CBFS 简化格式文档

目录

1. HEADER

1.1 Structures

1.1.1 MAGIC NUMBER

4 bytes

“CBFS” or “CBFF”

1.1.2 VERSION

4 bytes

1,0,0,1

1.1.3 TIME STAMP

4 bytes

1.1.4 MAXIMUM SIZE

8 bytes

1.1.5 SECTOR SIZE

2 bytes

(1<<size)*1k

1.1.6 INDICATOR LEVEL COUNT

2 bytes

2~4, the count of INDICATOR LEVELS.

1.1.7 ROOT LEVEL INDICATOR

4 bytes

SECTORID

1.1.8 ROOT DIRECTOR NODE

8 bytes

STRUCTID

PADDING

1.1.9 INDEX NUM THRESH

2 bytes

1.1.10 INDEX COUNT THRESH

2 bytes

1.1.11 VARIABLE HEADER START

2 bytes

sector size unit

1.1.12 LOG SECTION START

2 bytes

sector size unit

1.1.13 LOG SECTION END

2 bytes

sector size unit

1.1.14 LOG ITEM GRANULATE

2 bytes

sector size unit

LOGSECTIONLENGTH must be even multiplications of GRANULATE

ALIGN TO SECTOR SIZE

2. VARIABLE HEADER

VARIABLE HEADER is formed of a list PAGE STRUCTURES

2.1 PAGE STRUCTURE

Within 1 sector

2.1.1 VALIDATION FLAG

4 bytes

0xFFFFFFFF

2.1.2 FREE SECTOR START

4 bytes

SECTORID

2.1.3 FREE NODE START

6 bytes

STRUCTID

2.1.4 FREE INDEX ENTRY START

6 bytes

STRUCTID

3. LOG SECTION

LOGSECTIONLENGTH=LOGSECTIONEND-LOGSECTIONSTART

3.1 LOG PAGE

LOGGRANULATE times SECTOR forms a LOG PAGE

3.2 LOG PAGE PAIR

2 contiguous LOG PAGES form a LOG PAGE PAIR

LOG SECTION is formed of a list of LOG PAGE PAIRS

3.3 LEADING LOG PAGE PAIR

The first LOG PAGE PAIR in the LOG SECTION is called LEADING LOG PAGE PAIR

3.3.1 Structures

3.3.1.1 ORDER FLAG

4 bytes

0x00000000 means positive order, 0xFFFFFFFF means reversed order.

3.4 LOGGED PAGES

a LOG PAGE PAIR contains LOGGED PAGES and LOGPAGEINFO, the order depends on ORDER FLAG.

LOGGED PAGES is formed of LOGGRANULATE LOGGED SECTORS, each LOGGED SECTOR is a backup copy of a SECTOR to be updated.

3.5 LOGPAGEINFO

a LOGPAGEINFO occupies the same size of LOGGED PAGES, it contains information of each corresponding LOGGED SECTOR.

3.5.1 Structures

3.5.1.1 VALIDATION FLAG

4 bytes

0xFFFFFFFF

3.5.1.2 PAGE PAIR START

2 bytes

The starting LOG PAGE PAIR ID in the LOG SECTION that contains LOG data. This number must be smaller than or equal to the ID of the LOG PAGE PAIR containing this LOGPAGEINFO

structure.

3.5.1.3 LOGGED SECTORS COUNT

2 bytes

The count of LOGGED SECTORS.

3.5.1.4 LOGGED SECTOR POSITIONS

a list of 4-byte SECTORIDS

the count of SECTORIDS must be equal to LOGGED SECTORS COUNT, each SECTORID specifies the position of the corresponding LOGGED SECTOR.

4. SECTOR

The storage system is managed through a minimum size unit called SECTOR, each SECTOR has a size equal to SECTOR SIZE and must align to SECTOR MARGIN.

The whole storage space is formed of contiguous SECTORS.

