

# File system operations

libuv provides a wide variety of cross-platform sync and async file system operations. All functions defined in this document take a callback, which is allowed to be NULL. If the callback is NULL the request is completed synchronously, otherwise it will be performed asynchronously.

All file operations are run on the threadpool. See [ref: threadpool](#) for information on the threadpool size.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src] fs.rst, line 12); backlink**  
Unknown interpreted text role "ref".

**Note**  
On Windows `uv_fs_*` functions use utf-8 encoding.

## Data types

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src] fs.rst, line 21)**  
Unknown directive type "ctype".  
  
`.. ctype:: uv_fs_t`  
  
File system request type.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src] fs.rst, line 25)**  
Unknown directive type "ctype".  
  
`.. ctype:: uv_timespec_t`  
  
Portable equivalent of ``struct timespec``.  
  
::  
  
typedef struct {  
 long tv\_sec;  
 long tv\_nsec;  
} uv\_timespec\_t;

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src] fs.rst, line 36)**  
Unknown directive type "ctype".  
  
`.. ctype:: uv_stat_t`  
  
Portable equivalent of ``struct stat``.  
  
::  
  
typedef struct {  
 uint64\_t st\_dev;  
 uint64\_t st\_mode;  
 uint64\_t st\_nlink;  
 uint64\_t st\_uid;  
 uint64\_t st\_gid;  
 uint64\_t st\_rdev;  
 uint64\_t st\_ino;  
 uint64\_t st\_size;  
 uint64\_t st\_blksize;  
 uint64\_t st\_blocks;  
 uint64\_t st\_flags;  
 uint64\_t st\_gen;  
 uv\_timespec\_t st\_atim;  
 uv\_timespec\_t st\_mtim;  
 uv\_timespec\_t st\_ctim;  
 uv\_timespec\_t st\_birthtim;  
} uv\_stat\_t;

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src] fs.rst, line 61)**  
Unknown directive type "cenum".  
  
`.. cenum:: uv_fs_type`  
  
File system request type.  
  
::  
  
typedef enum {  
 UV\_FS\_UNKNOWN = -1,  
 UV\_FS\_CUSTOM,  
 UV\_FS\_OPEN,  
 UV\_FS\_CLOSE,  
 UV\_FS\_READ,  
 UV\_FS\_WRITE,  
 UV\_FS\_SENDFILE,  
 UV\_FS\_STAT,  
 UV\_FS\_LSTAT,  
 UV\_FS\_FSTAT,  
 UV\_FS\_FTRUNCATE,  
 UV\_FS\_UTIME,  
 UV\_FS\_FUTIME,  
 UV\_FS\_ACCESS,  
 UV\_FS\_CHMOD,  
 UV\_FS\_FCHMOD,  
 UV\_FS\_FSYNC,  
 UV\_FS\_FDATASYNC,  
 UV\_FS\_UNLINK,  
 UV\_FS\_RMDIR,  
 UV\_FS\_MKDIR,  
 UV\_FS\_MKDTEMP,  
 UV\_FS\_RENAME,  
 UV\_FS\_SCANDIR,  
 UV\_FS\_LINK,  
 UV\_FS\_SYMLINK,  
 UV\_FS\_READLINK,  
 UV\_FS\_CHOWN,  
 UV\_FS\_FCHOWN,  
 UV\_FS\_REALPATH,  
 UV\_FS\_COPYFILE,  
}

```
UV_FS_LCHOWN,
UV_FS_OPENDIR,
UV_FS_READDIR,
UV_FS_CLOSEDIR,
UV_FS_MKSTEMP,
UV_FS_LUTIME
} uv_fs_type;
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]fs.rst, line 107)**

Unknown directive type "c:type".

```
.. c:type:: uv_statfs_t

Reduced cross platform equivalent of ``struct statfs``.
Used in :c:func:`uv_fs_statfs`.

