

C++ Club Meeting Notes

Gleb Dolgich

2018-06-21

Rapperswil trip reports

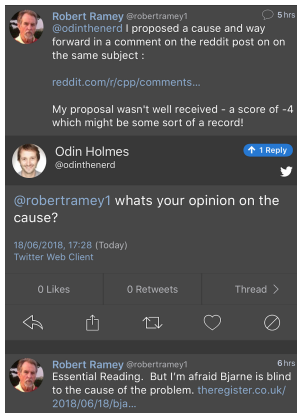
- ▶ Botond Ballo
- ▶ CppCast

The Register interview

- ▶ The Reg
- ▶ Reddit thread

Bjarne Stroustrup: Remember the Vasa

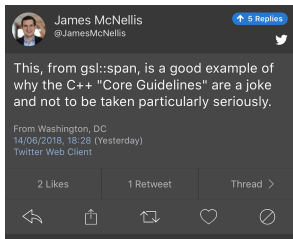
- ▶ P0977r0
 - ▶ Reddit thread
 - ▶ Robert Ramey's reply



Louis Dionne goes to Apple

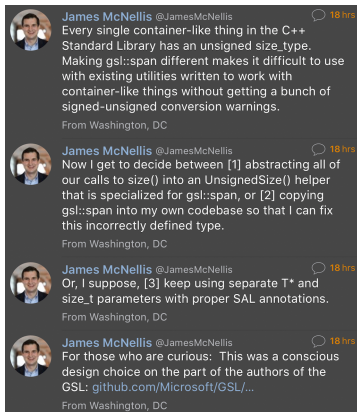


Twitter: C++ Core Guidelines and Microsoft's James McNellis (1)



```
363 // [span], class template span
364 template <class ElementType, std::ptrdiff_t Extent>
365 class span
366 {
367 public:
368     // constants and types
369     using element_type = ElementType;
370     using value_type = std::remove_cv_t<ElementType>;
371     using index_type = std::ptrdiff_t;
372     using pointer = element_type*;
373     using reference = element_type&;
374
375     using iterator = details::span_iterator<span<ElementType, Extent>, false>;
376     using const_iterator = details::span_iterator<span<ElementType, Extent>, true>;
377     using reverse_iterator = std::reverse_iterator<iterator>;
378     using const_reverse_iterator = std::reverse_iterator<const_iterator>;
379
380     using size_type = index_type;
```

Twitter: C++ Core Guidelines and Microsoft's James McNellis (2)

A screenshot of a Twitter thread by James McNellis (@JamesMcNellis). The thread consists of four tweets. Each tweet includes a circular profile picture of James McNellis, his name and handle, a timestamp of '18 hrs', and a location of 'From Washington, DC'. The tweets discuss C++ container-like things, the gsl::span type, and the UnsignedSize() helper.

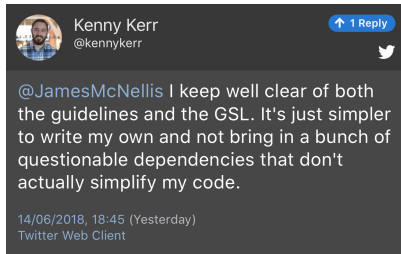
James McNellis @JamesMcNellis 18 hrs
Every single container-like thing in the C++ Standard Library has an unsigned `size_type`. Making `gsl::span` different makes it difficult to use with existing utilities written to work with container-like things without getting a bunch of signed-unsigned conversion warnings.
From Washington, DC

James McNellis @JamesMcNellis 18 hrs
Now I get to decide between [1] abstracting all of our calls to `size()` into an `UnsignedSize()` helper that is specialized for `gsl::span`, or [2] copying `gsl::span` into my own codebase so that I can fix this incorrectly defined type.
From Washington, DC

James McNellis @JamesMcNellis 18 hrs
Or, I suppose, [3] keep using separate `T*` and `size_t` parameters with proper SAL annotations.
From Washington, DC

James McNellis @JamesMcNellis 18 hrs
For those who are curious: This was a conscious design choice on the part of the authors of the GSL: [github.com/Microsoft/GSL/...](https://github.com/Microsoft/GSL/)
From Washington, DC

Twitter: C++ Core Guidelines and Microsoft's Kenny Kerr



Unions and type punning in C++

```
1 union CurrencyCode {  
2     char code[4];  
3     unsigned int hash;  
4 }
```

- ▶ C++ Core Guidelines: Don't use unions for type punning
- ▶ SO: Accessing inactive union member
- ▶ John Regehr: Type Punning, Strict Aliasing, and Optimization
- ▶ C++ standard committee mailing list: Unions and UB

Builder Pattern with Fluent API

Post

```
1 class Email {  
2 public:  
3     ...  
4     static EmailBuilder make();  
5 };  
6  
7 class EmailBuilder {  
8     ...  
9     operator Email&&() { return std::move(m_email); }  
10 private:  
11     Email m_email;  
12 }  
13  
14 EmailBuilder Email::make() { return EmailBuilder(); }
```

Builder Pattern with Fluent API (cont.)

```
1 int main() {  
2     Email mail = Email::make().from("me@mail.com")  
3         .to("you@mail.com")  
4         .subject("C++ builders")  
5         .body("I like this API");  
6 }
```

Before/after tables (“Tony Tables”) for C++17 and C++20 by Tony van Eerd

- ▶ C++20
- ▶ C++17

► [Playlist](#)



mcc

@mcclure111

2 Replies, 50 Quotes



In C++ we don't say "Missing asterisk" we say "error C2664: 'void std::vector<block,std::allocator<_Ty>>::push_back(const block &)': cannot convert argument 1 from 'std::_Vector_iterator<std::_Vector_val<std::_Simple_types<block>>' to 'block &&'" and i think that's beautiful

01/06/2018, 21:30 (Yesterday)
Twitter Web Client

Retweeted by @sebastiangood
02/06/2018, 18:56

4920 Likes

1705 Retweets

Thread >



رمزي ناص...

@ra

↑ 7 Replies, 68 Quotes



if you ever feel like a bad coder dont forget that the software engineers behind the bourne shell needed a way to end their `case` statement and one of them said "what about `esac`" and another said "wtf is esac" and the first one said "case spelled backwards" and they shipped it

22/05/2018, 22:49 (Tuesday)

Twitter Web Client

Retweeted by @aaronvegh

23/05/2018, 14:53

3519 Likes

1074 Retweets

Thread >