HEADERS and LOG SECTION must have sizes of multiplications of SECTOR SIZE and must align to SECTOR MARGINS.

5. SECTORID

For a SECTOR, SECTORID=SECTORSTARTOFFSET/SECTORSIZE.

a SECTORID occupies 4 bytes. So the storage system can have maximum 4G SECTORS, except the invalid SECTORIDS 0xFFFFFFF0~0xFFFFFFFF

6. STRUCTID

Sometimes we have to reference structures within a SECTOR, such as INDEX ENTRIES and NODE ENTRIES. A STRUCTID can do this. It is formed of a 4-byte SECTORID and a 2-byte STRUCT INDEX, and thus occupies 6 bytes.

7. LEVEL INDICATE SECTOR

To specify the type and address of each SECTOR, we introduce the concept of n-LEVEL INDICATE SECTORS,(n=0~3).

It can support up to 4 levels.

An i-LEVEL INDICATE SECTOR contains list of 4-byte SECTORIDS of (i-1)-LEVEL INDICATE SECTORS.

The 0-LEVEL INDICATE SECTOR contains list of 4-byte SECTORIDS of the next SECTOR if this is a stream SECTOR or free SECTOR and is not the last.

There are special SECTORIDS indicating specific SECTORS.

Value	Meaning
0xFFFFFFFF	Last SECTOR (stream or free)
0xFFFFFFFE	FREE SECTOR
0xFFFFFFFD	NODE SECTOR
0xFFFFFFFC	INDEX SECTOR
0xFFFFFFFB	LOWEST LEVEL INDICATE SECTOR
0xFFFFFFFA	LEVEL INDICATE SECTOR
0xFFFFFFF9	LOGSECTION
0xFFFFFFF8	HEADERS
0xFFFFFFF7 ~0xFFFFFFF0	Reserved

8. ROOT LEVEL INDICATE SECTOR

The highest LEVEL INDICATE SECTOR is called ROOT LEVEL INDICATE SECTOR, where the last

SECTORID is in substitute of a SECTORID INDICATING the next ROOT LEVEL INDICATE SECTOR.

9. NODE

A NODE is an in-SECTOR structure that specifies a directory or a file.

9.1 Structures

9.1.1 MAGIC

4 bytes

“node”

9.1.2 TYPE

2 bytes

0- DIRECTORY NODE

1- FILE NODE

2- EXTRA NODE

0xFF-FREE NODE

9.1.3 NEXT NODE(FILE NODE has not this field)

6 bytes

STRUCTID

If this is a DIRECTORY NODE, this points to the next sibling sub NODE.

If this is an EXTRA NODE, this points to the next EXTRA NODE.

If this is a FREE NODE, this is the next pointer to form a free node chain.

If no next NODE, this field is set to 0xFFFFFFFFFFFF

9.1.4 EXTRA NAME(For EXTRA NODE only)

Bytes stream extending to 256 bytes offset.

The following fields are invalid for EXTRA NODE.

9.1.5 PREV NODE(For DIRECTORY NODE and FREE NODE)

6 bytes

STRUCTID

If this is a DIRECTORY NODE, this field points to the previous sibling sub NODE.

If this is a FREE NODE, this field is the previous pointer to form a free node chain.

If no previous NODE, this field is set to 0xFFFFFFFFFFFF

The following fields are invalid for FREE NODE.

9.1.6 FIRST SUB NODE(For DIRECTORY NODE only)

6 bytes

STRUCTID

This points to the first sub NODE.

If no sub NODE, this field is set to 0xFFFFFFFFFFFF

9.1.7 PARENT NODE

6 bytes

STRUCTID

This points to the parent NODE.

