WG21 February 2025 Hybrid meeting
Minutes of Meeting

ISO/IEC JTC1 SC22 WG21 N5007— 2024-02-26
Nina Dinka Ranns, dinka.ranns_at_gmail.com

Chair: John Spicer

10-15 February 2025, Hagenberg, Austria

1. Opening activities

John Spicer opens the meeting at 09:01 AM UTC+1.

The wifi information is on the screen.

1.1 Opening comments, welcome from host

John Spicer welcomes the group.

Welcome from the host.
Thank you to the host.
Thank you to the sponsors.

1.2 Meeting guidelines

John Spicer presents.

We have a wiki that contains meeting specific information. Please do not edit the wiki unless you have been explicitly instructed to do so.

We also have a github tracker. Please do not update github issues unless you have been explicitly instructed to do so by the relevant chair.

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 : If you are a convener invited guest and want to continue participation, please let me know. The list of convener invited guests is on the wiki. If you think you should be there, but aren't, let me know. If you haven't received an email from me with the information about this meeting, please e-mail me or find me after the plenary.

Nina Ranns : 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.

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. You can find it on wiki, mattermost, and posted to the reflector. If you have registered, it's pre-populated with your information. Please mark your attendance. If you have any issues, let me know.

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 welcomes the group.

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

Additional, lower-priority goals include:
Subgroup work that may target versions after C++26

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	N5002	N5001

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

1.6 Approval of the minutes of the previous meetings

Meeting	Minutes
WG21 Wroclaw	N5000
WG21 pre-Hagenberg administrative telecon	N5005

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.

Jens Maurer presents local amenities.

Room assignments are on the wiki page. Any questions or issues with remote attending setup, please find Jens Maurer.

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. If the poll refers to the new paper that hasn't been in the mailing, it should be attached to the straw polls page. 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. Core group and library group have a staging area, so keep an eye out on those..

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.

If you need a paper number, please contact Nevin Liber.

Herb Sutter addresses the national bodies. If you find things that are of national concern to you, and you have a NB caucus, please let Herb know as soon as possible.
First question will be have you raised this concern in the subgroup.

7. Review of the meeting

Herb Sutter welcomes the group and thanks everyone for hard work.

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/Boehm/Arutyunyan)

SG1 met for 4 days.
No known Senders/Receivers (potential) blockers.
Started to look at C++29 Senders/Receivers papers. We expect to see more at next meetings.

Forwarded Concurrent Queues (again) and Pointer Lifetime Zap.
Provided feedback on Guarded Objects.
Resolved CWG and LWG issues.

SG4: Networking (Snyder/Ažman)

SG4 met to discuss “P3482R0: Design for C++ networking based on IETF TAPS” - a paper proposing a design for a networking library based on senders/receivers. The paper shows a promising direction for secure-by-default networking in C++, and SG4 encouraged the author to continue work in P3842’s direction.

Separately, in a combined session with LEWG, we discussed the status of the existing networking TS. We voted to withdraw it, since it is no longer the basis for the proposal being looked at in SG4, and we are not planning to use the TS process for its replacement.

SG6: Numerics (Kretz/Lippincott/McFarlane)

SG6 met for one afternoon, and, remarkably, had time to consider every paper for which we had a champion.

P2746r7 “Deprecate and Replace Fenv Rounding Modes”
We gave feedback to questions that came up after it had already left SG6 a few meetings earlier;

P3375r2 “Reproducible floating-point results”
The author asks for a delay; work continues.

P3565r0 “Virtual floating-point values”
We want to think longer about this paper; it’s subtle. We have asked for a revision with more examples, and more importantly, more time to think it through.

P3495r0 “Remarks on Basic Statistics, (P1708r9)”
No champion of P1708 was present. We asked to see a new revision of that paper, and made recommendations to the author. (P1708r9 was already forwarded by SG6; based on these new concerns, we are pulling it back.)

P3045r5 “Quantities and Units Library”
Work and review continues. We’re liking it.

SG7: Compile-time programming (Dusíková/Snyder)

SG7 saw three papers proposing improvements to reflection targeting C++29:

Papers that were forwarded to EWG / LEWG

- P3385R3: Attributes Reflection — approved by SG7 and forwarded to EWG

Papers reviewed (require more work)

- P3420R1: Reflection of Templates — this revision added functions for replacement and projection of names in templates.
- P3294R3: Code Injection with Construct Sequences — was previously seen by EWG but brought back to SG7 for review of its updated design. Formerly called Token Sequences, these were rebranded to Construct Sequences to better convey how they can include reflections.

SG9: Ranges (Hollman/Müller)

SG9 met on Monday and Tuesday. We forwarded three papers:

- P3059R1 Making user-defined constructors of view iterators/sentinels private, which makes user-defined constructors of view iterators/sentinels private (we want it to be a DR for C++20!)

- P3230R1 views::unchecked_(take|drop), which adds views::unchecked_take and views::unchecked_drop
- P3117R1 Extending Conditionally Borrowed, which extends conditionally borrowed

We gave feedback on two papers and expect them to be forwarded at the next meeting:

- P3351R2 views::scan, which adds views::scan a lazy version of the inclusive_scan algorithm
- P3411R1 any_view, which adds a type-erased view

While discussing P3555R0 An infinite range concept, which adds a concept to detect infinite ranges, we realized that infinite ranges don't even exist to begin with, so a lot more work is needed there. The author will attempt to tackle that problem and we expect to be talking about it a lot in the next year(s).

