

WG21 March 2024 Hybrid meeting

Minutes of Meeting

ISO/IEC JTC1 SC22 WG21 N4980— 2024-04-05

Nina Dinka Ranns, dinka.ranns_at_gmail.com

Chair: John Spicer

18-23 March 2024, Tokyo, Japan

1. Opening activities

John Spicer opens the meeting at 09:06 AM GMT+9

1.1 Opening comments, welcome from host

John Spicer welcomes the group.

Thank you to JF Bastien and Woven for hosting this meeting.

Welcome from the host.

JF Bastien provides local information. There is a meet-up on Thursday. If you registered and you can't make it, please e-mail us so we can make room for someone else. The topic is reflection.

There is a local high school visiting on Thursday.

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.

Procedures are important for understanding how we work during the week.

If you have questions, please approach me, the secretary, your NB chair, or any of the subgroup chairs.

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 : the link to the attendance list is on the wiki and Mattermost. Please don't change the formatting. Let me know if you need a new paper number.

Nina Ranns : for the purposes of this meeting, your NB membership is the one that is represented in the ISO directory.

John Spicer explains rules for the voting in person and over the Zoom.

1.3 Introductions

New members introduce themselves.

John Spicer welcomes new members.

Herb Sutter welcomes new members.

Herb Sutter : there are over 220 people at this meeting. This is the biggest meeting we had in a long time.

1.4 Agenda review and approval

See also [P1000R5](#)

John Spicer presents the agenda. Break is at 10am and 3pm.

John presents the agenda for the meeting (break at 10am and 3pm)

Herb Sutter presents. If you want to host the meeting where C++26 ships, let me know. We are already at our 3rd meeting for C++26. That's 2 meetings away from the last meeting for the new language proposals. You can find this information in the mailing.

Motion to approve the meeting agenda.

No discussion.

No objections.

Approved.

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

Document	Editor's report	Prospective WD
C++ 26 Working Draft	N4972	N4971

Motion to approve the documents above.
No objections.
Approved.

1.6 Approval of the minutes of the previous meetings

Meeting	Minutes
WG21 Kona	N4970
WG21 pre-Tokyo administrative telecon	N4978

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

Herb Sutter : ISO is starting to look into succession planning. We are doing some of it already by ensuring every subgroup has a co-chair to help share the workload or step in if a chair is otherwise engaged. ISO is thinking of introducing a co-convener. Traditionally, for us, the INCITS chair has had that role, but ISO may make it more formal.

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

Jens Maurer presents information on subgroup sessions.

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.

Please raise any concerns early.

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)

SG1 met most of the week. We could have used some of the time we had given back at the end, even.

We're starting to see new post-P2300 content - close to half of the minuted discussions. It's an exciting time!

Remote participation worked well this time again. Thanks to the remote people for their patience.

We had one minute-taking snafu -- our first paper has incomplete minutes. I apologize for this.

See you all in St Louis.

SG4: Networking (Snyder/Ažman)

SG4 met Friday morning and saw P3185R0. We took a poll and encouraged further exploration in the direction of marrying P2300 and the IETF-developed TAPS standard in preference to focusing on low-level socket primitives for networking.

Thomas Rodgers will undergo more exploration.

SG6: Numerics (Kretz/Lippincott/McFarlane)

SG6 met for a day and a half this week.

We forwarded two papers:

- P3008R1 Atomic floating-point min/max
- P2746R4 Deprecate and Replace Fenv Rounding Modes

We reviewed and provided feedback, expecting to see a next revision in St. Louis:

- P3045R0 Quantities and units library
- P2964R0 Allowing user-defined types in `std::simd`
- P2993R0 Constrained Numbers (a late paper)

We didn't have a presenter for P3133R0 "Fast first-factor finding function" and had to postpone the review.

SG7: Compile-time programming (Dusikova/Vandevoorde)

SG7 met on Friday morning, discussed 4 papers. We forwarded P3096R0 to EWG and LEWG for discussion (with wording and implementation experience). We watched a presentation about P3157R0 (there will be a paper on its own in post-meeting mailing) where we discussed generative use-cases of reflection. These examples and requirements presented will help us to design and focus on usability.

There was an all day long discussion in EWG (in the morning as a joint session with LEWG) about P3096, where authors gathered a lot of feedback from a wider audience. The paper will have finished wording at the next meeting and will go before that through telecons with LEWG to properly go through library design of it.

Papers seen:

- P3095R0: ABI comparison with reflection
- P3037R1: `constexpr std::shared_ptr`
- P3096R0: Function Parameter Reflection in Reflection for C++26
- P3157R0: Generative Extensions for Reflection

SG9: Ranges (Hollman/Carter)

SG9 met for two full days in Tokyo. We saw 11 papers and one LWG issue.