If no parent NODE, this field is set to 0xFFFFFFFFFFFF

9.1.8 PADDLING(For FILE NODE only)

2 bytes

9.1.9 NAME SIZE

2 bytes

9.1.10 ATTRIBUTE FLAGS

2 bytes

Bit	Usage
0x0001	Hidden
0x0002	Read only
0x0004	System
0x0008	Archive

9.1.11 LOCK COUNT

2 bytes

9.1.12 PRIVILEDGES

3 bytes

byte 1: non-owner privilege

byte 2: owner privilege

byte 3: administrator privilege

For each byte:

bit 0 read

bit 1: write

bit 3: execute

9.1.13 OWNER ID

1 bytes

0 for administrator

9.1.14 CREATION TIME

8 bytes

TIME STRUCT

9.1.15 MODIFICATION TIME

8 bytes

TIME STRUCT

9.1.16 ACCESS TIME

8 bytes

TIME STRUCT

9.1.17 CONTENT SIZE

4 bytes

For FILE NODE, this field is the file size.

For DIRECTORY NODE, this field is the count of sub NODES.

9.1.18 FILE SIZE HIGH(For FILE NODE only)

4 bytes

9.1.19 START SECTOR(For FILE NODE only)

4 bytes

SECTORID

9.1.20 CONTENT INDEX ENTRY

6 bytes

STRUCTID

If this is a DIRECTORY NODE, this field specifies a ROOT INDEX ENTRY for NODE INDEX.

If this is a FILE NODE, this field specifies a ROOT INDEX ENTRY for STREAM INDEX.
If no INDEX, this field is set to 0xFFFFFFFFFFFF.
9.1.21 EXTRA NODE FOR NAME
6 bytes
STRUCTID
This field specifies an EXTRA NODE for NODE NAME.
If no extra node, this field is set to 0xFFFFFFFFFFFF
9.1.22 NAME
bytes stream
This field extends to 256 bytes offset.
10. DIRECTORY NODE
See NODE.
11. FILE NODE
See NODE.
12. EXTRA NODE
See NODE.
13. INDEX
INDEX is an in-SECTOR structure that faster NODE searching and STREAM SECTOR POSITIONING.
Within 1 SECTOR, INDEX is formed of a list of INDEX ENTRIES.
The INDEX ENTRIES form a B tree.
13.1 INDEX ENTRIES
13.1.1 Structures
13.1.1.1 COUNT
4 bytes
13.1.1.2 NEXT ENTRY
6 bytes
STRUCTID
This addresses the next ENTRY of the B tree.
If this is a FREE ENTRY, this field is a next pointer to form a FREE ENTRY list.
If no next ENTRIES, this field is set to 0xFFFFFFFFFFFF.
13.1.1.3 PREV ENTRY
6 bytes
STRUCTID
This addresses the previous ENTRY of the B tree.
If this is a FREE ENTRY, this field is a previous pointer to form a FREE ENTRY list.
If no previous ENTRIES, this field is set to 0xFFFFFFFFFFFF.
13.1.1.4 FIRST CHILD ENTRY
6 bytes
STRUCTID
This addresses the first child ENTRY of the B tree.
If this is leaf ENTRY, the highest bit of STRUCT INDEX is set to 1.
If this is a FREE ENTRY, this field is set to 0xFFFFFFFFFFFF.
13.1.1.5 PARENT ENTRY

6 bytes
STRUCTID
This addresses the parent ENTRY of the B tree.
If no parent ENTRIES, this field is set to 0xFFFFFFFFFFFF.
Totally 28 bytes.
14. NODE INDEX
The COUNT field is unused.
The NODES is ordered by NAME.
15. STREAM INDEX
The COUNT field denotes the number of SECTORS this INDEX ENTRY covers.
The STRUCT INDEX field in the STRUCTID is unused.

Appendix: Standalone Structures
A.1 TIME STRUCT
1.1 Structures
1.1.1 YY
2 bytes
Year
1.1.2 MM
1 bytes
Month
1.1.3 DD
1 bytes
Day
1.1.4 HR
1 bytes
Hour
1.1.5 MI
1 bytes
Minute
1.1.6 SC
1 bytes
Second
1.1.7 MS
1 bytes
Milli-second/4
Totally 8 bytes