

WG21 June 2024 Hybrid meeting

Minutes of Meeting

ISO/IEC JTC1 SC22 WG21 4985— 2024-07-09

Nina Dinka Ranns, dinka.ranns_at_gmail.com

Chair: John Spicer

24-29 June 2024, St. Louis, MO, USA

1. Opening activities

John Spicer opens the meeting at 09.00 AM GMT-5.

The wifi information is on the screen.

1.1 Opening comments, welcome from host

John Spicer welcomes the group.

Welcome from the host.

Bill Seymour : Welcome to St Louis. I hope we have a really good meeting. Welcome to the place where the first meeting of what would eventually become IEC was held, all the way back in 1904.

John Spicer : This is no longer a joint meeting with INCITS.

1.2 Meeting guidelines

John Spicer presents.

Please speak into the microphone so people participating over Zoom can hear. Please introduce yourself when speaking.

Meetings are not public, we want everyone to be able to speak freely. Please refrain from live tweeting, blogging, taking photos of other people's screens or recording the meetings. You're allowed to take screenshots of presentations for your personal use.

Agenda is on the wiki.

Every participant is responsible for understanding and abiding by the following:

[The ISO Code of Conduct](#)

[The IEC Code of Conduct](#)

[The WG21 Practices and Procedures, and Code of Conduct](#)

Documents are on the wiki. Please get familiar with them. They also include a description of the process we follow.

You are expected to abide by the rules of the code of conduct of your respective NB.

Nina Ranns : Welcome to yet another WG21 meeting

This is a technical meeting where we're meant to keep our discussions technical. Something we all enjoy, I'm sure. After all, we're all here, despite almost certainly having more traditionally exciting places to be.

But we're here because we have a common goal. We all want to make C++ a better language.

Each one of us will bring ideas and comments based on our own technical experience. And while all our WG21 discussions will be technical, we are all first and foremost human.

We come from different backgrounds, different cultures, and different lived experiences.

As we PREPARE ourselves for working closely on topics we are passionate about, I want to ask you all to take a look around and see the people around you. See the people you will be working with to get to our common goal.

Be aware of them and of your interaction with them.

Understand that what one person considers friendly, another one might find intrusive. What one person finds short and to the point, another may find dismissive and offensive. These are normal situations when a group of individuals with varied backgrounds engages in prolonged conversation. Be mindful of others, but also yourself. If a situation is making you uncomfortable, it is ok to communicate that. Similarly, if someone is communicating their discomfort to you, hear them out. The path towards mutual understanding is communication and respect in that communication. When we interact in our meetings, we should all hold each other in unconditional positive regard. Even in the face of disagreements which can sometimes become passionate; it's what helps a group maintain a sense of belonging.

Colleen Passard gave an excellent introduction at CppCon much like this one. After some conversations in Tokyo, it was suggested that it would be good to have a similar introduction at WG21. We don't have Colleen here, we have me, but I have spoken to Colleen and taken inspiration from her. One thing that Colleen mentions is the notion of rumble, an approach to getting productive interaction in a group like ours. I would like to end by reading a quote by Brene Brown, the author of the idea of rumble.

“A rumble is a discussion, conversation, or meeting defined by a commitment to lean into vulnerability, to stay curious and generous, to stick with the messy middle of problem identification and solving, to take a break and circle back when necessary, to be fearless in owning our parts, and to listen with the same passion with which we want to be heard. More than anything else, when someone says, ‘Let’s rumble,’ it cues me to show up with an open heart and mind so we can serve the work and each other, not our egos.”

Should you at any point find yourselves overwhelmed or needing advice on an uncomfortable situation, do not hesitate to approach a WG21 officer. : Herb, John, or me. If you can't find us, send us an email and we will be in touch. We will hear you out and hopefully find a way

together with you to get you back into the rumble.
Hope you all have a good and productive meeting.

John Spicer: For plenary polls, you have to be in the ISO global directory to vote. One person, one vote. In working groups and study groups everyone can vote. Please refer to the best practices in the WG21 document - e.g. do not vote unless you are familiar with the issue.

Nevin Liber : Attendance sheet is live. If you have registered, it's pre-populated. If you have any issues, find me on mattermost, email, or in person. Link to the attendance sheet is on top of the wiki page, on the reflector, and on mattermost.

John Spicer explains voting procedure for remote and in person attendance.

1.3 Introductions

Introduction of the WG21 officers.
Introduction of admin support roles.
Introduction of the subgroup chairs.
First time attendees introduce themselves.

Herb Sutter polls for NB representation. We have at least 19 NB present at this meeting.

1.4 Agenda review and approval

John Spicer presents the agenda and timings for the week.

The primary goals of this meeting will be work on C++26 features. For more information on the schedule, please see P1000R6

Motion to approve the meeting agenda.
No objections.
Approved.

1.5 Editor's reports, approval/adoption of working drafts

Document	Editor's report	Prospective WD
C++ 26 Working Draft	N4982	N4981

Motion to approve the documents above.
No objections.

Approved.

1.6 Approval of the minutes of the previous meetings

Meeting	Minutes
WG21 Tokyo	N4980
WG21 pre-St. Louis administrative telecon	N4984

Motion to approve the documents above.

No objections.

Approved.

2. Liaison reports, and WG21 study group reports (see pre-meeting WG21 telecon minutes)

No discussion.

3. WG progress reports (Core, Evolution, Library, Library Evolution; see pre-meeting WG21 telecon minutes)

No discussion.