We forwarded 5 papers to LEWG:

- * P3060: Add `std::views::upto(n)` [from P2760 - A plan for C++26 ranges]
- * P3138: `views::cache_last` [from P2760 - A plan for C++26 ranges]
- * P3136: Retiring `niebloids`
- * P3052: `view_interface::at`
- * P1729: Text parsing [only the tiny part related to ranges]

We gave feedback and requested revisions of 4 papers:

- * P3137: `views::to_input` [from P2760 - A plan for C++26 ranges]
- * P3179: C++ parallel range algorithms
- * P2022: Rangified version of `lexicographical_compare_three_way`
- * P3117: Extending conditionally borrowed

We decided not to pursue the remaining two papers. More details, including polls and agreed-upon summaries of discussions, are posted to the corresponding GitHub issues, which are linked on the SG9 page in the wiki.

SG10: Feature test (Revzin/Wakely)

We did not meet this week.

Everyone is doing a good job with adding relevant feature test macros.

SG14: Games & low latency (Wong)

SG14 did not meet this week. We continue to meet online every month.

SG15: Tooling (Spencer/Boeckel)

The Tooling Study Group met for 1 and a half days this week and saw 7 papers.

We decided to pursue adding structured response files and a new build database in the same vein as compilation databases but with support for modules to the Ecosystem IS.

We also saw several papers on implementation details of modules and provided some guidance to implementers and plan to meet with the Itanium ABI group.

We took a look at the reflection proposal to discuss some tooling concerns related to how users can debug the reification features in the proposal.

Lastly we had a good discussion on libraries/package management/build systems and what we can do to improve the dependency management problem.

SG16: Unicode (Honermann/Downey)

SG16 (Unicode) did not meet in Tokyo. We have plenty to work on and will continue to meet at our usual twice monthly cadence.

SG17: EWG Incubator (Keane)

SG17 (Evolution Incubator) met all day Thursday and managed to look at 8 papers. Extensive, lively, and productive feedback was given to each author, so each paper is expecting a revision. 6 of the 8 papers were advanced. P2865 is in early development and while it got great feedback, we will not be looking to advance to EWG for a while yet. P3158 did not get consensus to continue and is likely to be abandoned.

The following papers were discussed:

P3087R0: Make direct-initialization for enumeration types at least as permissive as direct-list-initialization

P3140R0: `std::int_least_128_t`

P2034R3: Partially Mutable Lambda Captures

P2685R1: Language Support for Scoped Objects

P2822R0: Providing user control of associated entities of class types

P3158R0: Headless Template Template Parameters

P2830R2: Standardized Type Ordering

D2825R1: `declcall(unevaluated-postfix-expression)`

SG18: LEWG Incubator (Baker/Liber)

SG18 Library Evolution Working Group Incubator met Monday and Wednesday afternoons. 5 papers were discussed.

SG18 will likely have teleconferences once a month before St Louis. SG18 will also meet in St Louis.

Forwarded to LEWG

P3105 `constexpr std::uncaught_exceptions()`

Needs revision

P3086 Proxy: A Pointer-Semantics-Based Polymorphism Library

P3104 Bit permutations

P3094 `std::basic_fixed_string`

In progress

P3045 Quantities and units library

SG19: Machine Learning (Wong/Reverdy)

SG19 met this week. We reviewed the proposal for basic statistics and proposals for the graph library. We will continue to meet monthly.

SG20: Education (van Winkel)

SG20 did not meet in Tokyo.

SG21: Contracts (Spicer/Doumler)

Leading up to the Tokyo meeting, SG21 had ten telecons, during which we completed P2900R6 (the Contracts MVP paper) and polled to forward it to EWG and LEWG. The paper was seen this week by both groups (LEWG on Monday, EWG on Wednesday) and we received plenty of feedback from both groups as well as vendor feedback from clang.

SG21 was sitting on Thursday afternoon and Friday afternoon in Tokyo to process this feedback. On Thursday we discussed LEWG feedback. We adopted a bunch of naming changes and processed most of their feedback. We also saw P3191R0 "Feedback on the scalability of contract violations handlers in P2900" (Louis Dionne, Yeoul Na, Konstantin Varlamov). We incorporated the new contract evaluation semantic proposed in P3191R0, addressing the vendor feedback from clang

On Friday we had a joint session with SG23 (Safety & Security) to discuss the safety concerns about P2900R6 that have been brought up in EWG this week. As a result of this session, we decided that the directional paper P2680R1 "Contracts for C++: Prioritizing Safety" (which was previously considered, but not adopted, by SG21) should be seen by SG23. We request guidance from SG23 on whether we should reconsider the direction proposed by P2680R1 for the Contracts MVP; in case the answer will be yes, we expect a specification paper to be available for St. Louis.

