

Blocks Informations

Blocks informations are stored in an array in the nandfs struct :

```
typedef enum {
    NANDFS_BLOCK_STATE_UNKNOWN,    /* ??? */
    NANDFS_BLOCK_STATE_USED,       /* We use this block, but it's not full */
    NANDFS_BLOCK_STATE_DIRTY,      /* This block contain dirty pages */
    NANDFS_BLOCK_STATE_FULLY_DIRTY, /* This block contain dirty pages, and */
    NANDFS_BLOCK_STATE_FULL,       /* This block is full */
    NANDFS_BLOCK_STATE_BAD, /* It's a bad block, but with pages used on it */
    NANDFS_BLOCK_STATE_DEAD,      /* It's a bad block, but dead (unused) */
    NANDFS_BLOCK_STATE_FREE = 0xFF /* Free block */
} nandfs_block_state;

typedef struct {
    nandfs_block_state state;
    uint16 pages_dirty;
    uint16 pages_free;
    uint16 pages_bad;
} nandfs_block;

typedef struct {
    /* ... */
    /* Blocks */
    nandfs_block *blocks;
    uint alloc_block;
    uint gc_alloc_block;
    uint block_scan_dirty_i;
    uint block_scan_free_i
    /* ... */
} nandfs;
```

There are functions in nandfs_block.c to remove/add pages and change status for a specified block. These informations are used by the Garbage Collector to clean dirty blocks, move bad blocks, find a free page, etc ... Blocks indexes in nandfs->block are the same as physical block indexes on the Nand.

"block_scan_xxx_i" allow to get a better wear leveling, scanning the block array from where last scan stop.

"alloc_block" is the block where free pages will be taken. "gc_alloc_block" is a block reserved for the gc.