We also discussed and mostly rejected P3329R0 Healing the C++ Filter View, which wants to solve some beginner pitfalls with views::filter. However, the design space is very constrained with regards to backwards compatibility and silent semantic changes of core concepts, so we encouraged more work by the author to come up with a separate view that solves these problems without being called views::filter.

SG10: Feature test (Revzin/Wakely)

SG10 did not meet.

SG14: Games & low latency (Wong)

SG14 meets online on the 2nd Wednesday of every month to discuss low latency, games, finance, and embedded matters. Lately we have been working on Patrice's Games omnibus features paper.

SG15: Tooling (Spencer/Boeckel)

SG15 met for one session during Hagenberg to discuss how we plan to ship our deliverables now that we're not doing the Ecosystem IS. We decided to use the relatively new whitepaper process which has a significantly lower overhead than an IS, but some clarifications on the process are needed.

Papers reviewed

- [P1180R0](https://wg21.link/P1180R0): Response to P1156 — We discussed the issues that header units can cause with build systems that use approximate scanning rather than precise scanning. The macros they bring in cause problems if they would impact other imports. CMake, MSVC, and Clang's scanner don't have this problem since they use precise scanning, so we're waiting for field experience from build systems that do approximate scanning to see what's actually needed here.

SG16: Unicode (Honermann/Downey)

SG16 did not meet in Hagenberg due to the usual concerns about quorum and the potential to contribute to lack of quorum for other groups. We continue to meet twice a month with plenty of work ahead of us.

SG17: EWG Incubator (Keane/Touton)

EWGI Intended to meet all day Wednesday, and had 11 papers scheduled. We were able to see 7 papers total, due to quorum difficulties. Of these 7, 4 were forwarded, 3 were given feedback:

Forwarded:
P3412 String Interpolation
P3424 Define Delete with Throwing Exception Specification
P2490 Zero-overhead exception stack traces
P3588 Allow static data members in local and unnamed classes

Feedback Given:
P3550 Imports cannot…
P3530 Intrinsic for reading uninitialized memory
P3568 break label; and continue label;

SG18: LEWG Incubator (Baker/Liber)

SG18 did not meet.
We plan on resuming telecons and meeting in Sofia.

SG19: Machine Learning (Wong/Ratzloff)

SG19 meets online on the 2nd Thursday of every month to discuss ML. Lately we have been working on Stats and Graphs. Stats is in conjunction with SG6 but builds the foundation for most ML algorithms. Graph is not Numerics related and is used in ML for Retrieval Augmented Generation especially in Relational Vector AI databases.

SG20: Education (van Winkel/Sattler)

SG20 did not meet.

SG21: Contracts (Spicer/Doumler)

In Hagenberg, SG21 met for one day (Thursday) and discussed two papers that were proposing post-C++26 extensions to P2900.

The first paper, P3583R0, proposes an approach to make contracts work on function pointers. SG21 concluded that the approach doesn't quite work as proposed and the paper needs a revision; feedback to the author was given. This paper is now the *third* independent attempt to add contracts on function pointers, highlighting how hard this problem really is.

The second paper, P3400, proposes to add labels to Contracts, which addresses many additional use cases that P2900 on its own does not yet fully address. For example, you can have a label to specify in code that a particular contract assertion must always be enforced and can never be ignored or observed. This paper, too, needs a revision; feedback to the author was given.

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

SG22 has not met this week.

SG23: Safety/Security (Orr/[open])

SG23 met for a day on Tuesday, with attendance affected by the discussions elsewhere on Contracts and Profiles

We forwarded on three papers:
One to LEWG:

- P3566R0 "You shall not pass char* - Safety concerns working with unbounded null-terminated strings"

Two to EWG:

- P3442R1 "[[invalidate_dereferencing]] attribute"
- P3541R1 "Violation handlers vs noexcept", recommending that the issues described here are addressed before P3081 and P3100 are discussed further.

We had a presentation on security profiles from Ulfar Erlingsson and encouraged further work in this direction

We gave feedback to the authors for revising:

- P3356R0 "non_invalidating_vector"
- P3402R2 "A Safety Profile Verifying Class Initialization"

ABI Group (Vandevoorde/Merril)

ABI group did not meet.

Admin (Liber)

We had 201 attendees, of which 28 were guests. 128 appeared in person, and 73 were virtual.

The next three mailings are scheduled for:

3/17 post-Hagenberg
4/16
5/19 pre-Sofia

There will be telecons scheduled soon to discuss solutions to the github issues being locked away during the latter part of the meeting, covering long term and Sofia (if we can't get the longer term solution done by then). Announcement for this will be made on -admin.

We have a draft of wiki up for a review. For Sofia we may try a new wiki. It has per user access. More details on this in the coming months.

We are working on a paper system that is providing a more full fledged search engine.

There will be a new login for Sofia in the next few weeks. Keep an eye out for an email to the reflector.

