fault tolerance metadata for each data object (112) that specifies the fault tolerance techniques used in storing a data object (112), such as a particular RAID level. The fault tolerance metadata maintained

[Continued on next page]
for each object (112) allows different data objects (112) to be stored with different fault tolerance techniques on the same storage volume. Additionally, the metadata for a data object (112) may be modified in order to store the object (112) using different fault tolerance techniques.
Object Oriented Fault Tolerance

Background

The invention relates generally to file systems for computer systems, and more particularly, but not by way of limitation, to the application of fault tolerance features to the storage of computer system objects.

As information technology progresses, there is a growing need for more efficient and more reliable data storage. Many companies in the marketplace today employ computer systems (including computer networks) which utilize vast amounts of data storage space. Additionally, as corporations begin to rely increasingly on electronic data storage for record keeping, accounting, document storage, and other important business operations, data storage must be dependable.

Conventional computer systems may include fault tolerant mechanisms such as redundant arrays of independent disks (RAID) technology to compensate for hardware failures and/or to increase storage performance. RAID systems typically include several disk units that are used in conjunction to provide fault tolerant data storage at the volume level. A storage volume may be a specified partition within a single disk unit, an entire disk unit, or a collection of disk units. Various levels of RAID technology may be employed to improve performance and/or to maintain data integrity of a storage volume. For example, RAID Level 0 involves striping data objects across multiple disks. Data striping involves storing blocks of data over a series of disk units at the volume level, such that all blocks on a storage volume are divided into a series of bytes that are distributed across the series of disk units. Although data striping may allow parallel disk operations that improve performance, no redundancy is provided. RAID Level 1 allows for complete redundancy by utilizing a mirroring technique at the volume level. Data mirroring involves storing a duplicate copy of all data on a storage volume on a separate volume (the "mirror"). Although mirroring techniques typically require twice as much storage space to accommodate the mirrored data, they provide complete redundancy of a storage volume. With RAID Level 1, an array of disks is divided into two groups such that one group stores the original data while a second group stores the mirrored data. RAID Level 3
includes data striping at the volume level with an additional disk dedicated to parity data. Parity techniques involve the generation and storage of extra information bits that may be used to reconstruct lost data in the event of a storage failure. A parity block may be generated that produces information (the information constitutes the parity block) which allows the original data on a single disk to be recreated from the parity block and data on other disks. Referring to Table 1, an example of RAID Level 3 is shown such that data objects A-E are striped across disk units 1-4, by storing data blocks (e.g., A1-A4) of the data object (e.g., data object A) on different disk units. The location of a data block within an individual disk unit is arbitrary, and thus a data block may be stored at any location on a particular disk unit as shown in Table 1. All of the parity blocks P1-P5 for data objects A-E are stored on the dedicated parity disk unit 5. Each parity block may be constructed from data blocks in a parity stripe, illustrated as a row in Table 1, using any parity technique such as an exclusive OR operation (+). For example, parity block P4 may be the result of a series of exclusive OR operations with data blocks A3, E4, A1, D1 as operands such that P4 = A3 + E4 + A1 + D1. In addition, Parity blocks are computed from data blocks without regard to the particular object to which the data block belongs.

Table 1. RAID Level 3 Example

<table>
<thead>
<tr>
<th>Disk Units</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>C3</td>
<td>A4</td>
<td>B4</td>
<td>B3</td>
<td>P1</td>
</tr>
<tr>
<td></td>
<td>E2</td>
<td>B1</td>
<td>D3</td>
<td>A2</td>
<td>P2</td>
</tr>
<tr>
<td></td>
<td>D4</td>
<td>C2</td>
<td>C4</td>
<td>E3</td>
<td>P3</td>
</tr>
<tr>
<td></td>
<td>A3</td>
<td>E4</td>
<td>A1</td>
<td>D1</td>
<td>P4</td>
</tr>
<tr>
<td></td>
<td>B2</td>
<td>D2</td>
<td>E1</td>
<td>C1</td>
<td>P5</td>
</tr>
</tbody>
</table>

Another technique, RAID Level 5, involves striping of both data and parity blocks at the volume level. Data and parity striping are combined to provide increased reliability (due to the use of parity) and possibly improved access performance (due to the ability to access multiple portions of a single stored object
in parallel because it is distributed across multiple disk units). RAID Level 5 is
demonstrated in Table 2 with data objects A-E and parity blocks P1-P5 striped across
disk units 1-5. Because RAID Level 5 does not prescribe the pattern in which parity
blocks are stored, parity blocks P1-P5 may be interleaved with the series of data
blocks or stored elsewhere on the disk units 1-5.