::

typedef struct uv_statfs_s {
    uint64_t f_type;
    uint64_t f_bsize;
    uint64_t f_blocks;
    uint64_t f_bfree;
    uint64_t f_bavail;
    uint64_t f_files;
    uint64_t f_ffree;
    uint64_t f_spare[4];
} uv_statfs_t;
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]fs.rst, line 125)**

Unknown directive type "c:enum".

```
.. c:enum:: uv_dirent_t

Cross platform (reduced) equivalent of ``struct dirent``.
Used in :c:func:`uv_fs_scandir_next`.

::

typedef enum {
    UV_DIRENT_UNKNOWN,
    UV_DIRENT_FILE,
    UV_DIRENT_DIR,
    UV_DIRENT_LINK,
    UV_DIRENT_FIFO,
    UV_DIRENT_SOCKET,
    UV_DIRENT_CHAR,
    UV_DIRENT_BLOCK
} uv_dirent_type_t;

typedef struct uv_dirent_s {
    const char* name;
    uv_dirent_type_t type;
} uv_dirent_t;
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]fs.rst, line 148)**

Unknown directive type "c:type".

```
.. c:type:: uv_dir_t

Data type used for streaming directory iteration.
Used by :c:func:`uv_fs_opendir()`, :c:func:`uv_fs_readdir()`, and
:c:func:`uv_fs_closedir()`. 'dirents' represents a user provided array of
'uv_dirent_t's used to hold results. 'nentries' is the user provided maximum
array size of 'dirents'.

::

typedef struct uv_dir_s {
    uv_dirent_t* dirents;
    size_t nentries;
} uv_dir_t;
```

## Public members

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]fs.rst, line 167)**

Unknown directive type "c:member".

```
.. c:member:: uv_loop_t* uv_fs_t.loop

Loop that started this request and where completion will be reported.
Readonly.
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]fs.rst, line 172)**

Unknown directive type "c:member".

```
.. c:member:: uv_fs_type uv_fs_t.fs_type

FS request type.
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]fs.rst, line 176)**

Unknown directive type "c:member".

```
.. c:member:: const char* uv_fs_t.path

Path affecting the request.
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]fs.rst, line 180)**

Unknown directive type "c:member".

```
.. c:member:: ssize_t uv_fs_t.result
```

<p>Result of the request. &lt; 0 means error, success otherwise. On requests such as :c:func:`uv_fs_read` or :c:func:`uv_fs_write` it indicates the amount of data that was read or written, respectively.</p>
<p><b>System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]fs.rst, line 186)</b></p> <p>Unknown directive type "c:member".</p> <pre>.. c:member:: uv_stat_t uv_fs_t.statbuf</pre> <p>Stores the result of :c:func:`uv_fs_stat` and other stat requests.</p>
<p><b>System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]fs.rst, line 190)</b></p> <p>Unknown directive type "c:member".</p> <pre>.. c:member:: void* uv_fs_t.ptr</pre> <p>Stores the result of :c:func:`uv_fs_readlink` and :c:func:`uv_fs_realpath` and serves as an alias to `statbuf`.</p>
<p><b>System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]fs.rst, line 195)</b></p> <p>Unknown directive type "seealso".</p> <pre>.. seealso:: The :c:type:`uv_req_t` members also apply.</pre>

API

<p><b>System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]fs.rst, line 201)</b></p> <p>Unknown directive type "c:function".</p> <pre>.. c:function:: void uv_fs_req_cleanup(uv_fs_t* req)</pre> <p>Cleanup request. Must be called after a request is finished to deallocate any memory libuv might have allocated.</p>
<p><b>System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]fs.rst, line 206)</b></p> <p>Unknown directive type "c:function".</p> <pre>.. c:function:: int uv_fs_close(uv_loop_t* loop, uv_fs_t* req, uv_file file, uv_fs_cb cb)</pre> <p>Equivalent to :man:`close(2)`.</p>
<p><b>System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]fs.rst, line 210)</b></p> <p>Unknown directive type "c:function".</p> <pre>.. c:function:: int uv_fs_open(uv_loop_t* loop, uv_fs_t* req, const char* path, int flags, int mode, uv_fs_cb cb)</pre> <p>Equivalent to :man:`open(2)`.</p> <p>.. note:: On Windows libuv uses `CreateFileW` and thus the file is always opened in binary mode. Because of this the O_BINARY and O_TEXT flags are not supported.</p>
<p><b>System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]fs.rst, line 219)</b></p> <p>Unknown directive type "c:function".</p> <pre>.. c:function:: int uv_fs_read(uv_loop_t* loop, uv_fs_t* req, uv_file file, const uv_buf_t bufs[], unsigned int nbufs, int64_t off)</pre> <p>Equivalent to :man:`preadv(2)`.</p> <p>.. warning:: On Windows, under non-MSVC environments (e.g. when GCC or Clang is used to build libuv), files opened using `UV_FS_O_FILEMAP` may cause a fatal crash if the memory mapped read operation fails.</p>
<p><b>System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]fs.rst, line 228)</b></p> <p>Unknown directive type "c:function".</p> <pre>.. c:function:: int uv_fs_unlink(uv_loop_t* loop, uv_fs_t* req, const char* path, uv_fs_cb cb)</pre> <p>Equivalent to :man:`unlink(2)`.</p>
<p><b>System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]fs.rst, line 232)</b></p> <p>Unknown directive type "c:function".</p> <pre>.. c:function:: int uv_fs_write(uv_loop_t* loop, uv_fs_t* req, uv_file file, const uv_buf_t bufs[], unsigned int nbufs, int64_t off)</pre> <p>Equivalent to :man:`pwritev(2)`.</p> <p>.. warning:: On Windows, under non-MSVC environments (e.g. when GCC or Clang is used to build libuv), files opened using `UV_FS_O_FILEMAP` may cause a fatal crash if the memory mapped write operation fails.</p>
<p><b>System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]fs.rst, line 241)</b></p> <p>Unknown directive type "c:function".</p>

```
.. c:function:: int uv_fs_mkdir(uv_loop_t* loop, uv_fs_t* req, const char* path, int mode, uv_fs_cb cb)
```