Evolution (Bastien/Dusíková/Keane)

This week, EWG saw 56 papers and resolved 7 issues. The objective was to finalize C++26 features, "all bugs in". Meetings going forward will have EWG fixing any bugs for C++26, and reviewing features for C++29.
おつかれさまです！🙏

📝 Contracts

⏩ contracts are still in C++26, polls on the P2900 tracker
This week we:

- reviewed significant feedback
- disallowed pre/post contracts on virtual functions entirely
- contended, but unchanged: exceptions when they leave predicate evaluation

📈 Consensus on contracts has increased since the last meeting. 📈
Thank you to all the authors, and everyone who's provided feedback! Contracts in C++26 are a huge deal for programmers who want to increase their code's correctness and quality.
Papers considered:

- P3573 — Contract concerns
- P3506 — P2900 Is Still not Ready for C++26
- P3591 — Contextualizing Concerns About Contracts
- P3500 — Are Contracts "safe"?

- P3577 — Require a non-throwing default contract-violation handler
- P3229 — Making erroneous behaviour compatible with Contracts
- P3269 — Do Not Ship Contracts as a TS
- P3265 — Ship Contracts in a TS

📋 Profiles

We reviewed the following papers on profiles:

- ✅ P3589 — C++ Profiles: The Framework
- ✅ P3611 — Dealing with pointer errors: Separating static and dynamic checking
- ✅ P3081 — Core safety profiles for C++26
- ❌ P3586 — The Plethora of Problems With Profiles
- ❌ P3543 — Response to Core Safety Profiles (P3081)
- ✅ P3447 — Profiles syntax
- ❌ P3599 — Initial Implicit Contract Assertions
- ✅ P3274 — A framework for Profiles development
- ❌ P3541 — Violation handlers vs `noexcept`

For profiles, we voted the following:
Pursue a language safety white paper in the C++26 timeframe containing systematic treatment of core language Undefined Behavior in C++, covering Erroneous Behavior, Profiles, and Contracts. Appoint Herb and Gašper as editors.
What does this mean?
Many people felt that what profiles are trying to address (security, safety) is hugely critical... yet profiles as they stand today are not ready. The C++26 train is leaving the station, but we want progress, now!
Enter white papers: it's a tool that ISO is now encouraging us to use, whereby we need WG21 plenary approval and SC22 approval, and then we have an approved white paper. The implication: We can get profiles in a white paper, implemented in compilers (behind a flag) before C++26 is finalized.
How does that work? White papers are a lightweight TS, or a heavy paper. The way we manage this is fairly open and we heard concerns which Herb and Gašper will suggest ways to address. For now, we have them as editors, they choose what goes in the white paper, and our hope is that they are trusted by everyone to do so while increasing consensus. EWG will see updates, forward them to CWG, then to plenary, then SC22, with votes at each stop. This is actually lightweight, and will allow rapid improvements and feedback. One way to address issues brought up is to have a git repo on github.com/cplusplus where the white paper is developed, with great commit messages, with periodic reports (say, monthly), and with periodic EWG telecons to review (say, monthly). Herb and Gašper will publish details soon.
Of course, we cannot take implementations for granted. A white paper is a new tool, but we can't be shipping unstable white papers every week and expect implementations to ship them. But we know white papers will be lower overhead than a TS. We therefore expect that white paper editors will be mindful editors.
What is expected in the white paper? systematic treatment of core language Undefined Behavior in C++, covering Erroneous Behavior, Profiles, and Contracts. This is broad! The final white paper doesn't need to include all of these, but it's the scope that was given to them. The idea is to try to comprehensively address security and safety issues, and do so with a comprehensive design. The scope given to the white paper allows aligning these topics, together. Contracts are in C++26, but profiles will likely be usable in a production compiler before contracts are usable behind -std=c++26.

This is great for contracts as well! It means that we'll be able to address perceived shortcomings of contracts with respect to safety rapidly, with direct feedback, in the C++29 timeframe thanks to the white paper.
Why Herb and Gašper? Throughout the years they've shown themselves to be mediators, and great at obtaining consensus from groups who have a hard time agreeing. Herb is indefatigable, and has in the last few months put in incredible efforts in advancing a variety of proposals. Gašper goes into details and synthesizes it into consensus, we've seen this in action in contracts to help bridge gaps that seemed unbridgeable. The thinking is that they complement each other, and are well trusted by a variety of committee members to fairly take feedback and advance this critical topic.
This is a huge undertaking for both of them. Herb has signed up to dedicate 1.5 to 2 years of his life almost full-time on improving C++ safety and security. Thank you Herb! While Gašper wasn't here for this meeting, he's also signed up for significant work. Thank you!

