WG21 November 2023 Hybrid meeting
Record of Discussion

ISO/IEC JTC1 SC22 WG21 P3061— 2023-11-28
Nina Dinka Ranns, dinka.ranns_at_gmail.com

Chair: John Spicer

6-11 November 2023, Kona, HI, USA

1. Opening activities

John Spicer opens the meeting at 08:32 AM GMT-10

1.1 Opening comments, welcome from host

John Spicer welcomes the group.

Welcome from the host.
Herb Sutter welcomes the group.
Herb Sutter : If you have any questions, please come to me or Jens Maurer.

John Spicer : This is no longer a joint meeting with INCITS. We will have an INCITS only virtual meeting early December.

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.

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 very first link on the wiki is the attendance sheet. Please fill it in.
John Spicer : Mattermost information is on the wiki in the persistent information section.

1.3 Introductions

New members introduce themselves.
John Spicer welcomes new members.

Herb Sutter : we currently have 26 NB participating actively, and at least 21 are present today.

1.4 Agenda review and approval

John Spicer presents the agenda.

The primary goals of this meeting will be work on C++26 features.

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	N4965	N49640

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

1.6 Approval of the minutes of the previous meetings

Meeting	Minutes
WG21 Varna	N4957
WG21 pre-Kona administrative telecon	N4967

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 : annual C++ Foundation members meeting will take place at 5pm today.

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

Jens Maurer presents meeting amenities
Jens Maurer presents the time schedule.

Visitors from a local school on Friday, please be welcoming.

If you are new, it's a good idea to attend a session of each subgroup to get an idea for how the committee works.

Jens Maurer presents evening session
Evening sessions are open to everyone, but they are informal and no binding polls will be taken.

Jens Maurer presents various meeting information

Jens Maurer presents room assignments

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.

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)

Another short meeting for SG1, about a dozen papers were seen. There's renewed interest over the reflector in solving the OOTA problem; considering a 2-day working meeting on this topic either at a future WG21 meeting or as a one-off. If you want to work on OOTA issues, please contact us.

SG2: Modules (Stone)

Currently dormant.

SG4: Networking (Snyder/Ažman)

We continue looking at the paper to adapt the Networking TS. Thank you to Dietmar for picking this up again. Still lots to do.

Herb Sutter : Can you mention if you have any material that is aiming for C++26.
Jeff Snyder : It's too early to tell if Networking will be ready for C++26. Lots to do and the original author is not involved.

SG5: Transactional memory (Boehm/Maurer)

SG5 is semi-dormant. Michael Wong is negotiating with ISO about publishing the last TS.

SG6: Numerics (Kretz/Lippincott/McFarlane)

SG6 met for a whole day on Tuesday. We looked at the following papers and provided feedback:

- P3003R0 "The design of a library of number concepts"
- P2980R0 "A motivation, scope, and plan for a physical quantities and units library"
- P2982R0 "`std::quantity` as a numeric type"
- P2746R3 "Deprecate and Replace Fenv Rounding Modes"
- P3018R0 "Low-Level Integer Arithmetic"

SG6 forwarded P3018R0 and we expect P2746R3 to still be on target for C++26.

We also looked at a Core issue:
- CWG2752 "Excess-precision floating-point literals"

SG6 was split on how excess precision in C++ should behave, especially in relation to C. In order to make progress, it seems SG6 needs a paper.

SG7: Compile-time programming (Dusikova/Vandevoorde)

SG7 met Friday morning. We started with two papers that reported on experience with value-based reflection as partially implemented Lock3 in a Clang fork. After that, we discussed P2996, which proposes to include a reduced set of value-based reflection features in C++26. SG7 voted unanimously to forward P2996 to EWG and LEWG to target C++26.

SG9: Ranges (Hollman/Carter)

SG9 met all day Monday in Kona. We looked at the following papers and provided feedback:

- P1030R6 — std::filesystem::path_view
- P1729R3 — Text Parsing

We forwarded the following papers to LEWG for their consideration:

- P2997R0 — Removing the common reference requirement from the indirectly invocable concept
- P2760R0 — A Plan for C++26 Ranges

We also discussed the following issues
- LWG3913 — ranges::const_iterator_t<range R> fails to accept arrays of unknown bound
- LWG3988 — Should as_const_view and basic_const_iterator provide base()?

SG10: Feature test (Revzin/Wakely)

No report.

SG12: Undefined and unspecified behavior (Open/Wong)

SG12 is dormant and is handled in EWG now.

SG14: Games & low latency (Wong)

SG14 did not meet this week. SG14 continues to have regular teleconferences.