4. New business requiring action by the committee

No new business.

5. Organize working groups and study groups, establish working procedures

Jens Maurer presents room assignments.

Room assignments are on the wiki page. Any questions, please let me know.

If you will be in Wroclaw and can carry a projector, please let me know.

Any issues with remote attending setup, find me.

We will have an evening session on Tuesday talking about Senders and Receivers, with an extended Q&A session.

Thursday evening we have a presentation of the Beaman project.

Nevin Liber : if you need a paper number and you're not set up on the system, please let me know.

6. Subgroup sessions

John Spicer presents. The subgroup chairs must arrange for any proposals to be written up in the form of a motion, and made available by 8:00 PM Friday on the straw polls page together with associated papers. Groups are encouraged to make those papers and polls available as soon as possible during the week so people can have time to review them.

If you have any questions or issues, please bring them up to attention as soon as possible. on reflector, mattermost, or one of the subgroup chairs.

7. Review of the meeting

Reminder: Make sure you have marked the attendance sheet, if you have not already done so.

Subgroup status and progress reports. Presentation and discussion of proposals to be considered for consensus adoption by full WG21.

SG1: Concurrency (Giroux)

We discussed 24 papers. Significantly progressed some of the papers we had been iterating on for many years, in some cases a decade.

Addressed 3 library issues.

Hybrid meeting is still working very well. Most of the time we had ~15 people in the room and another ~7 online.

At the next meeting, the priority will be given to small papers that add to senders and receivers and hope to land in '26.

SG4: Networking (Snyder/Ažman)

SG4 did not meet this week

SG6: Numerics (Kretz/Lippincott/McFarlane)

SG6 met for one day on Monday. We reviewed 5 papers. P3306 "Atomic Read-Modify-Write Improvements" and P3111 "Atomic Reduction Operations" were sent out of SG6.

P3045 "Quantities and units library" needs more time but we're trying to converge on how we want to chunk our workload.

SG6 agrees with the direction P2964 "Allowing user-defined types in `std::simd`" is taking and encouraged further work.

P3161 "Unified integer overflow arithmetic" prompted a discussion about completing the set of functions that was started with saturating functions and P3018 "Low-Level Integer Arithmetic". We hope to see a combined paper in the future.

SG7: Compile-time programming (Dusikova/Vandevoorde)

We saw 7 papers and forwarded 6 of them, most of these are extensions for P2996 Reflection for C++26:

- P3294R0 Code Injection with Token Sequences: Paper was forwarded to EWG.
- P2825R2 `declcall(unevaluated-postfix-expressions)`: Paper was forwarded to EWG.
- P3157R1 Generative Extensions for Reflection: Author was encouraged to work in the direction presented.
- P3289R0 `consteval` blocks: Paper was forwarded to EWG.
- P3273R0 Introspection of Closure Types: Paper was forwarded to EWG.
- P3293R0 splicing a base class subobject: Paper was forwarded to EWG.
- P3295R0 Freestanding `constexpr` containers and `constexpr` exception types: Paper was forwarded to LEWG.

We didn't see paper P2830R4 Standardized `constexpr` Type Ordering on request from the author who expects to present it at the next meeting.

SG9: Ranges (Hollman/Müller)

SG9 met on Thursday and Friday in St. Louis. We discussed two papers about parallel range algorithms jointly with SG1 on Thursday, and we discussed 5 papers on Friday, working on the plan approved by P2760. We also held a joint session with LWG where we discussed three issues. In the end, we forwarded two papers to LEWG (P3137: `views::to_input` and P2848: `std::is_uniqued`) and approved one issue in the joint session with LWG. The results of our polls are on GitHub and the minutes are on the wiki.

SG10: Feature test (Revzin/Wakely)

No report.

SG14: Games & low latency (Wong)

SG14 did not meet this week.

SG15: Tooling (Spencer/Boeckel)

SG15 saw 4 papers and forwarded 2 of them for the Ecosystem IS.

-  [P2656R2](<https://wg21.link/p2656r2>) [C++ Ecosystem International Standard](<https://github.com/cplusplus/papers/issues/1323>): We're planning on a new international standard dedicated to tooling concerns called the Ecosystem IS. Forwarded to the evolution groups.
-  [P3051R1](<https://wg21.link/p3051r1>) [Structured Response Files](<https://github.com/cplusplus/papers/issues/1720>): Provides a portable response file format, and a structured option format. Forwarded to the evolution groups along with the initial Ecosystem IS.
-  [P3286R0](<https://wg21.link/p3286r0>) [Module Metadata Format for Distribution with Pre-Built Libraries](<https://github.com/cplusplus/papers/issues/1939>): Provides a way to find and parse module interface files from 3rd party libraries. We liked it and encouraged further work targeting the Ecosystem IS.
-  [P3267R1](<https://wg21.link/p3267r1>) [C++ contracts implementation strategies](<https://github.com/cplusplus/papers/issues/1923>): We discussed various implementation strategies for contracts with a few Itanium C++ ABI people, and a contracts implementer in the room. We had no tooling related concerns with contracts, but believe that the best ABI would require linker changes to get the best performance.

SG16: Unicode (Honermann/Brett/Downey)

SG16 did not meet.