 Various C++26 papers

- ✅ P2841 — Concept and variable-template template-parameters
- ✅ P2786 — Trivial Relocatability For C++26
- ✅ P3310 — Solving issues introduced by relaxed template template parameter matching
- ✅ P2719 — Type-aware allocation and deallocation functions
- ✅ P2866 — Remove Deprecated Volatile Features From C++26
- ✅ P2843 — Preprocessing is never undefined
- ✅ P2287 — Designated-initializers for base classes
- ✅ P3501 — The ad-dressing of cats (note: no cats were present)
- ✅ P0149 — Generalised member pointers
- ✅ P3618 — Allow attaching main to the global module
- ✅ P2825 — Overload resolution hook: declcall(unevaluated-call-expression)
- ✅ P3492 — Sized deallocation for placement new
- ✅ P2952 — auto& operator=(X&&) = default
- ✅ P1967 — #embed - a simple, scannable preprocessor-based resource acquisition method
- ✅ P3540 — `#embed` offset parameter
- ✅ P1306 — Expansion statements
- ✅ P3471 — Standard Library Hardening: forward to CWG for inclusion in C++26 this is a huge deal for safety and security
- ✅ P3111 — Atomic Reduction Operations
- ✅ Transactional Memory TS to a white paper
- ♻️ P3006 — Launder less
- ♻️ P0876 — fiber_context - fibers without scheduler
- ✅ P3074 — trivial unions (was std::uninitialized)
- 🌊 P3568 — break label; and continue label; (only input for WG14, with not strong preference either direction, but interested in this feature)
- ⚠️ P2434 — Nondeterministic pointer provenance (prospective pointer value was taken out, otherwise back in CWG)

- ❌ P2883 — `offsetof` Should Be A Keyword In C++26
- ❌ P3477 — There are exactly 8 bits in a byte
- ❌ P3232 — User-defined erroneous behaviour
- ❌ P3439 — Chained comparisons: Safe, correct, efficient

Paper P2843 "Preprocessing is never undefined" above resolves the following issues:

- ✅ CWG2577 — Undefined behavior for preprocessing directives in macro arguments
- ✅ CWG2581 — Undefined behavior for predefined macros
- ✅ CWG2580 — Undefined behavior with #line
- ✅ CWG2579 — Undefined behavior when token pasting does not create a preprocessing token
- ✅ CWG2578 — Undefined behavior when creating an invalid string literal via stringizing
- ✅ CWG2576 — Undefined behavior with macro-expanded #include directives
- ✅ CWG2575 — Undefined behavior when macro-replacing "defined" operator

🪞 Reflection

Reflection: "the renaissance of C++"
Reflection is still in C++26! This week we:

- added access control, need to opt-in to unchecked
- add function parameter reflection
- add immediate-escalating expressions

Papers seen:

- ❌ P3587 — Reconsider reflection access for C++26
- ✅ P3547 — Modeling Access Control With Reflection
- ✅ P3096 — Function Parameter Reflection in Reflection for C++26
- ✅ P3496 — Immediate-Escalating Expressions
- ❌ P3569 — Split define_aggregate from Reflection
- ✅ D2996R10 — Reflection for C++26 (changes back from CWG which needs our approval)

🧊 `constexpr`

- ✅ P3533 — constexpr virtual inheritance
- ✅ P3590 — Constexpr Coroutines Burdens
- ✅ P3367 — constexpr coroutines voted, but for C++29

🐾 Pattern matching

Pattern matching: "We hardly knew ye"

Pattern matching did not get consensus, but it was extremely close. Attendees felt that it wasn't quite ready for C++26. Let's get it in C++29!
Main papers which were discussed:

- ✅ P3572 — Pattern matching
- ✅ P2688 — Pattern Matching: `match` Expression

Library parts, not discussed this meeting:

- P3527 — Pattern Matching: **variant-like** and `std::expected`
- P3521 — Pattern Matching: Customization Point for Open Sum Types

EWG straw polls

1. Change the ship vehicle for Transaction Memory TSv2 (N4923) from a TS to a white paper. Appoint Michael Wong as editor.

Herb Sutter points out we already approved the content of the TS, the change is just to the ship vehicle.

No objection to unanimous consent.
Motion passes.

Library Evolution (Levi/Fracassi/Weis)

LEWG met during the full week, and reviewed 45 papers. We've been working mostly on improvements and fixes to our main features targeting C++26, but we also had a chance to have some smaller neat additions!

Main Topics Discussed

(for topics already forwarded, we discussed improvements / fixes)

- Sender Receiver (P2300 (forwarded in St. Louis))
- Reflection (P2996 (targeting Sofia))
- SIMD (P1928 (forwarded in wrocław
- Trivial Relocatability (P2786 (forwarded in Tokyo)
- Concurrent Qs (P0260 (forwarded during this meeting))
- Standard Library Hardening (P3471 forwarded during this meeting)
- Ranges (P0896, P2214R1, P2214R2 (accepted for C++20, additions since)
- P3348R1: C++26 should refer to C23 not C17 — rebasing C++ on C! (thank you, Jonathan Wakely!)