Equivalent to :man:`mkdir(2)`.

.. note::  
`mode` is currently not implemented on Windows.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]fs.rst, line 248)**

Unknown directive type "c:function".

```
.. c:function:: int uv_fs_mkdtemp(uv_loop_t* loop, uv_fs_t* req, const char* tpl, uv_fs_cb cb)
```

Equivalent to :man:`mkdtemp(3)`. The result can be found as a null terminated string at `req->path`.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]fs.rst, line 252)**

Unknown directive type "c:function".

```
.. c:function:: int uv_fs_mkstemp(uv_loop_t* loop, uv_fs_t* req, const char* tpl, uv_fs_cb cb)
```

Equivalent to :man:`mkstemp(3)`. The created file path can be found as a null terminated string at `req->path`. The file descriptor can be found as an integer at `req->result`.

.. versionadded:: 1.34.0

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]fs.rst, line 259)**

Unknown directive type "c:function".

```
.. c:function:: int uv_fs_rmdir(uv_loop_t* loop, uv_fs_t* req, const char* path, uv_fs_cb cb)
```

Equivalent to :man:`rmdir(2)`.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]fs.rst, line 263)**

Unknown directive type "c:function".

```
.. c:function:: int uv_fs_opendir(uv_loop_t* loop, uv_fs_t* req, const char* path, uv_fs_cb cb)
```

Opens `path` as a directory stream. On success, a `uv\_dir\_t` is allocated and returned via `req->ptr`. This memory is not freed by `uv\_fs\_req\_cleanup()`, although `req->ptr` is set to `NULL`. The allocated memory must be freed by calling `uv\_fs\_closedir()`. On failure, no memory is allocated.

The contents of the directory can be iterated over by passing the resulting `uv\_dir\_t` to `uv\_fs\_readdir()`.

.. versionadded:: 1.28.0

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]fs.rst, line 276)**

Unknown directive type "c:function".

```
.. c:function:: int uv_fs_closedir(uv_loop_t* loop, uv_fs_t* req, uv_dir_t* dir, uv_fs_cb cb)
```

Closes the directory stream represented by `dir` and frees the memory allocated by `uv\_fs\_opendir()`.

.. versionadded:: 1.28.0

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]fs.rst, line 283)**

Unknown directive type "c:function".

```
.. c:function:: int uv_fs_readdir(uv_loop_t* loop, uv_fs_t* req, uv_dir_t* dir, uv_fs_cb cb)
```

Iterates over the directory stream, `dir`, returned by a successful `uv\_fs\_opendir()` call. Prior to invoking `uv\_fs\_readdir()`, the caller must set `dir->dirents` and `dir->nentries`, representing the array of :c:type:`uv\_dirent\_t` elements used to hold the read directory entries and its size.

On success, the result is an integer  $\geq 0$  representing the number of entries read from the stream.

.. versionadded:: 1.28.0

.. warning::  
`uv\_fs\_readdir()` is not thread safe.