SG21 will continue to have regular telecons post Tokyo. The priorities for those telecons will be first to finish processing EWG and LEWG feedback on P2900R6, and then to get consensus on a solution for supporting contract assertions on virtual functions.

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

SG22 did not meet in Tokyo.

SG23: Safety/Security (Orr/Craig)

SG23 met on Monday and looked at P3052R1 "view_interface::at()" We polled in favour of making progress on this paper, but do not need to see it again in SG23.

SG23 members were encouraged to attend the EWG day on contracts on Wednesday.

SG23 met on Friday afternoon with SG21 to discuss UB in contracts.

We agreed to look at P2680R1 Contracts for C++: Prioritizing Safety in SG23.

ABI Group (Vandevorde)




ABI group never meets and did not have any queries to address this week.

Admin (Liber)

We did not meet. Next mailing will be April 16th.

Evolution (Bastien/Stone/Keane/Dusikova)

Language Evolution met for the entire week, prioritizing big features that are attempting to make their way to C++26:






- An entire day was spent discussing Reflection. [P2996R2](#) — Reflection for C++26 —  feedback provided, very positive response, making good progress towards C++26
- An entire day was spent discussing Contracts. [P2900R6](#) — Contracts for C++ —  significant feedback was given to the group, though the polls show a significantly divided Committee which might have difficulty reaching consensus. [Timur's slides](#). [P3173R0](#) — P2900R6 may be minimal, but it is not viable — a set of sustained objection from Microsoft were provided and discussed.
- A half day was spent discussing Pattern Matching. [P2688R1](#) — Pattern Matching: match Expression —  an update was provided on the design, and feedback was given to the authors, an updated implementation will be needed soon if we want to have pattern matching in C++26.

These 3 features are the culmination of many years of work from dedicated committee members, and EWG spent significant effort trying to provide actionable feedback to help get the features ready, we hope for C++26.

Unlike in recent meetings, EWG was not able to see all papers scheduled. We will therefore be resuming bi-weekly telecons.

Bug fixes

We reviewed 7 issues.

- [CWG2843](#) — Undated reference to Unicode makes C++ a moving target —  agree to specify a specific version of the Unicode standard
- [CWG2836](#) — Conversion rank of long double and extended floating-point types —  keep the deviation between C and C++
- [CWG2819](#) — Cast from null pointer value in a constant expression —  support conversions from void* to a pointer-to-object type also when the original pointer is a null pointer value
- [CWG2565](#) — Invalid types in the parameter-declaration-clause of a requires-expression — we agree that this is an issue, but  paper needed
- [CWG2804](#) — Lookup for determining rewrite targets — we agree that this is an issue, but  paper needed

- [CWG2797](#) — Meaning of "corresponds" for rewritten operator candidates — ❌ no consensus for change
- [CWG2776](#) — Substitution failure and implementation limits — 📄 paper needed
- [CWG2784](#) — Unclear definition of member-designator for offsetof — 📄 paper needed
- [CWG2726](#) — Alternative tokens appearing as attribute-tokens — 📄 paper needed
- [CWG1699](#) — Does befriending a class befriend its friends? — 📄 paper needed
- [CWG2669](#) — Lifetime extension for aggregate initialization — 📄 paper needed
- [P3031R0](#) — Resolve CWG2561: conversion function for lambda with explicit object parameter — ✅ move forward with proposed option 1

Papers forwarded to CWG

We forwarded 7 papers to CWG.

- [P3034R0](#) — Module Declarations Shouldn't be Macros — ✅ Forward P3034R0 to CWG for inclusion in C++ as a DR against C++20.
- [P2786R3](#) — Trivial Relocatability For C++26 — ✅ progressing through CWG, disallow trivially_relocatable(💎) on unnamed struct/class/union. An issue remains to resolve regarding a new vexing parse in code such as `struct C trivially_relocatable (bool(my_constexpr_value)) { }`;
- [P2573R1](#) — `= delete("should have a reason");` — ✅ forward to CWG for inclusion in C++26
- [P2963R1](#) — Ordering of constraints involving fold expressions — ✅ forward to CWG for inclusion in C++26
- [P2686R2](#) — `constexpr` structured bindings and references to `constexpr` variables — ✅ progressing through CWG, but require implementation experience before moving to plenary
- [P3032R0](#) — Less transient `constexpr` allocation — ✅ Forward to CWG as DR for C++20
- [D0562R1](#) — Initialization List Symmetry — ✅ also add support for base class trailing commas, and send to CWG for inclusion in C++26.










Papers sent back to EWG from CWG

CWG sent us 7 papers back, as they do.