SG17: EWG Incubator (Keane/Touton)

On Thursday we didn't have a quorum so we met only on Friday with half time allocated than expected. We still got to see 6 out of 10 scheduled papers:

- - P3245R0 Allow `[[nodiscard]]` in type alias declaration: We gave author feedback on preferred design.
- - P3298R0 Implicit user-defined conversion functions as operator.(): Forwarded to EWG.
- - P2952R1 auto & operator=(X&&) = default: Forwarded to EWG.
- - P3312R0 Overload Set Types: We gave the author feedback on design.
- - P3176R0 The Oxford variadic comma: Forwarded to EWG.
- - P3166R0 Static Exception Specifications: We gave encouragement to the author to continue work on the proposed design.
-
- We didn't see papers: P3093R0 Attributes on expressions, P3218R0 const references to constexpr variables, P3259R0 const by default, and P3266R0 non referencable types. We plan to have a telecon soon to see these papers.

SG18: LEWG Incubator (Baker/Liber)

We had 10 attendees on Thursday morning, and 7 attendees on Friday afternoon. We discussed two papers:

- [P3094R2](#): `std::basic_fixed_string` - forwarding to LEWG
- [P3045R1](#): Quantities and units library - making good progress

We plan on continuing via telecons.

SG19: Machine Learning (Wong/Reverdy)

We met this week. We looked at all the graph papers. We have been trying to get the graph library out at every meeting. I estimate it will take another 4-5 online meetings. We looked at two papers on statistics.

We got good feedback from the audience. Statistics are in LEWG. After that we will look at more machine learning things. I'm inviting people to come and help us with that.

SG20: Education (van Winkel)

SG20 did not meet in St. Louis.

SG21: Contracts (Spicer/Doumler)

SG21 met for two days (Wednesday & Thursday) in St. Louis.

We started by discussing the results from EWG's review of P2900 (our Contracts MVP proposal), which happened on Monday & Tuesday, and in which we managed to successfully resolve several contentious design issues.

We then adopted a paper that makes contract assertions observable checkpoints (P3328R0), making contract assertions more robust to undefined behavior.

Following that, we had a productive discussion on contract assertions on function pointers (P3250R0, P3271R0) which will need more work, and an even more productive discussion on contract assertions on virtual functions (P3097R0) which ended in SG21 adopting the design proposed by that paper, which was then also approved by EWG on Friday, thereby plugging the most significant remaining design hole in P2900.

We also discussed some extensions to P2900 aimed at facilitating migration from existing macro-based facilities to contracts (P3290R0, P3311R0). Finally, we discussed a few other papers proposing changes to P2900, rejecting one (P3316R0) and not finishing discussion on two more (P3210R0, P3249R0) because we ran out of time.

We're planning to continue regular telecons in the lead-up to the next WG21 meeting in Wrocław in November. During those telecons we will focus on the remaining issues with P2900: constification, pre/post on function pointers, pre/post on coroutines, and any other papers proposing breaking changes to the MVP.

SG22: C/C++ Liaison (Ranns,Meneide(for WG14))

SG22 has not met this week. We had a few issues forwarded to us. Some are currently discussed on the reflector. We also plan on having a telcom in the next two months to discuss a couple of deprecation papers.

SG23: Safety/Security (Orr/Craig)

SG23 met for a day in St Louis.

We discussed three papers:

P3274R0 "A framework for Profiles development" by Bjarne Stroustrup
We took a poll on the preferred syntax; which showed greater support for the attribute-like syntax.

P3297R0 "C++26 Needs Contract Checking"
by Christian Eltzschig, Mathias Kraus, Ryan McDougall, and Pez Zarifian
We took one poll, "We would prefer to ship a watered down P2900 as described in D3297R1 than nothing in C++26" which had consensus in favour.

P3232R0 "User-defined erroneous behaviour" by Thomas Köppe
We forwarded this to EWG/LEWG

We also had a presentation by Sean Baxter on his implementation of Rust-like borrow checking. We had strong consensus for spending more committee time on borrow checking

We did not discuss P3100R0 "Undefined and erroneous behaviour are contract violations" by Timur Doumler, Gašper Ažman, and Joshua Berne as the authors were busy. You can probably guess why. We plan to discuss this in a future telecon.

ABI Group (Vandevoorde)

No report.

Admin (Liber)

The unofficial attendance as of this writing:

- F2F: 101
- Virtual: 81
- Overlap: 2

The post St. Louis mailing deadline is July 16th.

Evolution (Bastien/Stone/Keane/Dusikova)

Summary of the week :

- [P2900r7 Contracts](#): We spent a day and a half on contracts, and made significant progress towards consensus. There are still points of disagreement, but we have resolved a significant number of them and are hopeful that the next meeting will show yet more increases consensus on the design.
- [P2996R3 — Reflection for C++26](#): moving towards C++26.
- We reviewed 17 core issues and identified authors to write papers to resolve all of them.
- We saw 39 papers, of which the leading papers were:
 -  [P2434R1 Nondeterministic pointer provenance](#): promising way to resolve both issues of provenance and pointer zap.

