fs::current_thread_path()

Document #: DXXXXR0 Date: 2018-02-21

Project: Programming Language C++ Audience: Library Evolution Group

Reply-to: Egor Pugin <egor.pugin@gmail.com>

1 Revision history

• R0 – Initial draft

2 Abstract

This paper proposes an addition of a function «std::filesystem::current_thread_path()» to filesystem library.

```
namespace fs = std::filesystem;
fs::current_path();  // existing, returns process-wide working directory
fs::current_thread path(); // proposed, returns thread-local working directory
```

3 Motivation

When application performs extensively on filesystem in multithreaded environment, fs::current_directory() often is not enough. Programmers have to implement some thread's current working directory via: 1) global thread local variable, 2) a function with thread local variable or 3) local/member variable that is passed down to users.

Implementation of this facility in many cases is more or less common, so it can be considered as an addition to Standard Library.

4 Possible implementation

fs::current_thread_path() gets it initial value from fs::current_path() or user-provided value on the first call. Later, it can be retrieved or changed in the same way.

4.1 Declaration

```
namespace std::filesystem
    path current_thread_path();
    path current_thread_path(std::error_code& ec);
    void current_thread_path(const path& p);
    void current_thread_path(const path& p, std::error_code& ec) noexcept;
}
4.2
     Definition
namespace std::filesystem
{
    path __current_thread_path(const path &p = path(), std::error_code* ec = nullptr)
        thread_local auto thread_working_dir = ec ? current_path(*ec) : current_path();
        if (p.empty())
            return thread_working_dir;
        return thread_working_dir = fs::absolute(p);
    }
    path current_thread_path()
        return __current_thread_path();
    }
    path current_thread_path(std::error_code& ec)
    {
        return __current_thread_path(path(), &ec);
    }
    void current_thread_path(const path& p)
    {
        __current_thread_path(p);
    }
    void current_thread_path(const path% p, std::error_code% ec) noexcept
        __current_thread_path(p, &ec);
    }
```

}

4.3 Other variants

std::this_thread::path()?

5 Proposed wording

TBD

6 References

[N4727] Richard Smith, Working Draft, Standard for Programming Language C++ http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2018/n4727.pdf