- [P2809R3](#) — trivial infinite loops — validate the removal of `yield_forever`
- [P2747R1](#) — `constexpr` placement `new` — array support? currently not supported by `construct_at`; see LWG issue
- [P0609R2](#) — attributes for structured bindings — also adjust range-for-declaration
- [D2795R5](#) — Erroneous behaviour for uninitialized reads — EWG agrees that EB may create values which later trigger UB, for example an EB read of a pointer has a defined value but dereferencing the pointer may then be UB.
- [P2893R2](#) — variadic friends — friend T1, T2; is now supported, and the template expansion of friend expands to this new syntax.
- [D1061R8](#) — Structured Bindings can introduce a Pack — disallow the feature in namespace scope, send back to CWG
- [P2841R1](#) — Concept Template Parameters — send to CWG for inclusion in C++26



Papers which received feedback

9 papers received feedback and will be coming back.

- [P3068R0](#) — Allowing exception throwing in constant-evaluation —  feedback provided, needs implementation experience
- [P2481R2](#) — Forwarding reference to specific type/template —  feedback provided
- [P2826R2](#) — Replacement functions —  feedback provided
- [P3074R2](#) — `std::uninitialized` —  feedback provided
- [P2758R2](#) — Emitting messages at compile time —  feedback provided
- [P2994R1](#) — On the Naming of Packs —  feedback provided
- [P3115R0](#) — Data Member, Variable and Alias Declarations Can Introduce A Pack —  encourage more work
- [P2414R1](#) — Pointer lifetime-end zap proposed solutions —  encourage more work
- [P2806R2](#) — do expressions —  encourage more work

Papers which did not proceed

2 papers did not have consensus to proceed.

- [P3110R0](#) — Array element initialization via pattern expansion —  not interested to proceed with the paper as-is
- [P2355R2](#) — Postfix fold expressions —  not interested to proceed with the paper as-is

Papers we didn't have time to see

5 papers didn't have sufficient time to be seen.

- [P2992R1](#) — Attribute `[[discard]]` and attributes on expressions
- [P2989R0](#) — A Simple Approach to Universal Template Parameters
- [P0963R1](#) — Structured binding declaration as a condition
- [P0876R14](#) — `fiber_context` - fibers without scheduler
- [P3006R0](#) — Launder less

Papers without presenters available

4 papers came in without presenters identified.

- [P1203R0](#) — modular `main()`
- [P2607R0](#) — Let alignas specify minimum alignment
- [P2191R0](#) — Modules: ADL & GMFs do not play together well (anymore)
- [P1046R2](#) — Automatically Generate More Operators

Library Evolution (Levi/Fracassi/Craig)

Library Evolution Work Group continues to hold weekly telecons in order to handle the workload in between in-person WG21 meetings.

The Tokyo meeting has brought together a lot of our experienced attendees. This was very beneficial for reviewing a lot of fixes and giving valuable feedback to large features. LEWG had its first policies discussion during the Tokyo meeting. 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:

[P2267](#): Library Evolution Policies (The rationale and process of setting a policy for the Standard Library)

On Tuesday morning, LEWG conducted a combined session with EWG on the topic of "Reflection". We saw a first introduction to the paper: "[P2996R2](#): Reflection for C++26", and brought up concerns and issues that may occur and need to be considered, both from the standard library's perspective (implicit guarantees which may be implied between versions) and from reflection-based libraries' perspective in general (e.g. "[P3096R0](#): Function Parameter Reflection in Reflection for C++26"). We will follow up on these topics (and any others that may occur) in both telecons and future meetings, with the aim of having P2996 ready for delivery in C++26.

Features Approved

Forwarded (directly)

- [P3107R5](#): Permit an efficient implementation of `std::print` (Bug fix, support by implementers)
- [P2845R6](#): Formatting of `std::filesystem::path` (Minor fix)
- [LWG3918](#): `std::uninitialized_move/_n` and guaranteed copy elision

Forwarded (pending an Electronic Poll)

- [P2855R1](#): Member customization points for Senders and Receivers (P2300 Improvement)
- [P2019R5](#): Thread attributes
- [P2927R2](#): Observing exceptions stored in `exception_ptr`
- [P2997R0](#): Removing the common reference requirement from the indirectly invocable concepts
- [P3168R0](#): Give `std::optional` Range Support
- [P2075R4](#): Philox as an extension of the C++ RNG engines
- [P2835R3](#): Expose `std::atomic_ref`'s object address
- [P3016R2](#): Resolve inconsistencies in `begin/end` for `valarray` and braced initializer lists

Features Reviewed

Progress on Large Features

- [P2996R3](#): Reflection for C++26 (Joint EWG/LEWG session)
- [P2900R6](#): Contracts for C++
- [P3149R2](#): `async_scope` -- Creating scopes for non-sequential concurrency (addition to [P2300](#))
- [P2643R2](#): Improving C++ concurrency features
- [P1708R8](#) Basic Statistics