SG15: Tooling (Spencer/Boeckel)

SG15 met for a day and a half and saw 8 papers.
One of them was forwarded to LEWG
We had one paper which will become the proposed ecosystem IS.
We decided on a roadmap of the remaining work that is needed for common tools to be able to build modules.

Alisdair Meredith : would it be wise to fold SG2 into tooling ?
Herb Sutter : SG2 is dormant, any design work is EWG. Any non tooling issues should go to EWG.

SG16: Unicode (Honermann/Brett/Downey)

SG16 (Unicode) did not meet in Kona this week but continues its regular twice monthly cadence of virtual meetings. We continue to have no shortage of work to do.

SG17: EWG Incubator (Keane/TBD)

Met Monday, Tuesday afternoon and discussed 8 Papers

Forwarded 5 Papers to EWG:

- P2992R0: Attribute [[discard]] and attributes on expressions
- P2971R1: Implication for C++
- P2573R1: delete("should have a reason");
- P0963R1: Structured binding declaration as a condition
- P2826R1: Replacement Functions

Gave feedback to 3 Papers, and expect to see again:

- P2946R0: A flexible solution to the problems of noexcept
- P2986R0: Generic Function Pointer
- P2830R1: constexpr type comparison

SG18: LEWG Incubator (Baker/Liber)

SG18 did not meet separately from LEWG. The chairs of SG18 filled 1/3 of the chairing/minuting duties in LEWG including the session on Policies for the library (P2267).

There were 4 papers tagged for SG18 that were triaged to need input from other study groups. Three of the four were seen in at least one study group this week.

P3018 Low-level integer arithmetic (SG6)
P3016 Resolve inconsistencies in begin/end for valarray and braced initializer lists
P2980 A motivation scope, and plan for a physical quantities and units library (SG6, SG23, LEWG)
P2951 Shadowing is good for safety (SG23)

SG19: Machine Learning (Wong/Reverdy)

SG19 hasn't met this week, but continues to have teleconferences.

SG20: Education (van Winkel)

SG20 did not meet in Kona, but we continue to have subgroup meetings and extend our education guidelines in per timezone editing work sessions.

SG21: Contracts (Spicer/Doumler)

SG21 was sitting Tuesday, Thursday and Friday morning in Kona and we had a very productive meeting. We saw the following papers:

- P2935R4: An Attribute-Like Syntax for Contracts
- P2961R2: A natural syntax for Contracts
- P2038R0: An Overview of Syntax Choices for Contracts
- P2896R0: Outstanding design questions for the Contracts MVP
- D2932R2: A Principled Approach to Open Design Questions for Contracts
- P2957R0 Contracts and coroutines

We achieved strong consensus on a new syntax for contract checks called the "natural syntax" that no longer uses double square brackets:

```
int f(const int x)
  pre (x != 1)
  post (r : r != 2)
{
  contract_assert (x != 3);
  return x;
}
```

We also decided that

- Precondition and postcondition checks on functions defaulted on their first declaration are ill-formed
- Precondition and postcondition checks on coroutines are ill-formed

We're on track with our roadmap P2695R1 to get Contracts ready in time for C++26. At this point, we only have a handful of unresolved design questions left:

- Contracts and implicit lambda captures
- Constant evaluation of contracts (constexpr/consteval)
- Contracts and virtual functions
- Contracts on multiple declarations of the same function
- Contracts and deduced exception specifications

We scheduled six SG21 telecons between now and Tokyo to deal with the above. By Tokyo (next committee meeting in March 2024) we are aiming to have a complete Contracts working paper P2900R2 and to forward to EWG, LEWG, and SG21 for review.

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

SG22 has not met this week. We had a couple of issues forwarded in our direction. We'll be aiming to have an SG22 meeting before the end of the calendar year, depending on the availability of authors and SG22 members.

SG23: Safety/Security (Orr/Craig)

We met for one day on Wednesday. SG23 had a presentation on P2947 contracts avoiding leaking - forwarded to SG15.

P2973 erroneous behaviour on missing returns from assignment operators: consensus to produce further work.

Informational presentation on P2981 physical quantity and units. No polls taken, it was for informational purposes.

P2995 safer range access. Author pursuing a different approach.

P2951 shadowing is good for safety - consensus against.

D3038 concrete suggestions for profiles - consensus that this should provide SG23 with a medium term framework. Bjarne collecting volunteers to work with him

ABI Group (Vandevoorde)

No report.

Admin (Liber)

Please mark the attendance sheet.
The cut off for the final mailing for this year is December 16.

