

Scan Algorithm

Main Loop

```
static int scan(nandfs * fs)

for_each_blocks(block) {
    for_each_page(page) {
        object = object_from_spare(page);
        object->type
        | = NANDFS_OBJECT_TYPE_FREE
        | goto_next_block;
        | = NANDFS_OBJECT_TYPE_BAD
        | mark_page_bad(page);
        | = NANDFS_OBJECT_TYPE_DATA
        | link data to nandfs->chunk_tmp
        | = NANDFS_OBJECT_TYPE_HARDLINK
        | link hardlink to nandfs->hardlink
        | = NANDFS_OBJECT_TYPE_FILE/DIR/SOCKET/...
        | scan_generic(object);
        remove_free_page(page);
    }
}

add_chunks();
add_hardlinks();
```

Chunks and hardlinks are collected and added later because:

- Hardlinks need their equivalent object
- Chunks need their file

object from spare

(see SpareMapping)

```
nandfs_object *nandfs_object_from_spare(nandfs * fs, uint32 page);
```

```
object = alloc_objec();
read_spare(object)
| //read spare vars for this page
| if(verify_on_mount && ecc) {
|     // read data an verify ecc
|     // mark the object as bad if ecc correction failed
| }
```

scan_generic

```
static void scan_generic(nandfs * fs, nandfs_object * obj, uint32 page)
```

add_chunks

```
static void add_collected_chunks(nandfs * fs)
```

```
for_each_chunks(chunk) {  
    object = find_object_by_id(chunk->id);  
    if(object) {  
        add_chunk(object, chunk);  
    } else {  
        mark_page_dirty(chunk->page);  
        free_chunks(chunk);  
    }  
}
```

add_hardlinks

```
static void add_collected_hardlinks(nandfs * fs)
```

```
for_each_hardlink(hardlink) {  
    header = read_header(hardlink);  
  
    if(!header) {  
        mark_bad_page(hardlink->page);  
        add_to_parent(hardlink, fs->deleted);  
    } else {  
        equivalent = find_object_by_id(hardlink->id);  
        if(!equivalent) {  
            mark_bad_page(hardlink->page);  
            add_to_parent(hardlink, fs->deleted);  
        } else {  
            scan_generic(hardlink);  
            if(hardlink successfully added) {  
                add_hardlink_to_object(hardlink, equ);  
            }  
        }  
    }  
}
```