

ngx_open_file_cache.h Documentation:

Macros Defined:

```
#define NGX_OPEN_FILE_DIRECTIO_OFF  NGX_MAX_OFF_T_VALUE
```

Type Definitions:

```
typedef struct ngx_cached_open_file_s  ngx_cached_open_file_t;
```

Data Structures Defined:

```
typedef struct {
    ngx_fd_t      fd;
    ngx_file_uniq_t  uniq;
    time_t        mtime;
    off_t         size;
    off_t         fs_size;
    off_t         directio;
    size_t        read_ahead;
    ngx_err_t      err;
    char          *failed;
    time_t        valid;
    ngx_uint_t     min_uses;

    #if (NGX_HAVE_OPENAT)
        size_t      disable_symlinks_from;
        unsigned    disable_symlinks:2;
    #endif

    unsigned       test_dir:1;
    unsigned       test_only:1;
    unsigned       log:1;
    unsigned       errors:1;
    unsigned       events:1;
    unsigned       is_dir:1;
    unsigned       is_file:1;
    unsigned       is_link:1;
    unsigned       is_exec:1;
    unsigned       is_directio:1;
} ngx_open_file_info_t;
```

```
struct ngx_cached_open_file_s {
    ngx_rbtree_node_t  node;
    ngx_queue_t        queue;
    u_char             *name;
    time_t             created;
    time_t             accessed;
    ngx_fd_t           fd;
    ngx_file_uniq_t    uniq;
    time_t             mtime;
    off_t              size;
    ngx_err_t          err;
    uint32_t           uses;

    #if (NGX_HAVE_OPENAT)
        size_t      disable_symlinks_from;
        unsigned    disable_symlinks:2;
    #endif
}
```

```

    unsigned    count:24;
    unsigned    close:1;
    unsigned    use_event:1;
    unsigned    is_dir:1;
    unsigned    is_file:1;
    unsigned    is_link:1;
    unsigned    is_exec:1;
    unsigned    is_directio:1;
    ngx_event_t *event;
};

```

```

typedef struct {
    ngx_rbtree_t    rbtree;
    ngx_rbtree_node_t    sentinel;
    ngx_queue_t    expire_queue;
    ngx_uint_t    current;
    ngx_uint_t    max;
    time_t    inactive;
} ngx_open_file_cache_t;

```

```

typedef struct {
    ngx_open_file_cache_t    *cache;
    ngx_cached_open_file_t    *file;
    ngx_uint_t    min_uses;
    ngx_log_t    *log;
} ngx_open_file_cache_cleanup_t;

```

```

typedef struct {
    void    *data;
    ngx_event_t    *read;
    ngx_event_t    *write;
    ngx_fd_t    fd;
    ngx_cached_open_file_t    *file;
    ngx_open_file_cache_t    *cache;
} ngx_open_file_cache_event_t;

```

Functions Defined :

```

ngx_open_file_cache_t *ngx_open_file_cache_init( ngx_pool_t *pool,
                                                ngx_uint_t max,
                                                time_t inactive
                                                )

```

```

ngx_int_t ngx_open_cached_file( ngx_open_file_cache_t *cache,
                                ngx_str_t *name,
                                ngx_open_file_info_t *of,
                                ngx_pool_t *pool
                                );

```

Include Dependency Graph :