Evolution (Bastien/Stone/Keane/Dusikova)

Language Evolution met for most of the week, spending one day on undefined behavior. We ran out of content for the last half-day.
In our report, we use the following emoji-aided language:

- 👎 — consensus to stop work (unless new information comes, warranting revisiting)
- 👍 — advance the paper to Core (or other groups), targeting C++26
- ♻️ — consensus to continue work in EWG
- 🚫 — the issue is not a defect
- ⬆️ — see above (contextual keyword, see above where it appears, not here)

C alignments, bug fixes, deprecation
Make C++ hurt less.

- 👎 — P2958R0: typeof and typeof_unqual (came to us from C23 but did not have consensus for addition on C++. That said, it is expected that implementations will support these new keywords in C++ mode to enable interoperability with C headers, it is therefore likely that existing practice warrants adding these to C++ in the future)
- 👎 — P2984R0: Reconsider Redeclaring static constexpr Data Members (We agreed to neither un-deprecate, nor remove the feature entirely)
- 👍 — P2865R3: Remove Deprecated Array Comparisons from C++26 (Removes surprising code)
- 👍 — P2864R1: Remove Deprecated Arithmetic Conversion on Enumerations From C++26 (Removes surprising code)
- 👍 — P2795R3: Erroneous behaviour for uninitialized reads (Erroneous behaviour is a promising way to make UB hurt less. That said, there are significant wording issues in Core. This paper introduces the concept of EB, and uses it in one location. It is a different way of addressing the issue from P2723r1, which the committee wants to address)
- ♻️ — P2973R0: Erroneous behaviour for missing return from assignment (The second application of EB. There was discussion on making missing returns ill-formed instead, but the committee was interested in using EB for this instead)
- 👍 — P2809R2: Trivial infinite loops are not Undefined Behavior (The paper proceeded to Core, with option #2 (the simpler definition of trivial infinite loops) being chosen)

- ♻️ — P0901R11: Size feedback in operator new (Interest in seeing this paper again, feedback was given to the authors)
- 👎 — P2971R1: Implication for C++ (There was mild opposition and many neutrals, but also significant interest from a small but vocal set of people. It is expected that this paper will come back with stronger justification to increase consensus)

Issue processing
Remove sharp edges.

- 👍 — CWG2561: Conversion to function pointer for lambda with explicit object parameter (Proceeded with its option #1)
- ♻️ — CWG2804: Lookup for determining rewrite targets (Lookup is difficult. We'll need a paper to address the issue in its entirety)
- 🚫 — CWG2548: Array prvalues and additive operators (Not a defect, but a paper could be written to justify this change)
- 🚫 — CWG2776: Substitution failure and implementation limits (Not a defect, but a paper could be written to justify this change)
- ♻️ — CWG2784: Unclear definition of member-designator for offsetof (Interested in pursuing, needs a paper)
- 👍 — CWG2733: Applying maybe_unused to a label (Accept the proposed wording)
- 🚫 — CWG2751: Order of destruction for parameters for operator functions (Not a defect, implementations can choose the strategy)
- 👍 — CWG2560: Parameter type determination in a requirement-parameter-list (Accept the proposed resolution, and apply it as a Defect Report to prior versions of C++)
- ♻️ — CWG2726: Alternative tokens appearing as attribute-tokens (A paper is forthcoming)
- ♻️ — CWG1699: Does befriending a class befriend its friends? (A paper is forthcoming)
- ♻️ — CWG2669: Lifetime extension for aggregate initialization (A paper is forthcoming)
- ♻️ — CWG2797: Meaning of "corresponds" for rewritten operator candidates (Related to CWG2804)
- 👍 — CWG2825: Range-based for statement using a braced-init-list (Accept the proposed resolution, and apply it as a Defect Report to prior versions of C++)

Features
Make C++ nicer.

- ♻️ — P1046R2: Automatically Generate More Operators (The author wrote a new draft paper P3039R0 Automatically Generate operator-> which narrows the scope of change and has strong support)
- ♻️ — P1045R1: constexpr Function Parameters (This paper was discussed as a potential language solution for the following paper. We would need an updated paper)
- ♻️ — P2781R3: std::constexpr_t (LEWG was interested in a language feature to resolve this usecase. EWG agrees, but needs papers to advance this topic)
- ♻️ — P3009R0: Injected class name in the base specifier list (Consensus was not particularly strong, but there was mild interest)
- ♻️ — P2994R0: On the Naming of Packs (Interest, needs an update)

