

### std::filesystem::path

```
std::filesystem::path my_path =
"/some/directory";
std::filesystem::path
sub_path = my_path / " sub dir -
ect ory "; // appends to path
std::filesystem::path
file_path = my_path / " fil -
e.t xt";
```

Represents a filesystem path in a cross-platform way, handling differences between operating systems automatically.

Can be constructed from a string and supports path operations like appending with `.`

Part of the `std::filesystem` library introduced in C++17.

### std::filesystem::remove\_all()

```
std::filesystem::remove_all(directory_path)
// throws on error
std::filesystem::remove_all(directory_path, ec); // stores error in ec, no
throw
```

Removes a file OR a directory plus all of its contents recursively.

Returns the number of files and directories removed.

Accepts an optional `std::error_code` parameter to handle errors without throwing exceptions.

### std::filesystem::exists

```
std::filesystem::exists("/tmp/my_file.txt") // true if file exists
std::filesystem::exists("/tmp/my_directory") // true if directory exists
std::filesystem::exists("/tmp/missing") // false if nothing there
```

Checks whether a file or directory exists at the given path.

Returns true if something exists at that path, false if it doesn't.

### std::filesystem::permissions (cont)

```
> std::filesystem::perms::others_read | //
others can read
std::filesystem::perms::others_exec //
others can execute
);
```

Sets the access permissions on a file or directory.

Permissions control who can read, write, or execute a file.

Uses `std::filesystem::perms` flags combined with `|` to set multiple permissions at once.

### std::filesystem::create\_directories()

```
std::filesystem::create_directories("/tmp/my_dir/sub_dir/another");
// creates all three directories if they don't exist
std::filesystem::create_directory(directory_path)
// only creates my_dir, fails if /tmp doesn't exist
```

Creates a directory and any missing parent directories in the path.

Does nothing if the directory already exists.

Use `create_directory()` instead if you only need to create a single directory with an existing parent.

### std::filesystem::temp\_directory\_path()

```
std::filesystem::path temp_dir =
std::filesystem::temp_directory_path();
// temp_dir == "/tmp" on Linux
// commonly used to build a unique temp
path
std::filesystem::path my_temp =
std::filesystem::temp_directory_path() / " my_t em p_d ir";
// my_temp == "/tmp/my_t em p_d ir"
```

Returns the path to the system's temporary directory.

On Linux this is typically `/tmp`, on Windows it's usually `C:\Users\username\AppData\Local\Temp`.

Useful for creating temporary files and directories during tests that won't interfere with the real filesystem.

### std::filesystem::path::string()

```
std::filesystem::path my_path =
"/tmp/my_directory";
std::string path_string =
my_path.string(); // "/tmp -
p/m y_d ire cto ry"
// commonly needed when passing
to C functions
setenv("MY_VAR", my_path.string().c_str(), 1);
```

Converts a `std::filesystem::path` to a `std::string`.

Necessary when passing a path to functions that expect a plain string rather than a `fs::path` object.

Returns the path as a native string representation for the current operating system.

Commonly used in tests to verify files were created or deleted.

### std::filesystem::permissions

```
std::filesystem::permissions(  
    file_path,  
    std::filesystem::permissions::owner_all | //  
    owner can read, write, execute  
    std::filesystem::permissions::group_read | //  
    group can read  
    std::filesystem::permissions::group_exec | //  
    group can execute
```



By **blakecromar**

[cheatography.com/blakecromar/](https://cheatography.com/blakecromar/)

Not published yet.

Last updated 29th June, 2026.

Page 1 of 2.

Sponsored by **Readable.com**

Measure your website readability!

<https://readable.com>