

20 July 2017

Blog post

- ▶ 120+ people, 9 national bodies
- ▶ 1st meeting for C++20
- ▶ Concepts TS merged into draft C++20!
 - ▶ no “introducer syntax”
 - ▶ no “terse/natural syntax” for now (no consensus)
 - ▶ removed the need to write “bool”
 - ▶ removed function concepts “until we see a need to overload concepts”

- ▶ Allow lambda capture [=, this]
- ▶ Add designated initializers. Draft C++20 now allows code like:

```
1 struct A { int x; int y; int z; }; A b{.x = 1, .z = 2};
```

- ▶ Allow template parameter lists on lambdas:

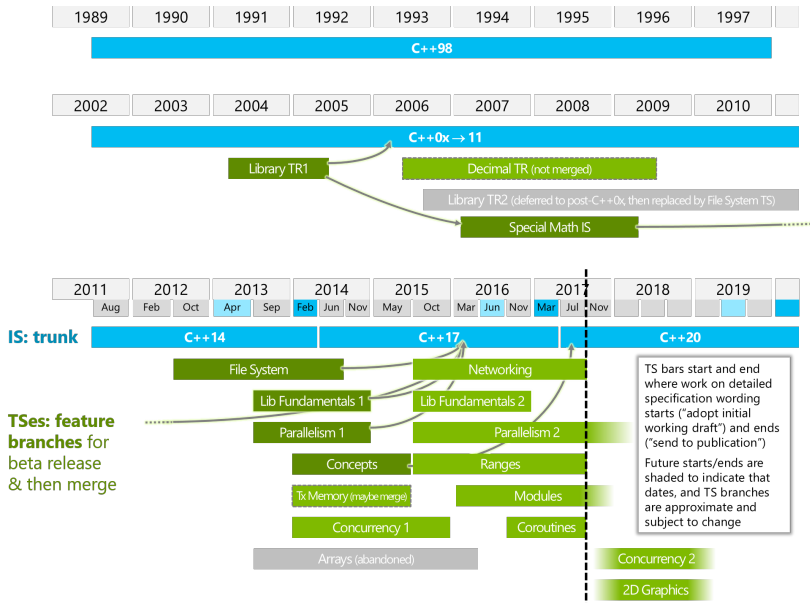
```
1 auto f = [<typename T>(std::vector<T> vector){/* ... */};
```

- ▶ Coroutines TS
- ▶ Ranges TS
- ▶ Networking TS
- ▶ Modules PDTS (no macros)

Not yet:

- ▶ Concurrency TS (published, parts to be merged in the next meeting)

C++ Feature Timeline



Toronto report by Guy Davidson

Blog post

I've read many trip reports but they tend to speak of the results more than the process. This post will describe more of the process.

- ▶ There is a wiki. Make sure you retrieve the URL and login details from another delegate.
- ▶ All the groups are different in attendance, character and operation. Try them out.
- ▶ wg21.link is your most useful tool.
- ▶ Volunteer for things. History is made by those who turn up.
- ▶ Everyone is a volunteer. Nobody is out to impress anybody.
- ▶ If you don't know something, take this rare opportunity to learn from the designers.
- ▶ Sleep well, eat properly and stay hydrated. It's easy to forget when you're concentrating on other things.

Reddit thread

Clang added option `-std=c++2a` for C++20.

Deduction guides

- ▶ [Paper p0702r0](#)
- ▶ [Reddit comment](#)

Current:

```
1 tuple t{tuple{1, 2}}; // Deduces tuple<int, int>
2 vector v{vector{1, 2}}; // Deduces vector<vector<int>>
```

C++20: prefer copy initialization

```
1 vector a {vector {1, 2},
2           vector {3, 4}}; // vector<vector<int>>
3 vector b {vector {1, 2}}; // vector<int>
```

Implementation by Chris Kohlhoff

This implementation is automatically generated from the Asio library. Consequently, it may still be missing small pieces of functionality, or contain artifacts from Asio that are not part of the TS.

In-Flight TS (1)

- ▶ Transactional Memory v1, published
- ▶ Concepts v1, merged into C++20
- ▶ Concurrency v1, published, parts will be merged into C++20 next meeting
- ▶ Library Fundamentals v2, published
- ▶ Ranges v1, published
- ▶ Networking v1, published
- ▶ Coroutines v1, published
- ▶ Modules v1, draft released (PDS)

In-Flight TS (2)

- ▶ Executors v1, in development
- ▶ Parallelism v2, in development
- ▶ Reflection v1, in development
- ▶ Concurrency v2, in early development
- ▶ Parallelism v2, in early development
- ▶ Library Fundamentals v3, in early development
- ▶ Contracts v1
- ▶ 2D Graphics v1, in early development

Blog post

- ▶ 46% don't plan to move to a newer C++ standard.
- ▶ 62% of those using older standards (C++ 98 and C++ 03) are moving to modern C++ standards.
- ▶ 25% of C++ developers are planning to move to C++17.

Blog post

- ▶ Support for C++17
- ▶ Integration with clang-tidy (including Core Guidelines checks)
 - ▶ JetBrains forked LLVM and Clang to integrate with CLion
- ▶ More accurate C++ parser
- ▶ PCH for MSVC
- ▶ Step into disassembly
- ▶ Support for Google Test
- ▶ Improved Find-in-Path command