.. note::  
This function does not return the "." and ".." entries.

.. note::  
On success this function allocates memory that must be freed using `uv\_fs\_req\_cleanup()`. `uv\_fs\_req\_cleanup()` must be called before closing the directory with `uv\_fs\_closedir()`.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]fs.rst, line 307)**

Unknown directive type "c:function".

```
.. c:function:: int uv_fs_scandir(uv_loop_t* loop, uv_fs_t* req, const char* path, int flags, uv_fs_cb cb)
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]fs.rst, line 308)**

Unknown directive type "c:function".

```
.. c:function:: int uv_fs_scandir_next(uv_fs_t* req, uv_dirent_t* ent)
```

Equivalent to :man:`scandir(3)`, with a slightly different API. Once the callback for the request is called, the user can use :c:func:`uv\_fs\_scandir\_next` to get `ent` populated with the next directory entry data. When there are no more entries ``UV\_EOF`` will be returned.

```
.. note::
    Unlike `scandir(3)`, this function does not return the "." and ".." entries.

.. note::
    On Linux, getting the type of an entry is only supported by some file systems (btrfs, ext2,
    ext3 and ext4 at the time of this writing), check the :man:`getdents(2)` man page.
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]fs.rst, line 322)**

Unknown directive type "c:function".

```
.. c:function:: int uv_fs_stat(uv_loop_t* loop, uv_fs_t* req, const char* path, uv_fs_cb cb)
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]fs.rst, line 323)**

Unknown directive type "c:function".

```
.. c:function:: int uv_fs_fstat(uv_loop_t* loop, uv_fs_t* req, uv_file file, uv_fs_cb cb)
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]fs.rst, line 324)**

Unknown directive type "c:function".

```
.. c:function:: int uv_fs_lstat(uv_loop_t* loop, uv_fs_t* req, const char* path, uv_fs_cb cb)
```

Equivalent to :man:`stat(2)`, :man:`fstat(2)` and :man:`lstat(2)` respectively.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]fs.rst, line 328)**

Unknown directive type "c:function".

```
.. c:function:: int uv_fs_statfs(uv_loop_t* loop, uv_fs_t* req, const char* path, uv_fs_cb cb)
```

Equivalent to :man:`statfs(2)`. On success, a `uv\_statfs\_t` is allocated and returned via `req->ptr`. This memory is freed by `uv\_fs\_req\_cleanup()`.

```
.. note::
    Any fields in the resulting `uv_statfs_t` that are not supported by the
    underlying operating system are set to zero.
```

```
.. versionadded:: 1.31.0
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]fs.rst, line 339)**

Unknown directive type "c:function".

```
.. c:function:: int uv_fs_rename(uv_loop_t* loop, uv_fs_t* req, const char* path, const char* new_path, uv_fs_cb cb)
```

Equivalent to :man:`rename(2)`.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]fs.rst, line 343)**

Unknown directive type "c:function".

```
.. c:function:: int uv_fs_fsync(uv_loop_t* loop, uv_fs_t* req, uv_file file, uv_fs_cb cb)
```

Equivalent to :man:`fsync(2)`.

```
.. note::
    For AIX, `uv_fs_fsync` returns `UV_EBADF` on file descriptors referencing
    non regular files.
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]fs.rst, line 351)**

Unknown directive type "c:function".

```
.. c:function:: int uv_fs_fdatasync(uv_loop_t* loop, uv_fs_t* req, uv_file file, uv_fs_cb cb)
```

Equivalent to :man:`fdatasync(2)`.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]fs.rst, line 355)**

Unknown directive type "c:function".

```
.. c:function:: int uv_fs_ftruncate(uv_loop_t* loop, uv_fs_t* req, uv_file file, int64_t offset, uv_fs_cb cb)
```

Equivalent to :man:`ftruncate(2)`.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]fs.rst, line 359)**

Unknown directive type "c:function".

```
.. c:function:: int uv_fs_copyfile(uv_loop_t* loop, uv_fs_t* req, const char* path, const char* new_path, int flags, uv_fs_cb cb)
```

Copies a file from `path` to `new\_path`. Supported `flags` are described below.

