
Subject: Re: How to mark `std::array<T, N>` moveable if only T is moveable

Posted by [Lance](#) on Fri, 19 Jul 2024 11:52:06 GMT

[View Forum Message](#) <> [Reply to Message](#)

mirek wrote on Fri, 19 July 2024 07:11 Same semantics does not cover U++ moveable. Now I agree it is unlikely, but so I thought about `std::string` until I met implementation that is not U++ moveable...

I looked into `std::basic_string` in this post, message #59178.

Quote:

Story of `std::basic_string` (GLIBCXX implementation)

It's surprising that a `basic_string<ch>` would cause trouble (core dump etc) when treated as raw bytes. Digging into its implementation (in `<bits/basic_string.h>`), we have the data members

....

On the same post, I proposed to use class traits to handle Moveable. There are details I could not iron out. Also, it might not be compatible with very old c++ because I have an added goal to allow non-trivially relocatable class objects, which can be corrected pre- or post relocation, be housed in `std::vector` or `upp::Vector`.