

---

Subject: Re: 2022(?).2 beta

Posted by [mirek](#) on Mon, 19 Dec 2022 09:08:56 GMT

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

---

Lance wrote on Mon, 19 December 2022 00:41: Hello Mirek and Klugier:

The following code is an excerpt from /usr/include/c++/11/bits/stl\_vector.h

```
iterator
begin() _GLIBCXX_NOEXCEPT
{ return iterator(this->_M_impl._M_start); }

/**
 * Returns a read-only (constant) iterator that points to the
 * first element in the %vector. Iteration is done in ordinary
 * element order.
 */
const_iterator
begin() const _GLIBCXX_NOEXCEPT
{ return const_iterator(this->_M_impl._M_start); }

/**
 * Returns a read/write iterator that points one past the last
 * element in the %vector. Iteration is done in ordinary
 * element order.
 */
iterator
end() _GLIBCXX_NOEXCEPT
{ return iterator(this->_M_impl._M_finish); }
```

Without looking into the definition of `_GLIBCXX_NOEXCEPT`, most experienced c++ users(, all participants of this thread for sure,) can tell that it will expand to `noexcept` when the `-std` version supports it and vanishes otherwise.

It might not be pleasant or pretty, but it certainly works and can be argued as the most reasonable solution in this particular situation.

It's a common problem that libraries with some history need to support different versions; maintaining backward compatibility should not mean stay backward. U++ necessarily has done similar thing for similar purposes, I believe.

Why is it so hard to swallow in this particular case?

BR,  
Lance

Uhm, I guess we are presented with 2 more or less equivalently ugly options here. One of them requires a significant amount of work....

---