

basic_istream_view::iterator should not be copyable

Document #: D1638R0
Date: 2019-04-23
Project: Programming Language C++
Audience: LEWG, LWG
Reply-to: Corentin Jabot <corentin.jabot@gmail.com>
Christopher Di Bella <cjdb.ns@gmail.com>

Proposed change

In [P1027] we discussed why iterators over a single-pass resource such as a stream should not be copyable. `basic_istream_view::iterator` proposed by [P1035] is one such iterator (and the only one proposed for C++20), we therefore propose to make it non-copyable.

Applicability

This paper depends on both [P1027] and [P1035] being accepted by LWG. They have been accepted by LEWG in Kona 2019.

Wording

The wording changes are to be applied on top of the wording changes proposed by [P1035]

◆ **Class template `basic_istream_view::iterator`** **[range.istream.iterator]**

```
namespace std::ranges {  
template<class Val, class CharT, class Traits>  
class basic_istream_view<Val, CharT, Traits>::iterator { // exposition only  
    public:  
        using iterator_category = input_iterator_tag;  
        using difference_type = ptrdiff_t;  
        using value_type = Val;  
  
        iterator() = default;  
        constexpr explicit iterator(basic_istream_view& parent) noexcept;  
  
        iterator(const iterator&) = delete;  
        constexpr iterator(iterator&&) noexcept = default;  
  
        iterator& operator=(const iterator&) = delete;  
};
```

```

constexpr iterator& operator=(iterator&&) noexcept = default;

iterator& operator++();
void operator++(int);

Val& operator*() const;

friend bool operator==( iterator x, default_sentinel );
friend bool operator==( default_sentinel y, iterator x );
friend bool operator!=( iterator x, default_sentinel y );
friend bool operator!=( default_sentinel y, iterator x );
private:
    basic_istream_view<Val, CharT, Traits>* parent_ = nullptr; // exposition only
};

```

References

- [P1027] Corentin Jabot *Movability of single-pass iterators* <https://wg21.link/P1027>
- [P1035] Christopher Di Bella *Input range adaptors*
<https://wg21.link/P1035>