- `UV\_FS\_COPYFILE\_EXCL`: If present, `uv\_fs\_copyfile()` will fail with `UV\_EEXIST` if the destination path already exists. The default behavior is to overwrite the destination if it exists.
- `UV\_FS\_COPYFILE\_FICLONE`: If present, `uv\_fs\_copyfile()` will attempt to create a copy-on-write reflink. If the underlying platform does not support copy-on-write, or an error occurs while attempting to use copy-on-write, a fallback copy mechanism based on :c:func:`uv\_fs\_sendfile()` is used.
- `UV\_FS\_COPYFILE\_FICLONE\_FORCE`: If present, `uv\_fs\_copyfile()` will attempt to create a copy-on-write reflink. If the underlying platform does not support copy-on-write, or an error occurs while attempting to use copy-on-write, then an error is returned.

```
.. warning::
    If the destination path is created, but an error occurs while copying
    the data, then the destination path is removed. There is a brief window
```

```
of time between closing and removing the file where another process
could access the file.

.. versionadded:: 1.14.0

.. versionchanged:: 1.20.0 `UV_FS_COPYFILE_FICLONE` and
`UV_FS_COPYFILE_FICLONE_FORCE` are supported.

.. versionchanged:: 1.33.0 If an error occurs while using
`UV_FS_COPYFILE_FICLONE_FORCE`, that error is returned. Previously,
all errors were mapped to `UV_ENOTSUP`.
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]fs.rst, line 391)**

Unknown directive type "c:function".

```
.. c:function:: int uv_fs_sendfile(uv_loop_t* loop, uv_fs_t* req, uv_file out_fd, uv_file in_fd, int64_t in_offset, size_t length,
Limited equivalent to :man:`sendfile(2)`.
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]fs.rst, line 395)**

Unknown directive type "c:function".

```
.. c:function:: int uv_fs_access(uv_loop_t* loop, uv_fs_t* req, const char* path, int mode, uv_fs_cb cb)
Equivalent to :man:`access(2)` on Unix. Windows uses ``GetFileAttributesW()``.
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]fs.rst, line 399)**

Unknown directive type "c:function".

```
.. c:function:: int uv_fs_chmod(uv_loop_t* loop, uv_fs_t* req, const char* path, int mode, uv_fs_cb cb)
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]fs.rst, line 400)**

Unknown directive type "c:function".

```
.. c:function:: int uv_fs_fchmod(uv_loop_t* loop, uv_fs_t* req, uv_file file, int mode, uv_fs_cb cb)
Equivalent to :man:`chmod(2)` and :man:`fchmod(2)` respectively.
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]fs.rst, line 404)**

Unknown directive type "c:function".

```
.. c:function:: int uv_fs_utime(uv_loop_t* loop, uv_fs_t* req, const char* path, double atime, double mtime, uv_fs_cb cb)
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]fs.rst, line 405)**

Unknown directive type "c:function".

```
.. c:function:: int uv_fs_futime(uv_loop_t* loop, uv_fs_t* req, uv_file file, double atime, double mtime, uv_fs_cb cb)
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]fs.rst, line 406)**

Unknown directive type "c:function".

```
.. c:function:: int uv_fs_lutime(uv_loop_t* loop, uv_fs_t* req, const char* path, double atime, double mtime, uv_fs_cb cb)
Equivalent to :man:`utime(2)`, :man:`futimes(3)` and :man:`lutimes(3)` respectively.

.. note::
z/OS: `uv_fs_lutime()` is not implemented for z/OS. It can still be called but will return
`UV_ENOSYS`.

.. note::
AIX: `uv_fs_futime()` and `uv_fs_lutime()` functions only work for AIX 7.1 and newer.
They can still be called on older versions but will return `UV_ENOSYS`.

.. versionchanged:: 1.10.0 sub-second precision is supported on Windows
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]fs.rst, line 420)**

Unknown directive type "c:function".

```
.. c:function:: int uv_fs_link(uv_loop_t* loop, uv_fs_t* req, const char* path, const char* new_path, uv_fs_cb cb)
Equivalent to :man:`link(2)`.
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]fs.rst, line 424)**

Unknown directive type "c:function".

```
.. c:function:: int uv_fs_symlink(uv_loop_t* loop, uv_fs_t* req, const char* path, const char* new_path, int flags, uv_fs_cb cb)
Equivalent to :man:`symlink(2)`.

