

Constexpr in `std::pointer_traits`

Document #: D1006R0
Date: 2018-03-18
Project: Programming Language C++
Audience: LEWG
Reply-to: Louis Dionne <ldionne.2@gmail.com>

1 Abstract

As part of the `constexpr` reflection effort, and in particular making `std::vector` `constexpr`, we need to make `std::pointer_traits` `constexpr` (it is used in the implementation).

2 Proposed wording

This wording is based on the working draft [N4727]. Change in [pointer.traits] 23.10.3/1:

```
namespace std {
    template<class Ptr> struct pointer_traits {
        using pointer          = Ptr;
        using element_type     = see below;
        using difference_type  = see below;

        template<class U> using rebind = see below;

        static constexpr pointer pointer_to(see below r);
    };

    template<class T> struct pointer_traits<T*> {
        using pointer          = T*;
        using element_type     = T;
        using difference_type  = ptrdiff_t;

        template<class U> using rebind = U*;

        static constexpr pointer pointer_to(see below r) noexcept;
    };
}
```

3 References

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