Papers forwarded to LWG

Reflection

- ✅ P3394R1: Annotations for Reflection — new feature allowing users to append information for reflection to build upon
- ✅ P3293R1: Splicing a base class subobject — addresses concerns
- ✅ P3491R1: define_static_(string,object,array) — adds compile time structures improving usability of reflection
- ✅ P3547R1: Modeling Access Control With Reflection — address concerns raised regarding access
- ✅ P3560R0: Error Handling in Reflection — adds `std::meta::exception`, utilize constexpr exceptions to improve error reporting in reflection

Senders Receivers

- ✅ P2079R7: Parallel Scheduler (was: System Execution Context) — addition for managing execution context
- ✅ P3149R8: async_scope -- Creating scopes for non-sequential concurrency — addition for managing async-sync integration
- ✅ P3296R3: let_async_scope — managing async-sync integration, designed to provide simpler default
- ✅ P3481R2: std::execution::bulk() issues — improvements to utility (joined paper with "P3564R0 Make the concurrent forward progress guarantee usable in `bulk`" thank you to the authors for working together to merge the papers!)
- ✅ P3570R0: Optional variants in sender/receiver — utility for improved integration with coroutines
- ✅ P3164R3: Early Diagnostics for Sender Expressions — improved errors!
- ✅ P3557R0: High-Quality Sender Diagnostics with Constexpr Exceptions — utilize constexpr exceptions for senders!
- ✅ P3425R2: Reducing operation-state sizes for sub-object child operations — optimization
- ✅ P3433R0: Allocator Support for Operation States — improvement

Safety

- ✅ P3471R3: Standard Library Hardening — turning preconditions into hardened ones, provides stronger guarantees.

Other Features

- ✅ P3516R0: Uninitialized algorithms for relocation — library interface for Relocatability
- ✅ P2988R10: std::optional<T&> — adding support for ref types in optional
- ✅ P0260R15: C++ Concurrent Queues — concurrent container!
- ✅ P3179R6: C++ parallel range algorithms

- ✅ P3070R2: Formatting ~~enums~~ (was `enums` only, extended to all user defined types) — easier way to define formatters for users
- ✅ P3111R3: Atomic Reduction Operations — API extension
- ✅ P3383R1: mdspan.at() — API addition
- ✅ P3044R0: sub-string_view from string — API addition
- ✅ P3060R2: Add std::views::indices(n) — avoid off by one
- ✅ P1317R1: Remove return type deduction in std::apply — fixes
- ✅ P3623R0: Add noexcept to [iterator.range] (LWG 3537) —
- ✅ P3567R0: `flat_meow` Fixes — fixes
- ✅ P3016R5: Resolve inconsistencies in begin/end for valarray and braced initializer lists — fixes
- ✅ P3037R4: constexpr std::shared_ptr — extension
- ✅ P3416R2: exception_ptr_cast: Add && = delete overload — fixes
- ✅ P2319R4: Prevent path presentation problems — API update (Breaking Change, fixes `filesystem::path`)

Papers / issues sent from LWG seen by LEWG

- ✅ P3019R13: Vocabulary Types for Composite Class Design — apply design changes, send back to LWG
- ✅ P2019R7: Thread Attributes — apply SG16 recommendation, send back to LWG
- ✅ P2663R6: Proposal to support interleaved complex values in std::simd — approved, sent back to LWG
- ✅ P2664R9: Proposal to extend std::simd with permutation API — approved, sent back to LWG
- ✅ P2993R3: Extend <bit> header function with overloads for std::simd — approved, sent back to LWG

Papers that got feedback and will be seen again by LEWG

- 🔄 P3552R0: Add a Coroutine Lazy Type
- 🔄 P3380R1: Extending support for class types as non-type template parameters — no implementation, requires more work (reflection)

Papers that did not get consensus

- ❌ P3559R0: Trivial relocation: One trait or two?
- ❌ P3477R1: There are exactly 8 bits in a byte
- ❌ P3160R2: An allocator-aware `inplace_vector`

Policies discussion

We will resume our discussion about policies in Sofia!

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

We will discuss the following topics:

- Explicit Constructors
- Overload resolution with concepts
- Unicode support (Collaboration with SG16)

Worth noting that Evolution Work Group (EWG) have also introduced policies, and have accepted: “SD-10: Language Evolution (EWG) Principles” during Wroclaw.

Evening Sessions

In addition to the work meeting, we had two evening sessions during the week (initiated by WG21 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 "Relocatability") , or for introducing topics which are not procedurally related but are relevant to WG21 (such is the case for “Perspectives on Contracts").

- 🔎 Tuesday: “Concurrent Queues”

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 Sofia! ˋ(~ᵕ~)ˊ

Core (Maurer/Merrill/Caves)

Core met the whole week. We also had evening sessions for east coast participants. Reflection is not on the straw polls page. We will have telecons every 2 weeks until Sofia to work on reflection. Contracts are on the straw polls page. Thank you to Brian and Roger for scribing.

#embed paper was readied up, but concerns were raised since then. We can address those concerns via core issues. We suggest to forward as is and I will open any core issues needed. For trivial relocation we have changes to annex C that are not accurate. We can amend the motions to apply the change minus annex C, or we can ask the editor not to apply that particular change.

It is agreed that the change needed for trivial relocation is an editorial change. There is no normative text in annex C.

There were concerns with trivial unions regarding corner cases. CWG thinks we should continue with the motion and address those cases separately.

CWG polls

1. Accept as Defect Reports and apply the proposed resolutions of all issues in P3638R0 (Core Language Working Group "ready" Issues for the February, 2025 meeting) to the C++ Working Paper.

No discussion.
No objection to unanimous consent.
Motion passes

2. Apply the changes in P3542R0 (Abolish the term "converting constructor") to the C++ Working Paper.

No discussion.
No objection to unanimous consent.
Motion passes

3. Apply the changes in P3074R7 (trivial unions (was std::uninitialized)) to the C++ Working Paper.

No discussion.
Objection to unanimous consent.