.. note::
On Windows the `flags` parameter can be specified to control how the symlink will
be created:

* `UV_FS_SYMLINK_DIR`: indicates that `path` points to a directory.
* `UV_FS_SYMLINK_JUNCTION`: request that the symlink is created
using junction points.
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]fs.rst, line 437)**

Unknown directive type "c:function".

```
.. c:function:: int uv_fs_readlink(uv_loop_t* loop, uv_fs_t* req, const char* path, uv_fs_cb cb)

Equivalent to :man:`readlink(2)`.
The resulting string is stored in `req->ptr`.
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]fs.rst, line 442)**

Unknown directive type "c:function".

```
.. c:function:: int uv_fs_realpath(uv_loop_t* loop, uv_fs_t* req, const char* path, uv_fs_cb cb)

Equivalent to :man:`realpath(3)` on Unix. Windows uses `GetFinalPathNameByHandle` <https://docs.microsoft.com/en-us/windows/win32/api/shellapi/nf-shellapi-getfinalpathnamebyhandle>.
The resulting string is stored in `req->ptr`.

.. warning::
    This function has certain platform-specific caveats that were discovered when used in Node.

    * macOS and other BSDs: this function will fail with UV_ELOOP if more than 32 symlinks are found while resolving the given path. This limit is hardcoded and cannot be sidestepped.
    * Windows: while this function works in the common case, there are a number of corner cases where it doesn't:

        - Paths in ramdisk volumes created by tools which sidestep the Volume Manager (such as ImDisk) cannot be resolved.
        - Inconsistent casing when using drive letters.
        - Resolved path bypasses subst'd drives.

    While this function can still be used, it's not recommended if scenarios such as the above need to be supported.

    The background story and some more details on these issues can be checked here <https://github.com/nodejs/node/issues/7726>_.

.. note::
    This function is not implemented on Windows XP and Windows Server 2003. On these systems, UV_ENOSYS is returned.

.. versionadded:: 1.8.0
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]fs.rst, line 472)**

Unknown directive type "c:function".

```
.. c:function:: int uv_fs_chown(uv_loop_t* loop, uv_fs_t* req, const char* path, uv_uid_t uid, uv_gid_t gid, uv_fs_cb cb)
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]fs.rst, line 473)**

Unknown directive type "c:function".

```
.. c:function:: int uv_fs_fchown(uv_loop_t* loop, uv_fs_t* req, uv_file file, uv_uid_t uid, uv_gid_t gid, uv_fs_cb cb)
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]fs.rst, line 474)**

Unknown directive type "c:function".

```
.. c:function:: int uv_fs_lchown(uv_loop_t* loop, uv_fs_t* req, const char* path, uv_uid_t uid, uv_gid_t gid, uv_fs_cb cb)

Equivalent to :man:`chown(2)`, :man:`fchown(2)` and :man:`lchown(2)` respectively.

.. note::
    These functions are not implemented on Windows.

.. versionchanged:: 1.21.0 implemented uv_fs_lchown
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]fs.rst, line 483)**

Unknown directive type "c:function".

```
.. c:function:: uv_fs_type uv_fs_get_type(const uv_fs_t* req)

Returns `req->fs_type`.

.. versionadded:: 1.19.0
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]fs.rst, line 489)**

Unknown directive type "c:function".

```
.. c:function:: ssize_t uv_fs_get_result(const uv_fs_t* req)

Returns `req->result`.

.. versionadded:: 1.19.0
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]fs.rst, line 495)**

Unknown directive type "c:function".

```
.. c:function:: int uv_fs_get_system_error(const uv_fs_t* req)

Returns the platform specific error code - `GetLastError()` value on Windows and `-(req->result)` on other platforms.

.. versionadded:: 1.38.0
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]fs.rst, line 502)**

Unknown directive type "c:function".

```
.. c:function:: void* uv_fs_get_ptr(const uv_fs_t* req)

Returns `req->ptr`.

.. versionadded:: 1.19.0
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src] fs.rst, line 508)**

Unknown directive type "c:function".

```
.. c:function:: const char* uv_fs_get_path(const uv_fs_t* req)

    Returns `req->path`.

.. versionadded:: 1.19.0
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src] fs.rst, line 514)**

Unknown directive type "c:function".

```
.. c:function:: uv_stat_t* uv_fs_get_statbuf(uv_fs_t* req)

    Returns `&req->statbuf`.

.. versionadded:: 1.19.0
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src] fs.rst, line 520)**

Unknown directive type "seealso".