<table>
<thead>
<tr>
<th>Disk Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
</tr>
<tr>
<td>C3</td>
</tr>
<tr>
<td>E2</td>
</tr>
<tr>
<td>D4</td>
</tr>
<tr>
<td>P4</td>
</tr>
<tr>
<td>B2</td>
</tr>
</tbody>
</table>

Current fault tolerance techniques such as those described above are
implemented at the storage volume level, requiring the same level of fault tolerance
to be applied to all data stored in a storage volume, regardless of the nature of the
data. Storing data in this manner is inflexible and does not allow different fault
tolerant techniques to be applied to different data objects on a single storage
volume. Also, reconstructing data of an object from a failed disk unit may require
retrieving data from unrelated objects which otherwise would not need to be
retrieved. Thus, it would be beneficial to provide a more flexible fault tolerant
computer storage system.

Summary

In one embodiment the invention provides a method to store data objects in
a storage medium. The method includes receiving an identifier associated with a
data object and associating a fault tolerance indication with the identifier. The
method also includes storing the data object in the storage medium in accordance
with the fault tolerance indication. In another embodiment of the invention, a
program storage device that is readable by a programmable control device includes
instructions stored on the program storage device. The instructions cause the
programmable control device to receive an identifier associated with a data object
and associate a fault tolerance indication with the identifier. The instructions also
cause the programmable control device to store the data object in a storage medium
in accordance with the fault tolerance indication. In yet another embodiment, the
invention provides a memory for storing data for access by a storage retrieval
program. The memory includes a data structure having a plurality of entries for use
by the storage retrieval program. The entries in the data structure include an object
identifier to identify a stored object and a fault tolerance identifier to identify a level
of fault tolerance applied to the stored object.

**Brief Description of the Drawings**

Figure 1 shows an illustrative computer system with a storage access routine
in accordance with one embodiment of the invention.

Figure 2 shows an illustrative object index in accordance with the invention.

Figure 3 shows a flow diagram for storing a data object with fault tolerance
metadata in accordance with the invention.

**Detailed Description**

Techniques (including methods and devices) to provide electronic data
storage are described. The following embodiments of the invention, described in
terms of RAID-type fault tolerant techniques at the storage object level, are
illustrative only and are not to be considered limiting in any respect.

Referring to Fig. 1, an illustrative computer system 100 in accordance with
one embodiment of the invention may include a processor unit 102 including a
programmable control device 104 and a storage controller 106. The programmable
control device 104 may be a single microprocessor or microcontroller, a plurality of
processors coupled by a communications link, or a custom designed state machine.
The storage controller 106 may include a storage access routine 108 to both store
and retrieve data from a mass storage device 110. The mass storage device 110 may be a single magnetic or optical disk unit or may include one or more storage units in a redundant array of independent disks (RAID). A RAID device may allow for several different fault tolerance techniques, such as those previously described, to be employed in storing and retrieving data.

The storage access routine 108 may store data objects 112 on the storage media 110. The term "object" may be construed as any data that may be stored and retrieved by a memory label, such as a filename or handle. Illustrative data objects may include any type of data that a file system may store including file objects or directory objects. A file object may be a sequential array of records, such as a text file or a binary file. A directory object may be an ordered collection of records. Members of the ordered collection may be file objects or other directory objects that form subdirectories. Additionally, a data object may be a volume of data storage or even an entire file system.

Data objects 112 may be created and manipulated in the system 100 by instructions comprising the storage access routine 108. The storage access routine 108 may include instructions to create, read, write, and delete data objects. Additionally, the storage access routine 108 may maintain an object index of currently stored data objects. When a data object is created, the storage access routine 108 may include an object handle for the new data object in the object index. The object handle may be used by the storage access routine 108 to reference the data object when retrieving or modifying the data object.