-  [P1494R3](#) — [Partial program correctness](#): seen as part of contracts, prevents propagating undefined behavior across boundaries.
 -  [P3032R2](#) — [Less transient constexpr allocation](#): moving towards C++26.
 -  [P0876R16](#) — [fiber_context - fibers without scheduler](#): track exceptions on a per-fiber basis rather than leaving it implementation-defined., request implementation experience.
 -  [P3096R1](#) — [Function Parameter Reflection in Reflection for C++26](#): encourage further work.
 -  [P3068R2](#) — [Allowing exception throwing in constant-evaluation](#): moving towards C++26.
 -  [P0963R2](#) — [Structured binding declaration as a condition](#): moving towards C++26.
 -  [P3310R2](#) — [Solving partial ordering issues introduced by P0522R0](#): received support, but CWG sent back.
 -  [P2758R3](#) — [Emitting messages at compile time](#): moving towards C++26.
 -  [P2992R1](#) — [Attribute \[\[discard\("reason"\)\]\]](#): no consensus.
 -  [P2971R2](#) — [Implication for C++](#): no consensus, but feedback given on how to increase consensus.
 -  [P3232R0](#) — [User-defined erroneous behaviour](#): encourage further work.
 -  [P2719R0](#) — [Type-aware allocation and deallocation functions](#): encourage further work.
 -  [P3140R0](#) — [std::int_least128_t](#): encourage further work.
 -  [P3253R0](#) — [Distinguishing between member and free coroutines](#): no consensus.
 -  [P3254R0](#) — [Reserve identifiers preceded by @ for non-ignorable annotation tokens](#): no consensus.
 -  [P2822R1](#) — [Providing user control of associated entities of class types](#): weak consensus, feedback provided.
 -  [P2989R1](#) — [A Simple Approach to Universal Template Parameters](#): encourage further work.
 -  [P3074R3](#) — [trivial union \(was std::uninitialized\)](#): encourage further work.
 -  [P2786R6](#) — [Trivial Relocatability For C++26](#): sent back from CWG to EWG, feedback was given and volunteers identified to resolve open issues.
 -  [P3097R0](#) — [Contracts for C++: Support for Virtual Functions](#): encourage further work.
 -  [P2825R2](#) — [Overload Resolution hook: declcall\(unevaluated-postfix-expression\)](#): encourage further work.
 -  [P3087R0](#) — [Make direct-initialization for enumeration types at least as permissive as direct-list-initialization](#): no consensus.
 -  [P1112R5](#) — [Language support for class layout control](#): no consensus for this specific paper, but consensus was previously expressed to resolve the issue.
 -  [P3177R0](#) — [const prvalues in the conditional operator](#): encourage further work.
- We ran out of time to see 4 papers.
 - 5 papers were without presenter.
 - 3 papers were deferred at the request of the author.

Library Evolution (Levi/Fracassi/Craig)

LEWG met during the full week, and reviewed multiple features for C++26. The main features that captured our time were:

- [P2300R10](#): `std::execution` (forwarded in a previous meeting, LEWG saw related papers)
- [P2996R4](#): Reflection for C++26

“[P2300R10](#): `std::execution`” adding the foundational library concepts for async programming along with an initial set of generic async algorithms.

Additional async programming facilities following this paper are being worked on, and are also targeting C++26; including work on a system execution context/thread-pool, parallel algorithms, concurrent queue supporting both synchronous and async push/pop, and `counting_scope` which lets you join a set of spawned async operations. The paper was already forwarded to the wording group, LWG, which have been wording on it throughout the week (and voted in plenary by the end of the meeting), but there are still design improvements and fixes papers related to P2300 which LEWG spent time on during the week (and will continue to do so during telecons).

“[P2996R4](#): Reflection for C++26” is under review on LEWG. It provides the `std::meta` namespace, which contains library functions to support “reflection” functionality, such as traits-equivalent functions and query functions, as well as functions to construct structures based on information from reflected code.

EWG (the language evolution group) approved the language aspect of the proposal, and LEWG (the standard library evolution group) is in the work of reviewing the library aspects of it.

The full list of papers seen by LEWG is below.

The following papers forwarded from LEWG (to SGs/LWG)

- [P3175R2](#): Reconsidering the `std::execution::on` algorithm
- [P3303R0](#): Fixing Lazy Sender Algorithm Customization
- [P0843R13](#): `inplace_vector` - plenary approved.
- [P3235R3](#): `std::print` more types faster with less memory - plenary approved.
- [P3187R1](#): remove `ensure_started` and `start_detached` from P2300
- [P3309R0](#): `constexpr atomic` and `atomic_ref` - require input from SG22 and approval by electronic poll.
- [P3323R0](#): cv-qualified types in `atomic` and `atomic_ref` - require approval by electronic poll.
- [P2897R1](#): `aligned_accessor`: An `mdspan` accessor expressing pointer overalignment - require approval by electronic poll.
- [P3008R2](#): Atomic floating-point min/max - require approval by electronic poll.

The following paper was not sent to LWG but merged into the Parallelism TS 2:

- [P1928R10](#): Merge data-parallel types from the Parallelism TS 2 - Merged into TS
- [P3287R0](#): Exploration of namespaces for `std::simd`

The following papers need to be seen again by LEWG

-  [P3164R1](#): Improving diagnostics for sender expressions
-  [P1030R6](#): `std::filesystem::path_view`
-  [P3275R0](#): Replace `simd operator[]` with getter and setter functions - or not

