

readdir_r() – Read an entry from a directory

Last Updated: 2021-03-03

Standards

Standards / Extensions	C or C++	Dependencies
Single UNIX Specification, Version 2	both	OS/390® V2R8
Single UNIX Specification, Version 3		

>

Format

```
#define _XOPEN_SOURCE 500
#include <dirent.h>

int readdir_r(DIR *__restrict__dir, struct dirent *__restrict__entry,
              struct dirent **__restrict__result);
```

General description

The `readdir_r()` function initializes the `dirent` structure referenced by *entry* to represent the directory entry at the current position in the directory stream referred to by *dir*, stores a pointer to this structure at the location referenced by *result*, and positions the directory stream at the next entry.

The storage pointed to by *entry* will be large enough for a `dirent` with an array of `char d_name` member containing at least `NAME_MAX+1` elements.

On successful return, the pointer returned at **result* will have the same value as the argument *entry*. Upon reaching the end of the directory stream, this pointer will have the value `NULL`.

The `readdir_r()` function will not return directory entries containing empty names. It is unspecified whether entries are returned for dot or dot-dot.

If a file is removed from or added to the directory after the most recent call to `opendir()` or `rewinddir()`, whether a subsequent call to `readdir_r()` returns an entry for that file is unspecified.

>

The `readdir_r()` function may buffer several directory entries per actual read operation. The `readdir_r()` function marks for update the `st_atime` field of the directory each time the directory is actually read.

Applications wishing to check for error situations should set `errno` to 0 before calling `readdir_r()`. If `errno` is set to non-zero on return, an error occurred.

Returned value

If successful, `readdir_r()` returns 0.

If unsuccessful, `readdir_r()` sets `errno` to one of the following values:

Error Code

Description

EBADF

dir does not refer to an open directory stream.

Related information

- [dirent.h](#) – POSIX directory access
- [stdio.h](#) – Standard input and output
- [sys/types.h](#) – typedef symbols and structures
- [closedir\(\)](#) – Close a directory
- [opendir\(\)](#) – Open a directory
- [__opendir2\(\)](#) – Open a directory
- [readdir\(\)](#) – Read an entry from a directory
- [__readdir2\(\)](#), [__readdir2_64\(\)](#) – Read directory entry and get file information
- [rewinddir\(\)](#) – Reposition a directory stream to the beginning
- [seekdir\(\)](#) – Set position of directory stream
- [telldir\(\)](#) – Current location of directory stream

Parent topic:

→ [Library functions](#)

[Previous](#)

`__readdir2()`, `__readdir2_64()` – Read directory entry
and get file information

[Next](#)

`readlink()` – Read the value of a symbolic link

>