Other Library Features Reviewed

- [P2988R3](#): `std::optional<T&>`
- [P3091R0](#): Better lookups for ``map`` and ``unordered_map``
- [P2946R1](#): A flexible solution to the problems of ``noexcept``
- [P2721R0](#): Deprecating function
- [P3166R0](#): Static Exception Specifications
- [P1255R12](#): A view of 0 or 1 elements: `views::maybe`

Library Evolution Policies Progress

Policies approved (pending an electronic poll)

Approved:

- [P3201R0](#): LEWG `[[nodiscard]]` policy

Discussion included:

- [P3162R0](#): LEWG `[[nodiscard]]` policy
- [P3122R0](#): `[[nodiscard]]` should be Recommended Practice
- [P2422R0](#): Remove `nodiscard` annotations from the standard library specification

Policies Requires Additional Discussion

- [P2946R1](#): A flexible solution to the problems of ``noexcept``
- [P2837R0](#): Planning to Revisit the Lakos Rule
- [P3085R0](#): ``noexcept`` policy for SD-9 (throws nothing)
- [P3155R0](#): `noexcept` policy for SD-9 (The Lakos Rule)

Papers not supported or needs-revision

- [P3022R0](#): A Boring Thread Attributes Interface
- [P3147R0](#): A Direction for Vector
- [P3160R0](#): An allocator-aware ``inplace_vector``

We will continue our weekly telecons until St. Louis.

On St. Louis, we might have an evening session on one of the following “[P3045R0](#): Quantities and units library”, “[P1708R8](#): Basic Statistics”, [P2786R4](#): Trivial Relocatability For C++26”, “[P0260R5](#): C++ Concurrent Queues”, or other major features.

Thank you to all our assistant chairs, paper writers, minute takers, and most of all, to our participants!

Core (Maurer)

We had a productive week, with 47 core issues and 9 papers ready for today.

I'd like to highlight "Erroneous behaviour for uninitialized reads", which removes undefined behavior for some cases of uninitialized objects.

Paper P2747R2 "constexpr placement new" did not make it onto today's straw polls and will be delayed to St. Louis.

There is no significant backlog for paper review in CWG.

We intend to hold teleconferences every two weeks between now and St. Louis.

I'd like to remind everybody that C++26 is near, and large papers such as pattern matching, contracts, and reflection seem to be on the horizon, competing for last-minute CWG review bandwidth. We should be prepared to make tough choices here, unless at least one of those papers arrives in CWG soon.

CWG polls

1. Accept as Defect Reports and apply the proposed resolutions of all issues in [P3196R0](#) (Core Language Working Group "ready" Issues for the March, 2024 meeting) to the C++ Working Paper.

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

2. Apply the changes in [P2748R5](#) (Disallow Binding a Returned Glvalue to a Temporary) to the C++ Working Paper.

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

3. Accept as a Defect Report and apply the changes in [P3106R1](#) (Clarifying rules for brace elision in aggregate initialization) to the C++ Working Paper, resolving core issue 2149.

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

4. Apply the changes in [P0609R3](#) (Attributes for Structured Bindings) to the C++ Working Paper.

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

5. Accept as a Defect Report and apply the changes in [P3034R1](#) (Module Declarations Shouldn't be Macros) to the C++ Working Paper.

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

6. Accept as a Defect Report and apply the changes in [P2809R3](#) (Trivial infinite loops are not Undefined Behavior) to the C++ Working Paper.

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

7. Apply the changes in [P2795R5](#) (Erroneous behaviour for uninitialized reads) to the C++ Working Paper.

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

8. Accept as a Defect Report and apply the changes in [P3032R1](#) (Less transient constexpr allocation) to the C++ Working Paper.

A concern is raised that the wording doesn't do what this proposal wants to achieve. Herb Sutter asks the CWG chair if this was considered. The paper was considered as a whole and the group thought it was done. There is agreement that the wording doesn't achieve what the paper claims and that the change needed is pretty fundamental. Point raised that there is no pressure to progress this paper at this meeting.

Any objection to removing the poll ?
No objection.

Poll number 8 is removed.

9. Apply the changes in [P2573R2](#) (= delete("should have a reason");) to the C++ Working Paper.

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

10. Apply the changes in [P2893R3](#) (Variadic friends) to the C++ Working Paper.

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

Library (Wakely/Garland/Kuhl)

After weekly telecons leading up to the meeting, LWG spent one quarter of the meeting on P2300. We achieved the milestone of completing a first pass through the entire paper! We will continue to work in telecom on a second pass and are now mildly optimistic of completing P2300 in St. Louis.