The storage access routine 108 may also maintain metadata for each data object as a part of the object index. Metadata for a data object may be any type of data that describes particular characteristics of the data object, such as file type or access permissions, or indicates storage locations for portions of the data object. Metadata is usually generated by the storage access routine 108, and in some embodiments, metadata may be generated according to user input. In addition to conventional metadata that may be associated with a data object, the storage access routine 108 may maintain fault tolerance metadata for each data object. Fault tolerance metadata for a data object may include information regarding a RAID level
or other fault tolerance technique to be used in storing the data object as well as indications of storage locations for fault tolerance data such as parity disk blocks or mirror disk blocks. For example, a user may create a file object and may choose data mirroring for the file object. The storage access routine may generate and assign fault tolerance metadata to the file object designating data mirroring as the fault tolerance technique for the object and providing indications of storage locations for mirror disk blocks. The user may create a second file object and choose data striping for the second file object (e.g., RAID level 3). Data striping may be indicated in the fault tolerance metadata for the second file object. Therefore, a different fault tolerance technique may be defined for each object stored in the system 100.

Referring to FIG. 2, an illustrative object index 200 includes one entry for each data object 202, 204, and 206 stored in the system 100. Each data object entry includes a handle 208 to reference the data object. A data object entry also includes conventional metadata 210 such as file type or access permissions. Additionally, a data object entry includes fault tolerance metadata 212 to specify the stored object's level of fault tolerance. The object index 200 maintains separate fault tolerance metadata 212 for each individual data object 202, 204, and 206. Also included in an object's metadata 210 may be indications of storage locations for various blocks of the data object. The fault tolerance metadata 212 may include indications of storage locations for fault tolerance data such as parity data or mirror data. The storage locations referenced by conventional metadata 210 and fault tolerance metadata 212 may be on the local system 100 or on a distal system (not shown) coupled via a communications network, such as a local area network (LAN), wide area network (WAN), or the Internet. In accordance with another embodiment of the invention, the storage access routine 108 may be one routine or a series of routines for storing objects in a distributed environment. For example, the storage access routine 108 may store only a portion of an object while another storage access routine (not shown in FIG. 1) associated with a different processor unit and/or storage controller stores another portion of the object.

Referring to FIG. 3, an illustrative storage access routine method is shown. Following (or in concert with) object creation, the storage access routine 108 creates
an entry in the object index with the handle for the data object, at block 300. Then, at block 302, the storage access routine 108 stores the data object on the storage media 110 according to a specified fault tolerance technique. The storage access routine 108 also associates indications of the fault tolerance technique utilized with the fault tolerance metadata and generates fault tolerance data for the data object, as shown in block 304. For example, if data mirroring is specified as the fault tolerance technique for the data object, then the storage access routine 108 stores a duplicate copy of the data object on one or more additional storage devices and provides indications of storage locations for the copy in the fault tolerance metadata.

In one embodiment, the storage access routine 108 may (by default or by user specification) automatically assign the highest level of fault tolerance protection to system information such as directory and object index structures. In another embodiment, all users associated with a specified account (e.g., workgroup) may have a default fault tolerance level of RAID 5, while users associated with another specified account may have a default fault tolerance level of no protection. To access the data object at a later time, the storage access routine 108 utilizes metadata to determine the physical locations of the data blocks that make up the data object. The data object may then be retrieved and/or modified by the storage access routine 108.

In addition to maintaining fault tolerance metadata for each object, the storage access routine 108 may allow the metadata for an object to be modified to accommodate different fault tolerance techniques. Meanwhile, other objects may remain stored according to the previously established level of fault tolerance. For example, Table 3 shows a series of disk units 1-5 configured to operate as a single storage medium in accordance with one embodiment of the invention. Using fault tolerance metadata, data objects D and E may be stored on the series of disk units according to RAID Level 5 as previously described. The storage access routine 108 may store the data objects D and E as series of sequential data blocks labeled D00 through D11 and E00 through E11. According to RAID Level 5, the data blocks of a data object may be striped across the series of disk units 1-5 beginning, for example, with disk unit 3. Additionally, parity blocks labeled PD0-PD2 for data object
D and PE0-PE2 for data object E may be interleaved with the data blocks D and E on the series of disk units, as shown in Table 3. One parity block may be constructed for every four data blocks, such that each of the data blocks and the associated parity block are stored on separate disk units. For example, the data blocks D00 through D03 are striped across disk units 3, 4, 5, and 1 while the corresponding parity block PD0 is stored on disk unit 2. Again, in accordance with the invention, data blocks may be stored at any location on a disk unit.

