

Spares

Object ID

Unique id for each files/dirs/links. Data chunks use this field to keep their file id.

Size:

32 Bits

Limit:

2^{32} -16 objects

Special:

- 1 root
- 2 lost+found
- 3 deleted

Version ¶

Version is incremented each time an object is modified. For data chunks, it works like that (the version is shared between header and chunks):

- In-Ram - 7 (max version)
- Header - 7
- Data chunk 1 - 6
- Data chunk 2 - 4
- Data chunk 3 - 5
- Data chunk 2 - 3 <- invalid chunk (3 < 4)

Size:

32 Bits

Limit:

2^{32} modifications per file/dir/links (we can always rewrite the entire object with a new id when the version is too big)

~ = 27 modification/sec for 5 years (without cache, without object rewriting)

Type

Object type

Size:

8 Bits

Limit:

256 Types

Special:

- NANDFS_OBJECT_TYPE_UNKNWON,
- NANDFS_OBJECT_TYPE_BAD,
- NANDFS_OBJECT_TYPE_FILE,
- NANDFS_OBJECT_TYPE_DIR,

- NANDFS_OBJECT_TYPE_DATA,
- NANDFS_OBJECT_TYPE_FREE = 0xFF

Marker

Just a marker to know if it's a NandFs? page

Value :

NANDFS_MAGIC_MARKER 0xAA

Size:

8 Bits

Files/Dirs - Parent Id

See Object Id

Size:

32 Bits

Files/Dirs - Name Hash

A name hash to speed dirs lookup.

Size:

16 Bits

Data chunk - offset

Size:

32 Bits

Limit:

$2^{32} * 2^{16}$ max file size (it's big for a Nand Flash)

Data chunk - Length (unused)

Size:

16 Bits

Limit:

2^{16} LogicalPage? size (it will never exceed 2048, so it's ok)

Conclusion

We need 16B of free spare per object.