Herb Sutter explains the voting rules.

In favour : 65 (42 in person + 23 online)
Opposed : 9 (8 in person + 1 online)
Abstain : 27 (12 in person + 15 online)

Motion passes

4. Apply the changes in P1494R5 (Partial program correctness) to the C++ Working Paper.

No discussion.
No objection to unanimous consent.
Motion passes

5. Apply the changes in P2900R14 (Contracts for C++) to the C++ Working Paper.

No discussion.
Objection to unanimous consent.

In favour : 100 (62 in person + 38 online)

Opposed : 14 (9 in person + 5 online)
Abstain : 12 (5 in person + 7 online)

Motion passes.

Herb thanks everyone for their effort. Raising issues and giving feedback is appreciated when we are trying to standardise a feature.

6. Apply the changes in P3475R2 (Defang and deprecate memory_order::consume) to the C++ Working Paper.

No discussion.
No objection to unanimous consent.
Motion passes

7. Apply the changes in P2841R7 (Concept and variable-template template-parameters) to the C++ Working Paper.

No discussion.
Objection to unanimous consent.

In favour : 80 (54 in person + 26 online)
Opposed : 3 (3 in person + 0 online)
Abstain : 23 (8 in person + 15 online)

Motion passes.

8. Apply the changes in P2786R13 (Trivial Relocatability For C++26) to the C++ Working Paper.

No discussion.
Objection to unanimous consent.

In favour : 76 (54 in person + 22 online)
Opposed : 11 (8 in person + 3 online)
Abstain : 30 (11 in person + 19 online)

Motion passes.

9. Apply the changes in P1967R14 (#embed - a simple, scannable preprocessor-based resource acquisition method) to the C++ Working Paper.

No discussion.
Objection to unanimous consent.

In favour : 87 (59 in person + 28 online)
Opposed : 1 (0 in person + 1 online)
Abstain : 25 (13 in person + 12 online)

Motion passes.

Library (Wakely/Garland/Kuhl)

LWG had a busy week continuing to review simd extensions, contracts, reflection, ranges additions, and senders-receivers fixes. As we reach the end of the cycle we were in a position to review and move at least a couple smaller papers LEWG forwarded this meeting. Given that the design is now frozen we'll be putting together a telecon schedule and a plan to address as much of the work as we can.

Special thanks to Dietmar and Neelofer Banglawala for scribing, Christian Trott for chairing and scribing -- and of course Tomasz Kaminski and Tim Song for working evenings with authors to cleanup proposals. And we'll humbly accept Hana's award as most pleasant room -- credit to Jonathan Coe for his calming words about keeping interactions civil.

LWG polls

1. Apply the changes for all Tentatively Ready issues in P3615R0 (C++ Standard Library Ready Issues to be moved in Hagenberg, Feb. 2025) to the C++ working paper.

No discussion.
No objection to unanimous consent.
Motion passes

2. Apply the changes in P2830R9 (Standardized Constexpr Type Ordering) to the C++ working paper.

A concern is raised that this paper has not been seen by Core. No objection to postponing this poll until Sofia. Poll withdrawn.

3. Apply the changes in P3137R3 (views::to_input) to the C++ working paper.

No discussion.
No objection to unanimous consent.
Motion passes

4. Apply the changes in P0472R3 (Put std::monostate in ⟨utility⟩) to the C++ working paper.

LWG chair points out this is the same poll as in Wroclaw, but with the correct revision number.

No objection to unanimous consent.
Motion passes

5. Apply the changes in P3349R1 (Converting contiguous iterators to pointers) to the C++ working paper.

No discussion.
No objection to unanimous consent.

Motion passes

6. Apply the changes in P3372R3 (constexpr containers and adaptors) to the C++ working paper.

No discussion.
No objection to unanimous consent.
Motion passes

7. Apply the changes in P3378R2 (constexpr exception types) to the C++ working paper.

No discussion.
No objection to unanimous consent.
Motion passes

8. Apply the changes in P3441R2 (Rename simd_split to simd_chunk) to the C++ working paper.

No discussion.
No objection to unanimous consent.
Motion passes

9. Apply the changes in P3287R3 (Exploration of namespaces for std::simd) to the C++ working paper.

No discussion.
No objection to unanimous consent.
Motion passes

10. Apply the changes in P2976R1 (Freestanding Library: algorithm, numeric, and random) to the C++ working paper.

No discussion.
No objection to unanimous consent.
Motion passes

11. Apply the changes in P3430R3 (simd issues: explicit, unsequenced, identity-element position, and members of disabled simd) to the C++ working paper.

No discussion.
No objection to unanimous consent.
Motion passes

12. Apply the changes in P2663R7 (Interleaved complex values support in std::simd) to the C++ working paper.

No discussion.
No objection to unanimous consent.
Motion passes

13. Apply the changes in P2933R4 (Extend ⟨bit⟩ header function with overloads for std::simd) to the C++ working paper.

No discussion.
No objection to unanimous consent.
Motion passes

14. Apply the changes in P2846R6 (reserve_hint: Eagerly reserving memory for not-quite-sized lazy ranges) to the C++ working paper.

