Cheatography

C++ (std::filesystem) Cheat Sheet

by blakecromar via cheatography.com/218070/cs/48229/

std::filesystem::path

```
std::filesystem::path my_path =
"/some/directory";
std::f ile sys tem ::path
sub_path = my_path / " sub dir -
ect ory "; // appends to path
std::f ile sys tem ::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::f ile sys tem ::r emo ve_ all (dir ect
tor y_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::f ile sys tem ::e xis ts( " /tm p/m
y_d ire cto ry"); // true if directory
exists
std::f ile sys tem ::e xis ts( " /tm p/m
iss ing "); // 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 direct ories if they don't exist
std::f ile sys tem ::c rea te_ dir ect ori es(
// 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 == " /tm p" on Linux
// commonly used to build a unique temp
path
std::f ile sys tem ::path my_temp =
std::f ile sys tem ::t emp _di rec -
tor y_p ath() / " my_ tem p_d ir";
// my_temp == " /tm p/m y_t emp _di r"
```

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::s tring path_s tring =
my_pat h.s tri ng(); // " /tm -
p/m y_d ire cto ry"
// commonly needed when passing
to C functions
setenv ("MY _VA R", my_pat h.s -
tri ng( ).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(
        fil e_path,
        std ::f ile sys -
tem ::p erm s:: own er_all | //
owner can read, write, execute
        std ::f ile sys -
tem ::p erm s:: gro up_read | //
group can read
        std ::f ile sys -
tem ::p erm s:: gro up_exec | //
group can execute
```

By **blakecromar**

cheatography.com/blakecromar/

Not published yet.
Last updated 29th June, 2026.