<table>
<thead>
<tr>
<th>Disk Units</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>D00</td>
<td>D01</td>
<td>D02</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>D03</td>
<td>D04</td>
<td>D05</td>
<td>D06</td>
<td>D07</td>
</tr>
<tr>
<td></td>
<td>D08</td>
<td>D09</td>
<td>D10</td>
<td>D11</td>
<td>PD2</td>
</tr>
<tr>
<td></td>
<td>PD1</td>
<td>PD0</td>
<td>E00</td>
<td>E01</td>
<td>E02</td>
</tr>
<tr>
<td></td>
<td>E03</td>
<td>E04</td>
<td>E05</td>
<td>E06</td>
<td>E07</td>
</tr>
<tr>
<td></td>
<td>E08</td>
<td>E09</td>
<td>E10</td>
<td>E11</td>
<td>PE2</td>
</tr>
<tr>
<td></td>
<td>PE1</td>
<td>PE0</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

When specifying fault tolerance techniques at the object level as shown in Table 3, fault tolerance metadata and data for the data object D may be modified to indicate a data mirroring technique, while the data object E remains stored according to RAID Level 5. In accordance with the changed fault tolerance metadata, storage access routine 108 may create a mirror of the data object D as shown in Table 4 by first creating mirror data blocks labeled PD00 to PD11 that are copies of data blocks D00 through D11 and discarding the previous parity blocks PD0 through PD2 previously stored on disk units 1, 2 and 5. The parity blocks PD00 to PD11 may be stored as a mirror of the data object D on any other suitable storage device, such as an additional series of disk units labeled 6-10. Alternatively, the parity blocks PD00 to PD11 may be stored on disk units labeled 1-5, such that the
parity blocks shown on units 6, 7, 8, 9, and 10 are stored on units 2, 3, 4, 5, and 1, respectively. While the parity blocks are shown distributed across disk units 6 through 10, each parity block may in fact be stored on any disk unit that does not contain data used to compute that parity block. For example, all parity blocks may be stored on a single separate disk unit.

Table 4. Data mirroring at the object level

<table>
<thead>
<tr>
<th></th>
<th>Disk Units</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>D00</td>
<td>D01</td>
<td>D02</td>
<td></td>
<td></td>
<td>PD00</td>
<td>PD01</td>
<td>PD02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D03</td>
<td>D04</td>
<td>D05</td>
<td>D06</td>
<td>D07</td>
<td>PD03</td>
<td>PD04</td>
<td>PD05</td>
<td>PD06</td>
<td>PD07</td>
<td></td>
</tr>
<tr>
<td>D08</td>
<td>D09</td>
<td>D10</td>
<td>D11</td>
<td></td>
<td>PD08</td>
<td>PD09</td>
<td>PD10</td>
<td>PD11</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>E00</td>
<td>E01</td>
<td>E02</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E03</td>
<td>E04</td>
<td>E05</td>
<td>E06</td>
<td>E07</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E08</td>
<td>E09</td>
<td>E10</td>
<td>E11</td>
<td>PE2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PE1</td>
<td>PE0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Additionally, an object entry in the object index corresponding to the data object D may be modified to indicate data mirroring as the fault tolerant technique to be used for the object D. Therefore, the fault tolerance technique applied to data object D may be modified without moving the data blocks D00 to D11 stored on disk units 1-5 and without disturbing the storage of data object E. This is in contrast with conventional fault tolerant techniques, which, in order to change the fault tolerant technique to mirroring for a single data object D, require modification of the entire volume on which the data object D is stored. For example, if conventional fault tolerant techniques had been used to mirror data object D, then data object E would have been mirrored as well, rather than remaining stored according to RAID Level 5. In accordance with the invention, the system 100 need only construct a mirror of the single data object D and modify the object's index entry.
When retrieving an object, part of whose data has been stored on a failed storage unit with fault tolerance data constructed based on RAID Level 5 techniques, if the fault tolerance data was constructed as described herein, only data from the object being retrieved is necessary to reconstruct all of the data of that object. In contrast, with conventional Raid Level 5 techniques (as shown previously in Table 2) substantial data from other objects may be necessary in reconstructing failed portions of an object being retrieved. This considerably affects performance in this degraded mode of operation.