No discussion.
Objection to unanimous consent.

In favour : 61 (46 in person + 15 online)
Opposed : 3 (2 in person + 1 online)
Abstain : 38 (17 in person + 21 online)

Motion passes.

15. Apply the changes in P3471R4 (Standard Library Hardening) to the C++ working paper.

No discussion.
Objection to unanimous consent.

In favour : 101 (64 in person + 37 online)
Opposed : 1 (1 in person + 0 online)
Abstain : 11 (5 in person + 6 online)

Motion passes.

16. Apply the changes in P0447R28 (Introduction of std::hive to the standard library) to the C++ working paper.

A concern is raised regarding implementation of this paper.
It has been discussed in LWG and LEWG.
Objection to unanimous consent.

In favour : 63 (41 in person + 22 online)
Opposed : 10 (8 in person + 2 online)
Abstain : 38 (23 in person + 15 online)

Motion passes.

17. Apply the changes in P3019R14 (indirect and polymorphic: Vocabulary Types for Composite Class Design) to the C++ working paper.

We approved an earlier version in Wroclaw. Some issues were noticed during the application. Further issues reported by users. We saw this again this week.

There is a technical comment in the room. I has been discussed in LEWG.

Objection to unanimous consent.

In favour : 75 (50 in person + 25 online)
Opposed : 4 (4 in person + 0 online)
Abstain : 29 (17 in person + 12 online)

Motion passes.

Direction Group (Wong)

A rough description of the DG, its composition, membership, and charter
https://www.open-std.org/jtc1/sc22/wg21/docs/papers/2022/p2000r4.pdf

You can contact us at direction@lists.isocpp.org.
You can write a paper directed to us too or just talk to any of us.

Howard Howard Hinnant has retired. We appreciate his many years of contributions.
Herb Sutter thanks Howard for his years of service.
Standing ovation in the room.

We are looking for Profiles and Contracts to work together for the future.
As called out in the Wroclaw DG report, we continue to urge all haste to support Safety Directions.
Upon the completion of C++26, we will put out our usual forward-looking paper for C++29

We noted proposals are taking longer and longer, sometimes not being noticed until the last minute, we will discuss what can be done.
Our groups are huge, with many interrelated moving parts, highest stress during release cutoff times, please be extra patient with each other, don't take things personally, it is never the end as much as you think because there are a lot of moving parts.

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.

- 2025-03-17: Post-Hagenberg
- 2025-05-19: Pre-Sophia

8.3 Plans for the future

No discussion.

8.4 Next and following meetings

- 2025-06-16/21: Sofia, Bulgaria (N4991)
- 2025-11-03/08: Kona, HI, USA (N4977)

Vassil Vassilev presents Sofia meeting information. Please register as it helps the organisers.

9. Adjournment

Meeting adjourned at 10:53 AM UTC+1.

10. Attendance

Attendee	NB
Adams, Michael	SCC
Adelstein Lelbach, Bryce	ANSI
Alday, Juan	ANSI
Alexandrescu, Andrei	ANSI
André Brand	DIN
Andreas Hametner	ASI
Arutyunyan, Ruslan	ANSI
Bagdonas, Ignas	BSI
Baker, Billy	ANSI
Baker, Lewis	ANSI
Balog, Pal	ANSI
Banglawala, Neelofer	BSI

Attendee	NB
Bastien, Jean-Francois	SCC
Bauman, Jon	ANSI
Benetkiewicz, Paweł	PKN
Berne, Joshua	ANSI
Bertolt Mildner	ASI
Bi, Brian	ANSI
Bindels, P.G.H.	NEN
Birbacher, Frank	ANSI
Blackwell, Bianca	SCC
Boeckel, Ben	ANSI
Bonaventura, Xavier	DIN
Brown, Bret	ANSI
Brown, Walter E.	SII
Butler, Matthew	ANSI
Büttner, Sebastian	ANSI
ÇAĞRI, Murat Can	TSE
Cardoso de Souza Rodrigues, Guilherme	ASI
Caves, Jonathan	ANSI
CHARLES, Hussong	JISC
Chen, Jolly	SNV
Chen, Yuxuan	ANSI
Childers, Wyatt	ANSI
Chochlik, Matus	UNMS SR
Clemens Scharfen	ASI
Coe, Jonathan Brian	BSI
Craig, Benjamin	ANSI
Craig, Philip	BSI
Cranmer, Joshua	ANSI
D'Angelo, Giuseppe	ANSI
DaCamara, Cameron	ANSI
Davidson, Guy	BSI
Delfino, Gianluca	UNI
Dionne, Louis	SCC

