Garbage Collector

(See: Block pools)

GC Thread

```
nandfs_gc(nandfs *fs)
```

This function should be called in a thread. The FS must be locked when this function is called. The granularity of the GC thread is defined by nandfs->gc_granularity (See Options)

Find a free page

```
int nandfs_find_free_page(nandfs * fs, uint32 * page, bool gc)
```

If gc = true the algorithm can use the reserved block (first free block), else nandfs_gc_clean_dirty_block can be called. The function try to use block pools (and scan the block array if they are empty). Return NANDFS_FAIL_NOSPC if no pages can be found.

Clean a dirty block

```
int nandfs_gc_clean_dirty_block(nandfs * fs, uint min)
```

This function try to get space from dirty blocks. Fully dirty blocks are cleaned first (partially dirty blocks need to be rewritten so it's slower). "min" is the minimum number of dirty page to consider a page as partially dirty (when we need space, min = 1, else when the function is called by the gc thread, min = nandfs->min_dirty_pages_to_clean (see Options))

Partially dirty

```
static int clean_dirty(nandfs * fs, uint block)
```

Rewrite and erase a block. If the rewrite didn't completed successfully, the block won't be erased, and the function will return NANDFS_FAIL_IO or NANDFS_FAIL_NOSPC.

Fully dirty

```
static int clean_fully_dirty(nandfs * fs, uint block)
```

Just erase the block.

Move a bad block

```
static int move_bad(nandfs * fs, uint block)
int nandfs_gc_move_bad_blocks(nandfs *fs)
```

move_bad just try to rewrite the block (we don't care if IO errors happen). Then erase it, and if the block is not full of FF, mark it as dead.

Rewriting a block

```
static int rewrite_block(nandfs *fs, uint block);
```

When we clean a partially dirty block and when we move a bad bloc, we need to rewrite this block. All pages are scanned, and non dirty pages are rewrited to a new block. To know if an object is not dirty, we search if the object is present in-ram, and if the page is the same. If we fail to read/write an object (ECC error, Hardware error ..), we try the next object, but the page is not marked dirty and the function will return NANDFS_FAIL_IO. If there is no space left on the device to move a page, the function return NANDFS_FAIL_NOSPC. To find a new free page, nandfs_find_free_page is called with gc = true, so we can use a reserved block and we avoid an infinit recursion (find_free_page -> clean -> find_free_page -> clean, etc ...).