

Doc.No.: D2854R0 DRAFT 1

Project: Programming Language - C++ (WG21)

Audience: LEWGI

Author: Andrew Tomazos <andrewtomazos@gmail.com>

Date: 2023-04-27

Proposal of `std::filesystem::program_path()`

Summary

We propose adding the following two functions to `std::filesystem`:

```
namespace std::filesystem {  
    path program_path();  
    path program_path(error_code& ec);  
}
```

They return a `std::filesystem::path` to the currently running program.

Motivation

It is useful to be able to get the program path in order to:

- 1) Log the program name, particularly in multi-program environments that are outputting log messages to a collated log file or stream - to distinguish log messages from one program from another. For example, glibc C assert outputs the program name as a prefix of its error message. SG21 wants this functionality to do a similar thing with contract violation error messages.
- 2) Find the parent directory of the current program in order to access other files that are deployed at relative paths to the program

Implementation

A similar function has been implemented in numerous C++ frameworks. For example, see `QCoreApplication::applicationFilePath()` (`src/corelib/kernel/qcoreapplication.cpp`)

Wording

[fs.filesystem.syn]

```
namespace std::filesystem {  
  
    /*...*/  
  
    path current_path();  
    path current_path(error_code& ec);  
    void current_path(const path& p);  
    void current_path(const path& p, error_code& ec) noexcept;  
  
    path program_path();  
    path program_path(error_code& ec);  
  
    /*...*/  
}
```

[fs.op.program.path]

```
path filesystem::program_path();  
path filesystem::program_path(error_code& ec);
```

Returns: The absolute path of the currently running program, whose pathname is in native format.

The signature with argument `ec` returns `path()` if an error occurs.

Throws: As specified in [fs.err.report].