Attendee	NB
Dominik Nussbaumer	ASI
Douglas, Niall	NSAI
Downey, Steve	ANSI
Dusikova, Hana	UNMZ
Engert, Daniela	DIN
Erlingsson, Ulfar	ANSI
Estell, Khalil	ANSI
Falcou, Joël	AFNOR
Fevold, Jake	ANSI
Floyd, Paul	ANSI
Foco, Marco	UNI
Fracassi, Fabio	DIN
García Sánchez, José Daniel	UNE
Garland, Jeff	ANSI
Genovese, Walter	ANSI
Gill, Mungo	NSAI
Giroux, Olivier	ANSI
Goodspeed, Nathaniel	ANSI
Gruber, Bernhard	ANSI
Gustafsson, Bengt	SIS
Halpern, Pablo	ANSI
Hauswedell, Hannes	IST
Hava, Michael Florian	ASI
Herring, Davis	ANSI
Hoemmen, Mark	ANSI
Hollman, Daisy	ANSI
Honermann, Tom	ANSI
Hughes, Lori	ANSI
Hunt, Oliver	ANSI
Izvekov, Matheus	ABNT
Jabot, Corentin	AFNOR
Joachim Kuebart	DIN
Josuttis, Nicolai	DIN
KAMINISKI, Tomasz	AFNOR
Kathryn Butler	ANSI

Attendee	NB
Kawulak, Robert	PKN
Keane, Erich	ANSI
Keir, Paul	BSI
Khlebnikov, Rostislav	ANSI
Khyzha, Artem	BSI
Klauser, Nikolas	ANSI
Koeppe, Thomas	ANSI
Kosunen, Elias	SFS
Kozicki, Bronek	BSI
Kretz, Matthias	DIN
Krzemienski, Andrzej	PKN
Kuhl, Dietmar	ANSI
Kulczycki, Peter	ASI
Lakos, John	ANSI
Larson, Brad	ANSI
Lauko, Henrich	UNMZ
Laverdière-Papineau, Marc-André	ANSI
Levi, Inbal	SII
Li, Yihe	ANSI
Liber, Nevin	ANSI
Lippincott, Lisa	ANSI
Machutova, Jana	UNMZ
Majumder, Abhilash	ANSI
Maness, Wesley	ANSI
Marr, Greg	ANSI
Maryam Karampour	SCC
Maurer, Jens	ANSI
Meerwald, Christof	ASI
Mejstrik, Thomas	ASI
Meredith, Alisdair	ANSI
Merrill, Jason	ANSI
Michael, Maged	ANSI
Mohamed Ayoub AKKAOU	UNMZ
Morales, Nicolas	ANSI
Moschovakos, Paris	SNV

Attendee	NB
Mueller, Gideon	ANSI
Müller, Jonathan	DIN
Murzin, Sergei	ANSI
Nadolski, Maikel	ANSI
Nathan Myers	
Naumann, Axel	SNV
Neatu, Darius	ASRO
Niebler, Eric	ANSI
Nolan, Edward	ANSI
O'Dwyer, Arthur	ANSI
Orr, Roger	BSI
Owen, Nathan	ANSI
Park, Michael	SCC
Paul Groke	ASI
Persson, Jonas	SIS
Peter Feichtinger	ASI
Petersen, Ian	ANSI
Philipp Zimmermann	DIN
Polukhin, Anton	GOST R
Preney, Paul	SCC
Pusz, Mateusz	PKN
Radu Nichita	ASRO
Ranns, Nina Dinka	BSI
Regev, Ran	SII
Revzin, Barry	ANSI
Rifkin, Jeremy	ANSI
Rigault, Jean-Paul	AFNOR
Rivera Morell, René Ferdinand	ANSI
Robert Schimkowitsch	ASI
Ronkainen, Jari	SFS
Roy, Patrice	SCC
Ryan, Christopher	ANSI
Sandoe, Iain	BSI
Sankel, David	ANSI
Satle, Ankur	BIS

Attendee	NB
Sattler, Florian	ANSI
Schultke, Jan	ANSI
Scogland, Thomas	ANSI
Serebrennikov, Vladislav	ANSI
Simeon Kuran	ASI
Šimerda, Pavel	UNMZ
Simpson, Robert	ANSI
Snyder, Jeff	BSI
Song, Tim	ANSI
Spencer, Michael	ANSI
Spicer, John	ANSI
St. Amour, Bryan	SCC
Starosz, Sebastian	PKN
Stroustrup, Bjarne	ANSI
Sutter, Herb	ISO/IEC JTC 1/SC 22
Talbot, Alan	ANSI
Taylor, Matthew	BSI
Teeple, Doug	ANSI
TEODORESCU, Lucian Radu	ASRO
Teoh, Joon Nam	ANSI
Thomas Pollak	ASI
Tong, Hubert	SCC
Touton, James	ANSI
Towner, Daniel	ANSI
Trott, Christian	ANSI
Tsaousis-Seiras, Isidoros	ANSI
Vandevoorde, Daveed	ANSI
Varlamov, Konstantin	ANSI
Vasama, Lauri	SFS
Vasilev, Vasil	BDS
Vollmann, Detlef	SNV
Vormwald, Steven	ANSI
Voss, Michael	ANSI
Voutilainen, Ville	SFS

Attendee	NB
Wakely, Jonathan	ANSI
Walker, Kelly	ANSI
Wang, Mingxin	SAC
Waterloo, Jarrad	ANSI
Weis, Andreas	ANSI
Williams, Anthony	BSI
Wong, Jessica	ANSI
Wong, Michael	SCC
Xie, Hui	BSI
xu, chuanqi	SAC
Yuan, Zhihao	ANSI
Zissu, Andrei	SII
Zverovich, Victor	ANSI