

ngx_file.h Documentation :

Macros Defined :

```
#define NGX_MAX_PATH_LEVEL 3
```

Type Definitions :

```
typedef time_t (*ngx_path_manager_pt) (void *data)
typedef void (*ngx_path_loader_pt) (void *data)
typedef struct ngx_tree_ctx_s ngx_tree_ctx_t
typedef ngx_int_t (*ngx_tree_init_handler_pt) (void *ctx, void *prev)
typedef ngx_int_t (*ngx_tree_handler_pt) (ngx_tree_ctx_t *ctx, ngx_str_t *name)
```

Data Structures :

```
struct ngx_file_s {
    ngx_fd_t          fd;
    ngx_str_t         name;
    ngx_file_info_t    info;
    off_t             offset;
    off_t             sys_offset;
    ngx_log_t         *log;
#ifdef NGX_HAVE_FILE_AIO
    ngx_event_aio_t    *aio;
#endif
    unsigned           valid_info:1;
    unsigned           directio:1;
}

struct ngx_tree_ctx_s {
    off_t             size;
    off_t             fs_size;
    ngx_uint_t        access;
    time_t            mtime;
    ngx_tree_init_handler_pt init_handler;
    ngx_tree_handler_pt file_handler;
    ngx_tree_handler_pt pre_tree_handler;
    ngx_tree_handler_pt post_tree_handler;
    ngx_tree_handler_pt spec_handler;
    void              *data;
    size_t            alloc;
    ngx_log_t         *log;
}
```

```
typedef struct {
    ngx_str_t      name;
    size_t          len;
    size_t          level[3];
    ngx_path_manager_pt  manager;
    ngx_path_loader_pt  loader;
    void            *data;
    u_char          *conf_file;
    ngx_uint_t      line;
} ngx_path_t;
```

```
typedef struct {
    ngx_str_t      name;
    size_t          level[3];
} ngx_path_init_t;
```

```
typedef struct {
    ngx_file_t      file;
    off_t            offset;
    ngx_path_t      *path;
    ngx_pool_t      *pool;
    char            *warn;
    ngx_uint_t      access;
    unsigned         log_level:8;
    unsigned         persistent:1;
    unsigned         clean:1;
} ngx_temp_file_t;
```

```
typedef struct {
    ngx_uint_t      access;
    ngx_uint_t      path_access;
    time_t          time;
    ngx_fd_t        fd;
    unsigned         create_path:1;
    unsigned         delete_file:1;
    ngx_log_t        *log;
} ngx_ext_rename_file_t;
```

```
typedef struct {
    off_t      size;
    size_t     buf_size;
    ngx_uint_t access;
    time_t     time;
    ngx_log_t  *log;
} ngx_copy_file_t;
```

Functions Defined :

```
ngx_int_t ngx_get_full_name( ngx_pool_t *pool,
                             ngx_str_t *prefix,
                             ngx_str_t *name
                             )
```

```
ssize_t ngx_write_chain_to_temp_file( ngx_temp_file_t *tf,
                                       ngx_chain_t *chain
                                       )
```

```
ngx_int_t ngx_create_temp_file( ngx_file_t *file,
                                ngx_path_t *path,
                                ngx_pool_t *pool,
                                ngx_uint_t persistent,
                                ngx_uint_t clean,
                                ngx_uint_t access
                                )
```

```
void ngx_create_hashed_filename( ngx_path_t *path,
                                 u_char *file,
                                 size_t len
                                 )
```

```
ngx_int_t ngx_create_path( ngx_file_t *file,
                           ngx_path_t *path
                           )
```

```
ngx_err_t ngx_create_full_path( u_char *dir,
                                ngx_uint_t access
                                )
```

```
ngx_int_t ngx_add_path( ngx_conf_t *cf,
                        ngx_path_t **slot
                        )
```

```

ngx_int_t ngx_create_paths( ngx_cycle_t *cycle,
                            ngx_uid_t user
                            )

ngx_int_t ngx_ext_rename_file( ngx_str_t *src,
                               ngx_str_t *to,
                               ngx_ext_rename_file_t *ext
                               )

ngx_int_t ngx_copy_file( u_char *from,
                        u_char *to,
                        ngx_copy_file_t *cf
                        )

ngx_int_t ngx_walk_tree( ngx_tree_ctx_t *ctx,
                        ngx_str_t *tree
                        )

ngx_atomic_uint_t ngx_next_temp_number( ngx_uint_t collision
                                        )

char *ngx_conf_set_path_slot( ngx_conf_t *cf,
                             ngx_command_t *cmd,
                             void *conf
                             )

char *ngx_conf_merge_path_value( ngx_conf_t *cf,
                                ngx_path_t **path,
                                ngx_path_t *prev,
                                ngx_path_init_t *init
                                )

char *ngx_conf_set_access_slot( ngx_conf_t *cf,
                               ngx_command_t *cmd,
                               void *conf
                               )

```

Include Dependency Graph :