```
.. seealso:: The :c:type:`uv_req_t` API functions also apply.
```

Helper functions

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src] fs.rst, line 525)**

Unknown directive type "c:function".

```
.. c:function:: uv_os_fd_t uv_get_osfdhandle(int fd)

    For a file descriptor in the C runtime, get the OS-dependent handle.
    On UNIX, returns the ``fd`` intact. On Windows, this calls `_get_osfhandle` <https://docs.microsoft.com/en-us/cpp/c-runtime-library/reference/get-osfhandle>.
    Note that the return value is still owned by the C runtime,
    any attempts to close it or to use it after closing the fd may lead to malfunction.

.. versionadded:: 1.12.0
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src] fs.rst, line 534)**

Unknown directive type "c:function".

```
.. c:function:: int uv_open_osfdhandle(uv_os_fd_t os_fd)

    For a OS-dependent handle, get the file descriptor in the C runtime.
    On UNIX, returns the ``os_fd`` intact. On Windows, this calls `_open_osfhandle` <https://docs.microsoft.com/en-us/cpp/c-runtime-library/reference/open-osfhandle>.
    Note that this consumes the argument, any attempts to close it or to use it
    after closing the return value may lead to malfunction.

.. versionadded:: 1.23.0
```

File open constants

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src] fs.rst, line 546)**

Unknown directive type "cmacro".

```
.. c:macro:: UV_FS_O_APPEND

    The file is opened in append mode. Before each write, the file offset is
    positioned at the end of the file.
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src] fs.rst, line 551)**

Unknown directive type "cmacro".

```
.. c:macro:: UV_FS_O_CREAT

    The file is created if it does not already exist.
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src] fs.rst, line 555)**

Unknown directive type "cmacro".

```
.. c:macro:: UV_FS_O_DIRECT

    File I/O is done directly to and from user-space buffers, which must be
    aligned. Buffer size and address should be a multiple of the physical sector
    size of the block device.

.. note::
    `UV_FS_O_DIRECT` is supported on Linux, and on Windows via
    `FILE_FLAG_NO_BUFFERING` <https://docs.microsoft.com/en-us/windows/win32/fileio/file-buffering>.
    `UV_FS_O_DIRECT` is not supported on macOS.
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src] fs.rst, line 566)**

Unknown directive type "cmacro".

```
.. c:macro:: UV_FS_O_DIRECTORY

    If the path is not a directory, fail the open.

.. note::
    `UV_FS_O_DIRECTORY` is not supported on Windows.
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src] fs.rst, line 573)**



Unknown directive type "cmacro".

```
.. c:macro:: UV_FS_O_DSYNC
```

The file is opened for synchronous I/O. Write operations will complete once all data and a minimum of metadata are flushed to disk.

.. note::

'UV\_FS\_O\_DSYNC' is supported on Windows via  
'FILE\_FLAG\_WRITE\_THROUGH' <<https://docs.microsoft.com/en-us/windows/win32/fileio/file-buffering>>`\_.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]fs.rst, line 582)**

Unknown directive type "cmacro".

```
.. c:macro:: UV_FS_O_EXCL
```

If the 'O\_CREAT' flag is set and the file already exists, fail the open.

.. note::

In general, the behavior of 'O\_EXCL' is undefined if it is used without 'O\_CREAT'. There is one exception: on Linux 2.6 and later, 'O\_EXCL' can be used without 'O\_CREAT' if pathname refers to a block device. If the block device is in use by the system (e.g., mounted), the open will fail with the error 'EBUSY'.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]fs.rst, line 593)**

Unknown directive type "cmacro".

```
.. c:macro:: UV_FS_O_EXLOCK
```

Atomically obtain an exclusive lock.

.. note::

'UV\_FS\_O\_EXLOCK' is only supported on macOS and Windows.

.. versionchanged:: 1.17.0 support is added for Windows.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]fs.rst, line 602)**

Unknown directive type "cmacro".

```
.. c:macro:: UV_FS_O_FILEMAP
```

Use a memory file mapping to access the file. When using this flag, the file cannot be open multiple times concurrently.

.. note::

'UV\_FS\_O\_FILEMAP' is only supported on Windows.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]fs.rst, line 610)**

Unknown directive type "cmacro".

```
.. c:macro:: UV_FS_O_NOATIME
```

Do not update the file access time when the file is read.