-  [P2769R2](#): get_element customization point object
-  [P2626R0](#): charN_t incremental adoption: Casting pointers of UTF character types - got encouragement to solve the issue, language changes will need to be applied by Core before we can see it back.
-  [P3149R5](#): async_scope -- Creating scopes for non-sequential concurrency - design made progress, wording required.
-  [P2996R4](#): Reflection for C++26 - we reviewed:
 - Wording that indicates no guarantees between different versions of the standard in regards to reflected code, and in particular, no guarantees for the standard library reflected implementation details.
 - Three name-returning functions (“name_of”, “qualified_name_of”, “display_name_of”) in a joint session with SG16 (the u8 versions are waiting for SG16’s input and will be reviewed by LEWG).
 - We approved 10 trait-like functions: “is_virtual”, “is_pure_virtual”, “is_override”, “is_deleted”, “is_defaulted”, “is_explicit”, “is_bit_field”, “is_const”, and “is_volatile”, and “is_noexcept”.
 - We gave feedback on the design of bit_offset functions (final design is to be approved by LEWG).
 - We will be continuing the review on P2996 during telecons.
-  [P3068R2](#): Allowing exception throwing in constant-evaluation
-  [P0260R10](#): C++ Concurrent Queues - got a lot of design feedback, and will be seen again after that feedback is applied.
-  [P3325R0](#): A Utility for Creating Execution Environments - design approved, wording review is required.
-  [P2746R5](#): Deprecate and Replace Fenv Rounding Modes
-  [P3299R0](#): Range constructors for std::simd - got design feedback, will be seen by LEWG again.

The following papers had no consensus

-  [P2413R1](#): Remove unsafe conversions of unique_ptr
-  [P2921R0](#): Exploring std::expected based API alternatives for buffer_queue

Policies discussion

Policies were created to guide authors of standard library proposals, and by doing so, improve the process and save both the group and the authors’ time.

Information about policies can be found in: “[P2267R1](#): Library Evolution Policies (The rationale and process of setting a policy for the Standard Library)”.

-  [P2422R1](#): Remove nodiscard annotations from the standard library specification (plenary approved)
-  [P3116R0](#): Policy for explicit (should be seen again by LEWG)

Evening Sessions

We had two evening sessions during the week (initiated by our members).

Evening sessions are informative sessions, during which we do not take any binding votes.

They are meant for either reviewing topics relevant to the committee in more depth than possible during the work sessions (such is the case for the Senders/Receivers (P2300) session) , or for introducing topics which are not procedurally related but are relevant to WG21 (such is

the case for “The Beman Project”, which is an initiative by members of WG21 but not as part of their role in WG21).

- 🔍 Tuesday: “[P2300R10](#): std::execution” (AKA Senders/Receivers) - Deep Dive Introduction. Presented by:
 - Dietmar Kühl (an updated first part of his [CppCon 2022 talk](#))
 - Lewis Baker (slides for his paper: “[P3143R0](#): An in-depth walk through of the example in P3090R0”))
- 🔍 Thursday: [The Beman Project](#). Presented by: Jeff Garland.

LEWG will continue to run weekly telecons, we expect to continue the review on “Reflection” and P2300 follow up papers, and have the major features already approved by the time we get to the next meeting (Wrocław, Poland). Tentative policies to be discussed in Poland are: “Explicit Constructors” and “Allocators support”.

Thank you to all our authors and participants, for a great collaboration in a productive and useful review process, and see you (in-person or online) in Wrocław! 🙃

Herb Sutter : LWG is saturated and may not process all the papers that are forwarded to it. Was LEWG able to take some time to prioritize ?

Inbal Levi: We have a process for that. We’re planning to do the process in Poland. We have a meeting, people will raise their hands and create a queue of papers and priorities. We are aware of this issue. We would appreciate your effort in wording review.

Herb Sutter : it would be good to have help with wording review. We are three meetings from the freeze. If you want your library feature, argue for why your paper should be prioritized and help with wording review.

Alisdair Meredith : where do I find a link for the process ?

Inbal Levi : there is no link, this is an old process. Just to add, we didn’t have contracts and relocatability. We have dedicated time for them in telecons. The plan is to finish with relocatability and reflection before Poland.

Core (Maurer/Merrill/Caves)

CWG had regular telcos before St. Louis. We look at core issues and put them into tentatively ready status. We spent time here to double check those resolutions and we move them to ready status. We have a paper that is on today’s straw polls that relies on the resolution of CWG2867. We had another review and changes for CWG 2867 and we felt it was ready for today.

We reviewed a paper P562 which is about adding optional trailing commas. Richard Smith discovered that some implementation techniques would fail with that change so the paper is not in the polls today. We started a first pass at reflection. We made good progress, but we didn't get through it. We will need to put more work into it later.

I would like to thank the scribes.

We did not schedule telecons yet, so there will be an email soon.

CWG polls

1. Accept as Defect Reports and apply the proposed resolutions of all issues except 2819, 2858, and 2876 in [P3345R0](#) (Core Language Working Group "ready" Issues for the June, 2024 meeting) to the C++ Working Paper.

No discussion.
No objection to unanimous consent.
Motion passes.

2. Apply the proposed resolution of issues 2819, 2858, and 2876 in [P3345R0](#) (Core Language Working Group "ready" Issues for the June, 2024 meeting) to the C++ Working Paper.

No discussion.
No objection to unanimous consent.
Motion passes.

3. Apply the changes in [P2747R2](#) (constexpr placement new) to the C++ Working Paper.

No discussion.
No objection to unanimous consent.
Motion passes.

4. Apply the changes in [P1061R9](#) (Structured Bindings can introduce a Pack) to the C++ Working Paper. (poll did not pass)

Concerns are raised about the implementability of this paper.

EWG were not aware of those concerns when they discussed this paper. The paper can be sent back to EWG for consideration.

Herb Sutter : CWG made a poll to forward this, but we are aware that EWG is happy to see it again.