In each of the previous examples, fault tolerance data may include parity information in addition to mirror data or any other fault tolerance techniques. Parity information may be stored as parity blocks generated from data blocks logically arranged in parity stripes. The parity blocks may be stored in any storage location as indicated by the corresponding fault tolerance metadata.

By maintaining fault tolerance metadata for data objects, several benefits may be gained. A computer storage system in accordance with the invention allows for flexibility in the manner in which fault tolerance is provided for data in the system. Because fault tolerance techniques are specified for individual data objects, each file or directory in the system may be stored in accordance with a different fault tolerance technique. Also, by defining fault tolerance at the data object level, the fault tolerance technique utilized to store a particular data object may be modified to store the data object using another technique -- without causing any other object stored on the same storage medium to be modified.

While the invention has been disclosed with respect to a limited number of embodiments, numerous modifications and variations will be appreciated by those skilled in the art. For example, the invention allows for a data object's level of fault tolerance to be modified from any initial state (e.g., no protection or RAID level 5) to any other state (e.g., no protection or RAID level 3). It is intended, therefore, that the following claims cover all such modifications and variations that may fall within the true spirit and scope of the invention.
What is claimed is:

1. A method to store a data object in a storage medium, comprising:
   associating a fault tolerance indication with the data object; and
   storing the data object in the storage medium in accordance with the
   fault tolerance indication.

2. The method of claim 1, wherein the act of storing the data object
   further comprises generating fault tolerance data for the data object in accordance
   with the fault tolerance indication.

3. The method of claim 2, wherein the fault tolerance data comprises
   parity information.

4. The method of claim 2, wherein the fault tolerance data comprises
   mirror data.

5. The method of claim 1, further comprising:
   receiving an identifier associated with the stored data object;
   modifying the fault tolerance indication of the stored data object; and
   modifying the storage of fault tolerance data for the data object in
   accordance with the modified fault tolerance indication.

6. The method of claim 5, wherein the act of modifying the storage of
   fault tolerance data for a first data object in accordance with a modified fault
   tolerance indication is independent of a fault tolerance indication for a second data
   object and the act of storing of fault tolerance data for second data object.

7. The method of claim 1, wherein the act of associating the fault
   tolerance indication with the data object comprises storing the identifier and fault
   tolerance indication in a memory accessible by a storage access program.
8. The method of claim 7, wherein the act of associating the fault
tolerance indication comprises associating an indication of a specified RAID level.

9. The method of claim 1, wherein the act of associating a first fault
tolerance indication with a first data object is independent of the act of associating a
second fault tolerance indication with a second data object.

10. The method of claim 1, wherein the storage medium comprises a
plurality of storage volumes.

11. The method of claim 10, wherein the act of associating the fault
tolerance indication assigns a first fault tolerance indication to a first data object and
a second fault tolerance indication to a second data object.

12. The method of claim 1, wherein the act of storing the data object
comprises storing a file data object.

13. The method of claim 1, wherein the act of storing the data object
comprises storing a directory object.

14. A computer program capable of running in a programmable control
device so that the control device so programmed carries out a method comprising:
associating a fault tolerance indication with a data object; and
storing the data object in a storage medium in accordance with the
fault tolerance indication.

15. The computer program of claim 14, wherein storing the data object
further comprises generating fault tolerance data for the data object in accordance
with the fault tolerance indication.
16. The computer program of claim 15, wherein the fault tolerance data comprises parity information.

17. The computer program of claim 15, wherein the fault tolerance data comprises mirror data.

18. The computer program of claim 14, wherein the method further comprises:
   receiving an identifier associated with the stored data object;
   modifying the fault tolerance indication of the stored data object; and
   modifying the storage of fault tolerance data for the data object in accordance with the modified fault tolerance indication.

19. The computer program of claim 18, wherein modifying the storage of fault tolerance data for a first data object in accordance with a first fault tolerance indication operates independently of a second fault tolerance indication for a second data object and storing the second data object.