In addition to P2300 we worked almost exclusively on other papers making significant progress on P0867 fiber_context and P3019 indirect and polymorphic. We also assisted Core in wording the library aspects of the Erroneous behavior paper.

We are moving 18 papers (although inplace_vector has late news which may lead to withdrawal). I'd highlight a small paper P2591 which will finally allow seamless concatenation of strings and string_views shaving off a long standing sharp edge in the standard library. Also in the related family views::concat P2542 will extend range facilities.

Remote participation was a challenge in particular in the european timezones.

C++26's deadlines are quickly approaching and LWG already has significant work in std::simd and P2300 follow-ons. It seems ever more likely that any last minute large features will face potential slip to c++29 due to lack of time.

LWG polls

1. Apply the changes for all Ready and Tentatively Ready issues in [P3180R0](#) (C++ Standard Library Ready Issues to be moved in Tokyo, Mar. 2024) to the C++ working paper.

No discussion.

No objection to unanimous consent.

Motion passes.

2. Apply the changes in [P2875R4](#) (Undeprecate polymorphic_allocator::destroy for C++26) to the C++ working paper.

No discussion.

No objection to unanimous consent.

Motion passes.

3. Apply the changes in [P2867R2](#) (Remove Deprecated strstreams From C++26) to the C++ working paper.

No discussion.

Objection to unanimous consent.

Herb Sutter reminds the room of the voting rules.

In favor : 85 (52 in person + 33 online)

Opposed : 1 (0 in person + 1 online)

Abstain : 11 (7 in person + 4 online)

Motion passes.

4. Apply the changes in [P2869R4](#) (Remove Deprecated shared_ptr Atomic Access APIs from C++26) to the C++ working paper.

No discussion.

Objection to unanimous consent.

In favor : 81 (52 in person + 29 online)

Opposed : 1 (0 in person + 1 online)

Abstain : 12 (7 in person + 5 online)

Motion passes.

5. Apply the changes in [P2872R3](#) (Remove wstring_convert From C++26) to the C++ working paper.

No discussion.

Objection to unanimous consent.

In favor : 76 (48 in person + 28 online)

Opposed : 1 (0 in person + 1 online)

Abstain : 11 (7 in person + 4 online)

Motion passes.

6. Accept as a Defect Report and apply the changes in [P3107R5](#) (Permit an efficient implementation of std::print) to the C++ working paper.

No discussion.

No objection to unanimous consent.

Motion passes.

7. Apply the changes in [P3142R0](#) (Printing Blank Lines with println) to the C++ working paper.

No discussion.

No objection to unanimous consent.

Motion passes.

8. Apply the changes in [P2845R8](#) (Formatting of std::filesystem::path) to the C++ working paper.

No discussion.

No objection to unanimous consent.

Motion passes.

9. Apply the changes in [P0493R5](#) (Atomic minimum/maximum) to the C++ working paper.

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

10. Apply the changes in [P2542R8](#) (views::concat) to the C++ working paper.

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

11. Apply the changes in [P2591R5](#) (Concatenation of strings and string views) to the C++ working paper.

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

12. Apply the changes in [P2248R8](#) (Enabling list-initialization for algorithms) to the C++ working paper.

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

13. Apply the changes in [P2810R4](#) (is_debugger_present is_replaceable) to the C++ working paper.

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

14. Apply the changes in [P1068R11](#) (Vector API for random number generation) to the C++ working paper.

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

15. Apply the changes in [P0843R11](#) (inplace_vector) to the C++ working paper.

Two concerns were raised: Should the feature support allocators? Is the feature implementable?

Herb Sutter asked the subgroup chairs to summarize whether these issues were considered. The allocator issue is part of a different paper that went to LEWG this week. The unimplementability was not discussed in LWG or LEWG this week.

A statement is made that the problem with implementability was mentioned before, and that it used to be mentioned in the paper. People were aware of it, some of LEWG said it's fine and can be implemented. It became obvious last night that it isn't implementable. Constexpr is purely for decoration and doesn't do anything.

The chairs and authors are happy to pull the poll.
One author has an objection.

A statement is made that the question is whether anyone implemented this, and that the answer to that question is no.
We don't have technical discussions in the plenary. If there are no objections in the room, we pull the poll.

The objection to removing the poll has been retracted.
No further objections, the poll has been retracted.

SG14 has scheduled this for discussion in the next telco.

16. Apply the changes in [P2944R3](#) (Comparisons for reference_wrapper) to the C++ working paper.

A statement is made that a minor issue was discovered when implementing this paper. The minor issue can be resolved as an issue. There is no need to postpone this poll.
No objection to unanimous consent.
Motion passes.