Objections in the room

Herb Sutter clarifies the voting rules.
In favour : 11 (3 in person + 8 online)
Opposed: 35 (26 in person + 9 online)
Abstain : 48 (33 in person + 15 online)

No consensus. The paper will be sent back to EWG.

People are reminded not to take photos of people's screens and of people voting.

5. Apply the changes in [P3144R2](#) (Deleting a Pointer to an Incomplete Type Should be Ill-formed) to the C++ Working Paper.

No discussion.
No objection to unanimous consent.
Motion passes.

6. Apply the changes in P2963R3 (Ordering of constraints involving fold expressions) to the C++ Working Paper.

No discussion.
No objection to unanimous consent.
Motion passes.

7. Apply the changes in P0963R3 (Structured binding declaration as a condition) to the C++ Working Paper.

No discussion
Objection in the room

In favour : 36 (21 in person + 15 online)
Opposed: 4 (3 in person + 1 online)
Abstain : 51 (34 in person + 17 online)

Herb Sutter : Are there any national concerns about this ?
No national concerns.

Motion passes

Library (Wakely/Garland/Kuhl)

LWG met all week, with pretty good attendance most days, thanks to everybody who took part. Especially to Jeff and Dietmar who ran the room, as I was remote half the time and absent the rest of the time.

Like every telecon for the past few months, most of the meeting was spent reviewing P2300, std::execution. The reviewers and Eric Niebler showed admirable stamina and patience. Completing that review was a huge achievement, well done to everybody. We also spent some time reviewing Matthias Kretz's simd proposal, with good progress on that. Jeff tells me they got about 60% through it, and should finish in Poland. Several "normal size" papers also got reviewed, and most of those were finished and are on today's polls. A couple of other papers need updates from the authors and we'll see them again soon. We dealt with a dozen or so issues, but the issues list keeps growing, so I think we'll spend some telecon time on issues when we resume telecons in August.

Monday had been announced as being a beginner friendly day for people to come to LWG and learn how we do things. We did get a couple of people with us for the first time, so thanks to them, I hope it was interesting or at least useful. If anybody is interested in seeing how LWG does things, please come to our telecons! I like to think that every LWG session is beginner friendly, except maybe Friday afternoon at the end of an in-person meeting when everybody is fairly tired ... or maybe that's the best time because we take everything less seriously by the end of the week.

LWG polls

1. Apply the changes for all Ready and Tentatively Ready issues in [P3341R0](#) (C++ Standard Library Ready Issues to be moved in St. Louis, Jun. 2024) to the C++ working paper.

No discussion.
No objection to unanimous consent.
Motion passes.

2. Apply the changes in [P2997R1](#) (Removing the common reference requirement from the indirectly invocable concepts) to the C++ working paper.

No discussion.
No objection to unanimous consent.
Motion passes.

3. Apply the changes in [P2389R2](#) (`dextents` Index Type Parameter) to the C++ working paper.

No discussion.
No objection to unanimous consent.
Motion passes.

4. Apply the changes in [P3168R2](#) (Give `std::optional` Range Support) to the C++ working paper.

Concern is raised that `optional` shouldn't be a range because it's confusing.
This issue has been discussed in the subgroup.

Objections in the room.
In favour : 56 (37 in person + 19 online)
Against :4 (3 in person + 1 online)
Abstain : 39 (27 in person + 12 online)
Motion passes

5. Apply the changes in [P3217R0](#) (Adjoints to "Enabling list-initialization for algorithms": `find_last`) to the C++ working paper.

No discussion.
No objection to unanimous consent.
Motion passes.

6. Apply the changes in [P2985R0](#) (A type trait for detecting virtual base classes) to the C++ working paper.

No discussion.
No objection to unanimous consent.
Motion passes.

7. Apply the changes in [P0843R14](#) (`inplace_vector`) to the C++ working paper.

Herb Sutter : This paper was delayed in Tokyo. Have all the concerns now been addressed ?

Jonathan Wakely : Yes. The paper was changed to limit the types that are constexpr which makes it possible to implement.

Inbal Levi : And there is also implementation experience now.

Objections in the room.

In favour : 70 (54 in person + 16 online)

Against : 1 (1 in person + 0 online)

Abstain : 29 (13 in person + 16 online)

Motion passes.

8. Accept as a Defect Report and apply the changes in [P3235R3](#) (`std::print` more types faster with less memory) to the C++ working paper.

No discussion.

No objection to unanimous consent.

Motion passes.

9. Apply the changes in [P2968R2](#) (Make `std::ignore` a first-class object) to the C++ working paper.

No Discussion.

Objections in the room.

In favour : 50 (30 in person + 20 online)

Against : 2 (2 in person + 0 online)

Abstain : 45 (33 in person + 12 online)

Motion passes.

10. Apply the changes in [P2075R6](#) (Philox as an extension of the C++ RNG engines) to the C++ working paper.

No discussion.

No objection to unanimous consent.

Motion passes.

11. Apply the changes in [P2422R1](#) (Remove `nodiscard` annotations from the standard library specification) to the C++ working paper.

No discussion.

No objection to unanimous consent.

Motion passes.

12. Apply the changes in [P2300R10](#) (`std::execution`) to the C++ working paper.

One national body reported that individual members had concerns. They would like to postpone this until Poland in order to increase concerns. The concerns are mostly about teachability and user story for beginners. Two people from that NB have promised time to work with authors of P2300 to produce such a paper.