20. The computer program of claim 14, wherein associating a fault tolerance indication with the data object comprises storing the identifier and associated fault tolerance indication in a memory accessible by a storage retrieval program.

21. The computer program of claim 14, wherein associating a fault tolerance indication comprises associating an indication of a specified RAID level.

22. The computer program of claim 14, wherein the storage medium comprises a plurality of storage volumes.
23. The computer program of claim 22, wherein associating a fault
tolerance indication comprises assigning a first fault tolerance indication to a first
data object and a second fault tolerance indication to a second data object.

24. The computer program of claim 14, wherein storing the data object
comprises storing a file data object.

25. The computer program of claim 14, wherein storing the data object
comprises storing a directory object.

26. A memory for storing data for access by a storage retrieval program
being executed on a programmable control device, comprising:
   a data structure stored in said memory for use by the storage retrieval
   program, said data structure including a plurality of entries, each entry including:
   an object identifier to identify a stored object, and
   a fault tolerant identifier to identify a level of fault tolerance
   applied to the stored object by the storage retrieval program during storage and
   retrieval operations.

27. The memory of claim 26, wherein a fault tolerant identifier for a first
entry in the data structure may be established by the storage retrieval program
independent of a fault tolerant identifier for a second entry in the data structure.

28. The memory of claim 27, wherein the first and second entries are
associated with objects stored on a single storage device volume.

29. The memory of claim 26, wherein each entry in the data structure
further comprises a stored object metadata field, the stored object metadata field
indicating a stored object access attribute.
30. The memory of claim 29, wherein the stored object metadata field further comprises an indication of a storage location of fault tolerance data.

31. The memory of claim 30, wherein the fault tolerance data comprises parity information.

32. The memory of claim 30, wherein the fault tolerance data comprises mirror data.

33. The memory of claim 29, wherein the stored object access attribute comprises a read/write access privilege indicator.

34. The memory of claim 26, wherein a stored object comprises a data file object.

35. The memory of claim 26, wherein a stored object comprises a directory object.

36. The memory of claim 26, wherein a stored object comprises the data structure.
FIG. 1
FIG. 2

START

CREATE ENTRY IN OBJECT INDEX

STORE DATA OBJECT

ASSOCIATE FAULT TOLERANCE INDICATION WITH METADATA AND GENERATE FAULT TOLERANCE DATA

END

FIG. 3
INTERNATIONAL SEARCH REPORT

A. CLASSIFICATION OF SUBJECT MATTER

IPC 7 G06F11/16

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC 7 G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic database consulted during the international search (name of database and, where practical, search terms used)

EPO-Internal, WPI Data, PAJ, INSPEC

C. DOCUMENTS CONSIDERED TO BE RELEVANT

<table>
<thead>
<tr>
<th>Category</th>
<th>Citation of document, with indication, where appropriate, of the relevant passages</th>
<th>Relevant to claim No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>WO 99 08173 A (SEAGATE TECHNOLOGY) 18 February 1999 (1999-02-18)</td>
<td></td>
</tr>
</tbody>
</table>

Further documents are listed in the continuation of box C. Patent family members are listed in annex.

* Special categories of cited documents:

- **A** document defining the general state of the art which is not considered to be of particular relevance
- **E** earlier document but published on or after the international filing date
- **L** document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
- **O** document referring to an oral disclosure, use, exhibition or other means
- **P** document published prior to the international filing date but later than the priority date claimed

* later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

**X** document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

**Y** document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

**A** document member of the same patent family

Date of the actual completion of the international search: 30 January 2001

Date of mailing of the international search report: 06/02/2001

Name and mailing address of the ISA

European Patent Office, P.B. 5818 Patentlaan 2
NL-2280 HV Hilversum
Tel. (+31-70) 340-2040, Tx. 31 651 epo nl
Fax (+31-70) 340-3016

Authorized officer

Absalom, R
<table>
<thead>
<tr>
<th>Patent document cited in search report</th>
<th>Publication date</th>
<th>Patent family member(s)</th>
<th>Publication date</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>DE 19882609 T</td>
<td>24-08-2000</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GB 2341466 A</td>
<td>15-03-2000</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DE 19882605 T</td>
<td>10-08-2000</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GB 2345167 A</td>
<td>28-06-2000</td>
</tr>
</tbody>
</table>