17. Apply the changes in [P2642R6](#) (Padded mdspan layouts) to the C++ working paper.

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

18. Apply the changes in [P3029R1](#) (Better mdspan's CTAD) to the C++ working paper.

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

Direction Group (Hinnant)

We don't have sufficient new material to update P2000, but we have been looking at safety, pattern matching, and senders/receivers that might lead to updates.

We plan to review the Government security document just published:

<https://www.whitehouse.gov/oncd/briefing-room/2024/02/26/press-release-technical-report/>
<https://www.whitehouse.gov/wp-content/uploads/2024/02/Final-ONCD-Technical-Report.pdf>

This press release has already led to some public discussion such as:

<https://www.infoworld.com/article/3713203/white-house-urges-developers-to-dump-c-and-c.html>

We do not believe this report reflects the dramatic evolution C++ has undergone since its initial specification in 1998. Each of C++11, C++14, C++17, C++20 and C++23 has added significant enhancements not only to memory safety, but to **resource** safety. That is, any resource, be it memory, file handles, the ownership of mutex lock states, thread handles, etc. must be handled safely. Modern C++ is uniquely equipped to meet this challenge, and with a clean migration path from older versions of C++, and even from C.

8. Closing activities

8.1 Issues delayed until today

Inbal Levi presents slides with photos from the Tokyo meeting.

Herb Sutter : thank you to everyone who is carrying equipment. Also, thank you to Jens for organizing everything. Thank you to the host.

JF Bastien : thank you for picking up the trash and being good guests.

8.2 Mailings

2024-04-16: Post-Tokyo

2024-05-15: Pre-St. Louis

8.3 Plans for the future

8.4 Next and following meetings

2024-06-24/29: St. Louis, MO, USA (N4966)

2024-11-18/23: Wroclaw, Poland (N4974)

2025-02-10/1515: Hagenberg, Austria; University of Applied Sciences, Upper Austria
2025 Jun: Sofia, Bulgaria; Chaos Group
2025-11-03 to 08: Kona, HI, USA; Standard C++ Foundation

Herb Sutter : the hotel information for the Wroclaw meeting is available. Thank you to JF and Toyota for hosting us. Thank you to Bill Seymour for hosting us in St Louis. We have dates for other meetings too, papers will be available soon. We are very thankful to our hosts who make it possible to have the meetings.

9. Adjournment

Meeting adjourned at 10:11 GMT+9

10. Attendance

Attendee	National body
Adams, Michael	SCC
Adelstein Lelbach, Bryce	ANSI
Alday, Juan	ANSI
Alexandrescu, Andrei	ANSI
Amini, Parsa	ANSI
Arutyunyan, Ruslan	ANSI
Ažman, Gašper	BSI
Baker, Billy	ANSI
Baker, Lewis	ANSI
Balog, Pal	ANSI
Banglawala, Neelofer	GUEST of BSI
Bastien, Jean-Francois	SCC
Berne, Joshua	ANSI
Bi, Brian	ANSI
Birbacher, Frank	ANSI
Blackwell, Bianca	GUEST of SCC
Boeckel, Ben	ANSI
Boehm, Hans	ANSI
Bonaventura, Xavier	DIN

Brauge, Corentin	GUEST
Brito Gadeschi, Gonzalo	ANSI
Brown, Walter E	GUEST of ANSI
Butler, Matthew	ANSI
Cardoso de Souza Rodrigues, Guilherme	ASI
Cardoso Lopes, Bruno	GUEST of ANSI
Caves, Jonathan	ANSI
Childers, Wyatt	ANSI
Chochlik, Matus	UNMS SR
Chorazewicz, Igor	ANSI
Coe, Jonathan Brian	BSI
Corden, Richard	ANSI
Coste, Maxime	GUEST
Craig, Benjamin	ANSI
Craig, Philip	BSI
Cranmer, Joshua	ANSI
D'Angelo, Giuseppe	ANSI
Davidson, Guy	BSI
de Wever, Mark	ANSI
Dionne, Louis	SCC
Dominiak, Michal	PKN
Dos Reis, Gabriel	AFNOR
Dosselmann, Richard	GUEST of SCC
Douglas, Niall	NSAI
Douglas, Robert	ANSI
Doumler, Timur	BSI
Downey, Steve	ANSI
Dusikova, Hana	UNMZ
Engert, Daniela	ANSI
Erdogmus, Kaan	ANSI
Estell, Khalil	ANSI
Fertig, Andreas	DIN
Fevold, Jake	ANSI
Floyd, Paul	GUEST
Fokin, Aleksandr	GOST R
Fracassi, Fabio	DIN

