

C++ Club UK

Gleb Dolgich

2019-08-08

What Happened to C++20 Contracts?

Nathan Myers: [https:](https://www.reddit.com/r/cpp/comments/cmk7ek/what_happened_to_c20_contracts/)

[//www.reddit.com/r/cpp/comments/cmk7ek/what_happened_to_c20_contracts/](https://www.reddit.com/r/cpp/comments/cmk7ek/what_happened_to_c20_contracts/)

This was the first time, in the (exactly) 30 years since ISO was first asked to form a Working Group to standardize C++, that the committee has removed from its Working Draft a major feature, for no expressible technical reason.

Almost immediately after the feature was voted in, one party to the original agreement -- authors of the rejected 2012 design -- began to post a bewildering variety of proposals for radical changes to the design, promoting them by encouraging confusion about consequences of the agreed-upon design.

What Happened to C++20 Contracts? (cont.)

One of the proposals, not seen before the day it was presented, seemed to offer that simplicity, and the group seized upon it, voting for it by a margin of 3 to 1. It was opposed by four of the five participants of the original design group, because it was fatally flawed: in use, programmers would need to define preprocessor macros, and put calls to those in their code instead of the core-language syntax defined. It would breed "macro hell".

On top of its inherent flaws, it amounted to a radical redesign from what was originally accepted by the full committee.

The immediate, predictable effect was panic. <...>

Two days later, the same Evolution Working Group voted to remove the feature entirely.

My word, what a thread.

Bryce Lebach on the C++ Committee



Bryce Lebach @blelbach

50% of C++ programmers: the committee is moving too fast!

50% of C++ programmers: the committee is moving too slow!

[#Cpp20](#)

18h • 22/07/2019 • 21:31 •

Twitter: who does what in the C++ Committee



Jorg Brown @jorgbrown

Overhead at the C++ Standards meeting:

"Libraries does naming; Core does punctuation"

"Evolution imagines that they provide solutions to real problems; Core provides real solutions to imaginary problems"

2d • 18/07/2019 • 18:44



Improved Linker Fundamentals in Visual Studio 2019

<https://devblogs.microsoft.com/cppblog/improved-linker-fundamentals-in-visual-studio-2019/>

https://www.reddit.com/r/cpp/comments/chqj93/visual_c_linking_speedup_by_23x_in_latest_visual/

<https://blog.qt.io/blog/2019/08/07/technical-vision-qt-6/>

https://www.reddit.com/r/cpp/comments/cn52ql/technical_vision_for_qt_6/

- ▶ C++17
- ▶ Strongly-typed QML
- ▶ QML to C++ compilation, JavaScript optional
- ▶ CMake as the build system
- ▶ Next-generation graphics support

How do C++ developers manage dependencies?

https://www.reddit.com/r/cpp/comments/c6l3eg/how_do_c_developers_manage_dependencies/

Through much pain and anguish.

Scott Meyers' TD trick

https://www.reddit.com/r/cpp/comments/c6vnb3/just_started_learning_c_coming_from_python_and/eshq8vb?utm_source=share&utm_medium=web2x

```
1 | template <typename T> struct TD; // no definition
```

Now you write something like `TD<decltype(thing)>` and the error message tells you the type of thing (as deduced by `decltype`, of course, but in this case that's probably what you want).

Just started learning C++ coming from Python

https://www.reddit.com/r/cpp/comments/c6vnb3/just_started_learning_c_coming_from_python_and/

The new GCC compiler with colour highlighting is a little bit better at pointing out errors. It's generally quite helpful for pure C/C++ until you make an error with the standard library and you get 200 lines about std:: whatever<random characters>

In C++ a trick I always use when the error message is massive is to just focus on the first error.

Use **constexpr** for faster, smaller, and safer code

<https://blog.trailofbits.com/2019/06/27/use-constexpr-for-faster-smaller-and-safer-code/>

https://www.reddit.com/r/cpp/comments/c646ng/use_constexpr_for_faster_smaller_and_safer_code/

<https://github.com/trailofbits/constexpr-everything> (Apache 2.0)

A closer look at **bake**: a tool that makes building C/C++ code effortless

<https://medium.com/@cortoproject/a-closer-look-at-bake-a-tool-that-makes-building-c-c-code-effortless-b2e0409fad8f>

- ▶ https://www.reddit.com/r/C_Programming/comments/a85f6w/meet_bake_a_new_build_system_package_manager_for/
- ▶ https://www.reddit.com/r/cpp/comments/a8d7ny/meet_bake_a_new_build_system_package_manager_for/
- ▶ <https://news.ycombinator.com/item?id=18787777>

<https://github.com/SanderMertens/bake> (GPLv3)

A cargo-like buildsystem and package manager for C/C++

Magic.

A virtual universe which lets you explore, analyze and present huge planetary datasets and large simulation data in real-time.

Uses C++17 and OpenGL.

<https://github.com/cosmoscout/cosmoscout-vr> (MIT) Copyright (c) 2019 German Aerospace Center (DLR)

https://www.reddit.com/r/cpp/comments/cn657d/the_german_center_for_aerospace_dlr_just_open/



99 little bugs in the code.
99 little bugs in the code.
Take one down, patch it around.
127 little bugs in the code...