“We have been trying to deliver things to make it easier for the users at least for half a year and will be happy to have more people to pitch in. I don't think we should be delaying the vote, we can write books for years after the feature. I will be happy to collaborate with people on this”

Herb Sutter : I polled other NBs yesterday. I heard from two with individual members having personal objections. There are no national objections.

“How is it possible that the committee only hears about this now ?”

“Sender receiver complexity is extremely high. Steps have been taken to improve the complexity, but this is not enough. An average programmers can not and will not use this. What we are developing is a full language in the library. This is a complex beast. LEWG review was not looking at realistic examples or any examples. There were many things that the people in the room didn't understand what was being discussed. Just now we're starting to get some real usage out of this by template metaprogramming, and the feedback coming from that group was unconvincing. I feel we didn't do proper design review. We got feedback about compiler times at CppNow. For a typical example the compile time is 30 seconds. I don't think this is something we can standardize. I think this is a detriment to the standard.”

“We started working on this paper 2 years ago. There was a lot of time for people to look at it and organize. “

“If you look at CppReference, examples have been there for a while now. This specific implementation and other implementations have been heavily used through the industry and many companies. My impression is that some statements made are not accurate.”

“ What we have is a proposal that has implementation experience and years of deployment experience. We have an extension on top of this proposal and they have implementation experience. This is plenty ready for adoption. Proposal authors have done everything we expect of proposal authors and more.”

“I think that people have expressed valid concerns. I feel delaying the paper by one meeting would allow better understanding. I feel a lot of people do not understand the proposal. I would like to know if the performance is comparable to competing models. I'm also concerned about teachability”

“We have been using this in production for 2 years and we have a friendly face. We have missing pieces in flight now. As soon as we have those we will be able to show examples. “

“Intel has a simplified implementation, In principle there is no broad adoption within Intel.”

“Meta has something similar. If we were standardizing that it would be a different story. One person implementing it isn't broad experience. This isn't broadly used. “

“Beman project should implement this and see.”

“This proposal has been in flight for years. I do not believe that people who have not spent time learning it until now will spend time learning it until next time. “

“There is an open source implementation out there, there is no need to do it in the Beman project.”

“We need this for coroutines, it is the final piece.”

Herb Sutter : We have a standing document about procedures. We want to hear your concerns, but we really appreciate not hearing them for the first time in the plenary. Please raise your concerns early so we can front load them.

Objections in the room

In favour : 57 (35 in person + 22 online)

Opposed : 20 (13 in person + 7 online)

Abstain : 27 (21 in person + 6 online)

Herb Sutter : Are there national bodies that are opposed to this ?

Two other national bodies reported that individual members had concerns, but did not determine national opposition at that time.

Herb Sutter : Numerically this is a weak consensus, but it is a consensus. I'm not hearing any NB concerns at this time, and it's been many months and there would have been time for NBs to raise concerns. Please talk about this and find answers to the questions.

Direction Group (Hinnant)

We have been meeting every other week. We have been looking at safety, pattern matching, contracts, and other subjects we believe could influence long term directions of C++.

Michael Wong : We have been discussing safety and security issues that have been coming up. We also gave specific directions to contracts. We do not think it should be a TS and that it should freeze. We discussed what that freezing mechanism would be as we don't have one yet.

8. Closing activities

8.1 Issues delayed until today

No discussion.

8.2 Mailings

Note: These are the closest regular mailings and not special pre/post meeting mailings.

- 2024-07-15: Post-St. Louis
- 2024-10-15: Pre-Wroclaw

8.3 Plans for the future

No discussion.

8.4 Next and following meetings

- 2024-11-18/23: Wroclaw, Poland ([N4974](#))
- 2025-02-10/15: Hagenberg, Austria ([N4979](#))

Bill Seymour: I would like to add my personal thanks to Jens Maurer for all the help he has been with logistics . I would like to thank all the people who showed up to unpack and pack up the equipment with a special shout out to Roger and Detlef. Most of all, thank you WG21 for giving me much more knowledge than I've ever given back.

9. Adjournment

Meeting adjourned at 10:10 AM GMT-5.

10. Attendance

Attendee	NB
Adams, Michael	SCC
Adelstein Lelbach, Bryce	ANSI
Alday, Juan	ANSI
Alexandrescu, Andrei	ANSI
Amini, Parsa	ANSI
Anisimov, Sergei	
Arkhipova, Olga	ANSI
Arutyunyan, Ruslan	ANSI
Ažman, Gašper	BSI
Baker, Billy	ANSI
Baker, Lewis	ANSI
Balog, Pal	ANSI
Banglawala, Neelofer	
Bastien, Jean-Francois	SCC

Attendee	NB
Bentley, Matthew	ANSI
Bernat, Yehezkel	SII
Berne, Joshua	ANSI
Bi, Brian	ANSI
Bindels, P.G.H.	NEN
Birbacher, Frank	ANSI
Blackwell, Bianca	SCC
Boeckel, Ben	ANSI
Boehm, Hans	ANSI
Bonaventura, Xavier	DIN
Bott, Harold	ANSI
Brito Gadeschi, Gonzalo	ANSI
Brown, Bret	ANSI
Brown, Walter E.	SII
Burylov, Ilya	ANSI
Butler, Matthew	ANSI
Büttner, Sebastian	ANSI
Chen, Yuxuan	ANSI
Childers, Wyatt	ANSI
Craig, Benjamin	ANSI
Craig, Philip	BSI
Cranmer, Joshua	ANSI
D'Angelo, Giuseppe	ANSI
Dathskovsky, Alex	SII
Dave, Jagrut	ANSI
Davidson, Guy	BSI
de Wever, Mark	ANSI
Delfino, Gianluca	UNI
Dionne, Louis	ANSI
Dos Reis, Gabriel	AFNOR
Douglas, Niall	NSAI
Douglas, Robert	ANSI
Doumler, Timur	BSI
Downey, Steve	ANSI