Ganetsky, Braden	SCC
García Sánchez, José Daniel	UNE
Genovese, Walter	ANSI
Gill, Mungo	NSAI
Gillard, Mark	SFS
Giroux, Olivier	ANSI
Goldblatt, David	ANSI
Goodspeed, Nathaniel	ANSI
Gordon, Fraser	SCC
Gruber, Bernhard Manfred	SNV
Gunnin, Cameron	ANSI
Hagins, Jody	ANSI
Halpern, Pablo	ANSI
Hauswedell, Hannes	IST
Hava, Michael Florian	ASI
Herring, Davis	ANSI
Hinnant, Howard	ANSI
Hoemmen, Mark	ANSI
Hollman, Daisy	ANSI
Honermann, Tom	ANSI
Hussong, Charles	GUEST
Jabot, Corentin	AFNOR
Josuttis, Nicolai	DIN
KAMINISKI, Tomasz	AFNOR
Keane, Erich	ANSI
Keir, Paul	BSI
Khlebnikov, Rostislav	ANSI
Koeppe, Thomas	ANSI
Kostur, Andre	SCC
Kosunen, Elias	SFS
Kozicki, Bronek	BSI
Kretz, Matthias	DIN
Krzemienski, Andrzej	PKN
Kuhl, Dietmar	ANSI
Kulczycki, Peter	ASI
Laine, Timothy	ANSI

Lakos, John	ANSI
Laverdière-Papineau, Marc-André	ANSI
Lazaric, Ivan	GUEST of NEN
Leahy, Robert	SCC
Lebrun-Grandie, Damien	ANSI
Lee, Hyungjin	ANSI
Levi, Inbal	SII
Li, Jianzhong	SAC
Li, Yihe	ANSI
Liber, Nevin	ANSI
Lippincott, Lisa	GUEST
Loiseau, Julien	ANSI
Machutova, Jana	UNMZ
Maness, Wesley	ANSI
Marr, Greg	ANSI
Maurer, Jens	ANSI
McDougall, Ryan	GUEST of ANSI
McKenney, Paul	ANSI
Meerwald, Christof	ASI
Meredith, Alisdair	ANSI
Michael, Maged	ANSI
Morales, Nicolas	ANSI
Mueller, Gideon	ANSI
Müller, Jonathan	DIN
Murzin, Sergei	ANSI
Na, Yeoul	ANSI
Nadolski, Maikel	GUEST of DIN
Neatu, Darius	ANSI
Ng, Song Guan	GUEST
Niebler, Eric	ANSI
Nishanov, Gor	ANSI
Nolan, Edward	ANSI
O'Dwyer, Arthur	ANSI
Olsen, David	ANSI
Orr, Roger	BSI
Owen, Nathan	ANSI

Panzica, Adam	ANSI
Park, Michael	SCC
Peacock, Antony	BSI
Perä, Tuomo	GUEST of FSF
Persson, Jonas	SIS
Petersen, Ian	ANSI
Polukhin, Anton	GOST R
Preney, Paul	SCC
Pusz, Mateusz	PKN
Ranns, Nina Dinka	BSI
Ratzloff, Phil	ANSI
Reverdy, Vincent	AFNOR
Revzin, Barry	ANSI
Rigault, Jean-Paul	AFNOR
Rivera Morell, René Ferdinand	ANSI
Robin, Rowe	ANSI
Rodgers, Tom	ANSI
Ronkainen, Jari	SFS
Roy, Patrice	SCC
Ruoso, Daniel	ANSI
Ryan, Christopher	ANSI
Sankel, David	ANSI
Satle, Ankur	BIS
Scogland, Thomas	ANSI
Serebrennikov, Vladislav	GUEST
Sharma, Saksham	ANSI
Shoop, Kirk	ANSI
Snyder, Jeff	BSI
Song, Tim	ANSI
Spencer, Michael	ANSI
Spicer, John	ANSI
St. Amour, Bryan	SCC
Stone, David	ANSI
Stroustrup, Bjarne	ANSI
Sutter, Herb	ISO/IEC JTC 1/SC 22
Talbot, Alan	ANSI

Teeple, Doug	ANSI
Tenty, David	SCC
Terashima, Yui	JISC
Tomazos, Andrew	GUEST of ANSI
Tong, Hubert	SCC
Touton, James	ANSI
Towner, Daniel	ANSI
Trott, Christian	ANSI
Valenty, Luke	ANSI
Vandevoorde, Daveed	ANSI
Varlamov, Konstantin	ANSI
Vasama, Lauri	SFS
Voicu, Alexandru	ANSI
Vollmann, Detlef	SNV
Vormwald, Steven	ANSI
Voss, Michael	ANSI
Voutilainen, Ville	SFS
Wakely, Jonathan	BSI
Walker, Kelly	ANSI
Wang, Mingxin	SAC
Weile, Wei	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
Yuan, Zhihao	ANSI
Zissu, Andrei	SII
Zverovich, Victor	ANSI