A SFINAE-Friendly `std::iterator_traits`, v3

Contents

Abstract

This paper provides wording to reformulate the specification of `iterator_traits` so as to avoid a hard error when its template argument does not have the member types expected of a non-pointer iterator, and thus to make the trait conveniently usable in a SFINAE context.

1 Proposed wording[1]

> Replace [iterator.traits], paragraph 2, with the following text:

2 The template `iterator_traits<Iterator>` shall have the following as publicly accessible members, and have no other members, if and only if `Iterator` has valid [temp.deduct] member types `difference_type`, `value_type`, `pointer`, `reference`, and `iterator_category`; otherwise, the template shall have no members:

```
typedef typename Iterator::difference_type   difference_type;
typedef typename Iterator::value_type        value_type;
typedef typename Iterator::pointer           pointer;
typedef typename Iterator::reference         reference;
typedef typename Iterator::iterator_category iterator_category;
```

2 Feature-testing macro

For the purposes of SG10, we recommend the macro name `__cpp_lib_iterator_traits_sfinae`.

3 Bibliography

[N3797] Stefanus Du Toit: "Working Draft, Standard for Programming Language C++." ISO/IEC JTC1/ SC22/WG21 document N3797 (post-Chicago mailing), 2013-10-13. http://www.open-std.org/ jtc1/sc22/wg21/docs/papers/2013/n3797.pdf.

---

[1]All proposed additions and ~~deletions~~ are relative to the post-Chicago Working Draft [N3797]. Editorial notes are displayed against a gray background.

4 Document history

Version	Date	Changes
1	2014-01-01	• Published as N3844.
2	2014-02-10	• Excised minor coding relics from legacy implementation. • Fixed copy-paste typo/thinko. • Provided positively-phrased alternative proposed wording. • Described Hinnant's idea: .... • Published as N3909.
3	2014-02-14	• Rephrased the proposed wording per LWG guidance. • Excised everything unrelated to the proposed wording. • Published as N3923.