Attendee	NB
Dusikova, Hana	UNMZ
Engert, Daniela	DIN
Fertig, Andreas	DIN
Floyd, Paul	
Foco, Marco	UNI
Fracassi, Fabio	DIN
García Sánchez, José Daniel	UNE
Garland, Jeff	ANSI
Garland, Michael	ANSI
Genovese, Walter	ANSI
Gill, Mungo	NSAI
Giroux, Olivier	ANSI
Goldblatt, David	ANSI
Goodspeed, Nathaniel	ANSI
Green, Bob	Guest
Gruber, Bernhard	ANSI
Gustafsson, Bengt	SIS
Hagins, Jody	ANSI
Halpern, Pablo	ANSI
Hava, Michael Florian	ASI
Herring, Davis	ANSI
Hoemmen, Mark	ANSI
Hollman, Daisy	ANSI
Honermann, Tom	ANSI
Hughes, Lori	ANSI
Hunt, Oliver	ANSI
Izvekov, Matheus	ANSI
Jabot, Corentin	AFNOR
Jha, Dheeraj	BIS
Josuttis, Nicolai	DIN
Katz, Dan	ANSI
Khlebnikov, Rostislav	ANSI
Koeppel, Thomas	ANSI
Kretz, Matthias	DIN

Attendee	NB
Kuhl, Dietmar	ANSI
Kutlov, Tymofii	
Larson, Brad	
Lauko, Henrich	UNMZ
Laverdière-Papineau, Marc-André	ANSI
Lebrun-Grandie, Damien	ANSI
Lee, Hyungjin	ANSI
Levi, Inbal	SII
Li, Yihe	ANSI
Liber, Nevin	ANSI
Lopes, Bruno	ANSI
Maness, Wesley	ANSI
Maurer, Jens	ANSI
McDougall, Ryan	ANSI
McMonagle, John	BSI
Meerwald, Christof	ASI
Meredith, Alisdair	ANSI
Merrill, Jason	ANSI
Michael, Maged	ANSI
Morales, Nicolas	ANSI
Moschovakos, Paris	SNV
Müller, Jonathan	DIN
Na, Yeoul	ANSI
Neatu, Darius	ANSI
Niebler, Eric	ANSI
Nishanov, Gor	ANSI
Nolan, Edward	ANSI
O'Dwyer, Arthur	ANSI
Olsen, David	ANSI
Orr, Roger	BSI
Owen, Nathan	ANSI
Park, Michael	SCC
Peacock, Antony	BSI
Persson, Jonas	SIS

Attendee	NB
Petersen, Ian	ANSI
Preney, Paul	SCC
Pusz, Mateusz	PKN
Ranns, Nina Dinka	BSI
Ratzloff, Phil	ANSI
Regev, Ran	SII
Revzin, Barry	ANSI
Rigault, Jean-Paul	AFNOR
Rivera Morell, René Ferdinand	ANSI
Rodrigues, Guilherme	ASI
Ronkainen, Jari	SFS
Rosten, Oliver	BSI
Roy, Patrice	SCC
Ryan, Christopher	ANSI
Sandoe, Iain	BSI
Sankel, David	ANSI
Satle, Ankur	BIS
Scogland, Thomas	ANSI
Serebrennikov, Vladislav	ANSI
Seymour, William	ANSI
Sharma, Saksham	ANSI
Sherman, Benjamin	ANSI
Snyder, Jeff	BSI
Song, Tim	ANSI
Spencer, Michael	ANSI
Spicer, John	ANSI
St. Amour, Bryan	SCC
Stroustrup, Bjarne	ANSI
Sutter, Herb	ISO/IEC JTC 1/SC 22
Szolnoki, Lénárd	
Talbot, Alan	ANSI
Taylor, Matthew	BSI
Tenty, David	SCC
TEODORESCU, Lucian Radu	ASRO

Attendee	NB
Teoh, Joon Nam	ANSI
Tong, Hubert	SCC
Touton, James	ANSI
Towner, Daniel	ANSI
Tsaousis-Seiras, Isidoros	ANSI
Vandevoorde, Daveed	ANSI
Varlamov, Konstantin	ANSI
Vasama, Lauri	SFS
Vollmann, Detlef	SNV
Vormwald, Steven	ANSI
Voss, Michael	ANSI
Voutilainen, Ville	SFS
Wakely, Jonathan	BSI
Walker, Kelly	ANSI
Waterloo, Jarrad	ANSI
Weis, Andreas	DIN
Williams, Anthony	BSI
Williamson, Gerald	ANSI
Wong, Jessica	ANSI
Wong, Michael	SCC
Xie, Hui	BSI
xu, chuanqi	SAC
Yaghmour, Shafik	ANSI
Yao, Chuanqi	
Yuan, Zhihao	ANSI
Zimmermann, Philipp	
Zissu, Andrei	SII
Zverovich, Victor	ANSI