.. note::

'UV\_FS\_O\_NOATIME' is not supported on Windows.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]fs.rst, line 617)**

Unknown directive type "cmacro".

```
.. c:macro:: UV_FS_O_NOCTTY
```

If the path identifies a terminal device, opening the path will not cause that terminal to become the controlling terminal for the process (if the process does not already have one).

.. note::

'UV\_FS\_O\_NOCTTY' is not supported on Windows.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]fs.rst, line 626)**

Unknown directive type "cmacro".

```
.. c:macro:: UV_FS_O_NOFOLLOW
```

If the path is a symbolic link, fail the open.

.. note::

'UV\_FS\_O\_NOFOLLOW' is not supported on Windows.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]fs.rst, line 633)**

Unknown directive type "cmacro".

```
.. c:macro:: UV_FS_O_NONBLOCK
```

Open the file in nonblocking mode if possible.

.. note::

'UV\_FS\_O\_NONBLOCK' is not supported on Windows.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]fs.rst, line 640)**

Unknown directive type "cmacro".

```
.. c:macro:: UV_FS_O_RANDOM
```

Access is intended to be random. The system can use this as a hint to optimize file caching.

```
.. note::
    'UV_FS_O_RANDOM' is only supported on Windows via
    'FILE_FLAG_RANDOM_ACCESS' <https://docs.microsoft.com/en-us/windows/win32/api/fileapi/nf-fileapi-createfilea>`_.
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]fs.rst, line 649)**

Unknown directive type "cmacro".

```
.. c:macro:: UV_FS_O_RDONLY

    Open the file for read-only access.
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]fs.rst, line 653)**

Unknown directive type "cmacro".

```
.. c:macro:: UV_FS_O_RDWR

    Open the file for read-write access.
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]fs.rst, line 657)**

Unknown directive type "cmacro".

```
.. c:macro:: UV_FS_O_SEQUENTIAL

    Access is intended to be sequential from beginning to end. The system can
    use this as a hint to optimize file caching.

.. note::
    'UV_FS_O_SEQUENTIAL' is only supported on Windows via
    'FILE_FLAG_SEQUENTIAL_SCAN' <https://docs.microsoft.com/en-us/windows/win32/api/fileapi/nf-fileapi-createfilea>`_.
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]fs.rst, line 666)**

Unknown directive type "cmacro".

```
.. c:macro:: UV_FS_O_SHORT_LIVED

    The file is temporary and should not be flushed to disk if possible.

.. note::
    'UV_FS_O_SHORT_LIVED' is only supported on Windows via
    'FILE_ATTRIBUTE_TEMPORARY' <https://docs.microsoft.com/en-us/windows/win32/api/fileapi/nf-fileapi-createfilea>`_.
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]fs.rst, line 674)**

Unknown directive type "cmacro".

```
.. c:macro:: UV_FS_O_SYMLINK

    Open the symbolic link itself rather than the resource it points to.
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]fs.rst, line 678)**

Unknown directive type "cmacro".

```
.. c:macro:: UV_FS_O_SYNC

    The file is opened for synchronous I/O. Write operations will complete once
    all data and all metadata are flushed to disk.

.. note::
    'UV_FS_O_SYNC' is supported on Windows via
    'FILE_FLAG_WRITE_THROUGH' <https://docs.microsoft.com/en-us/windows/win32/fileio/file-buffering>`_.
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]fs.rst, line 687)**

Unknown directive type "cmacro".

```
.. c:macro:: UV_FS_O_TEMPORARY

    The file is temporary and should not be flushed to disk if possible.

.. note::
    'UV_FS_O_TEMPORARY' is only supported on Windows via
    'FILE_ATTRIBUTE_TEMPORARY' <https://docs.microsoft.com/en-us/windows/win32/api/fileapi/nf-fileapi-createfilea>`_.
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]fs.rst, line 695)**

Unknown directive type "cmacro".

```
.. c:macro:: UV_FS_O_TRUNC

    If the file exists and is a regular file, and the file is opened
    successfully for write access, its length shall be truncated to zero.
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\[node-master] [deps] [uv] [docs] [src]fs.rst, line 700)**

Unknown directive type "cmacro".

```
.. c:macro:: UV_FS_O_WRONLY

    Open the file for write-only access.
```