- 👍 — P2662R2: Pack Indexing (Core pointed out a potential design oversight, EWG was OK with the potential closed design door)
- ♻️ — P2963R0: Ordering of constraints involving fold expressions (The author was encouraged to perform further work)
- ♻️ — P2686R2: constexpr structured bindings and references to constexpr variables (There is still strong interest in allowing structured bindings in constexpr, but doing so requires consideration around adding references that potentially dangle as constexpr. The room was not sure the design was comprehensible, further work was encouraged)
- 👍 — P2893R1: Variadic Friends (Forward to Core for C++26)
- 👍 — P2748R2: Disallow Binding a Returned Glvalue to a Temporary (Forward to Core for C++26)
- ♻️ — P2991R0: Stop Forcing std::move to Pessimize (Weak consensus, needs more work)
- 👍 — P2927R0: Observing exceptions stored in exception_ptr (EWG approves the direction, LEWG may proceed)
- 👎 — P0342R2: pessimize_hint (No consensus)
- ♻️ — P2989R0: A Simple Approach to Universal Template Parameters (Strong interest, the paper should be seen again. The precise syntax was not chosen yet, we will do so later)
- ♻️ — P2841R1: Concept Template Parameters (Feedback was provided)
- ♻️ — P2979R0: The Need for Design Policies in WG21 (We are interested in defining principles (or rules of thumb) and policies for the design and evolution of C++, with the goal of documenting our common understanding and simplifying our ongoing work, but without creating a bureaucracy)

Undefined behavior
Remove it, or at least document it.

- ♻️ — P2843R0: Preprocessing is never undefined (The paper surveys all UB in the preprocessor, and implementation behavior. It suggests solutions to each, making them not UB. There's strong support for this work. Alignment with the C committee has started)
- ⬆️ — CWG2577: UB for preprocessing directives in macro arguments
- ⬆️ — CWG2581: UB for predefined macros
- ⬆️ — CWG2580: UB with #line
- ⬆️ — CWG2579: UB when token pasting does not create a preprocessing token
- ⬆️ — CWG2578: UB when creating an invalid string literal via stringizing
- ⬆️ — CWG2576: UB with macro-expanded #include directives
- ⬆️ — CWG2575: UB when macro-replacing "defined" operator
- ♻️ — CWG2514: Modifying const subjects (Strong interest to resolve, but needs a paper)
- 👍 — P1705R1: Enumerating Core Undefined Behavior (This and the next paper will make their way to the standard. The editor can add them editorially. However, there is substantial wording in the explanation and examples, Core is therefore interested in reviewing the work. Collaboration is welcome. Note that this paper only documents

existing explicit Core UB. It does not change what it UB, nor does it document implicit UB, not library UB. It's a start to allow us to revisit what is UB)
- 👍 — P2234R1: Consider a UB and IF-NDR Audit (As above, this will add an index of IF-NDR, or an appendix with justification and examples)

What will EWG do in Tokyo?
東京は何をしましょうか。
- Continue improving C++.
- Focus on reflection.
- Prepare for focusing on contracts.

Given the light open workload, EWG will not host any telecons between now and Tokyo.

Jens Maurer : We had a meeting on Thursday after hours and we formulated a plan and agreement on how to proceed regarding UB and IFNDR issues. We will have two distinct lists - one for UB and one for IFNDR going into the annex allowing examples to be added.
From CWG perspective, I'm not happy adding that much material editorially, so I would want CWG to review it in some shape or form.

Library Evolution (Levi/Fracassi/Craig)

Features Approved
- P2267R0: Library Evolution Policies
- P0447R24: Introduction of std::hive to the standard library (D0447R25 to be published)
- P2019R4: Thread attributes
- P2845R4: Formatting of std::filesystem::path
- P2971R1: Implication for C++ (library part tentatively approved)
- P2663R4: Proposal to support interleaved complex values in std::simd
- P2933R0: std::simd overloads for standard C++ <bit> header
- P1928R8: std::simd - Merge data-parallel types from the Parallelism TS 2
- P2642R4: Padded mdspan layouts
- P2664R4: Proposal to extend std::simd with permutation API (requires review of the requested changes)
- P1684R5: mdarray: An Owning Multidimensional Array Analog of mdspan (requires review of the requested changes)
- P2760R0: A Plan for C++26 Ranges (schedule and priorities)

Small fixes and deprecations
- P2862R1: text_encoding::name() should never return null values
- P2300R7: std::execution (small fixes)
- P2809R2: Trivial infinite loops are not Undefined Behavior (naming discussion on the library part)
- P2845R4: Formatting of std::filesystem::path
- P2918R1: Runtime format strings II
- P2944R2: Comparisons for reference_wrapper
- P2863R2: Review Annex D for C++26

- P2869R2: Remove Deprecated `shared_ptr` Atomic Access APIs From C++26
- P2866R1: Remove Deprecated Volatile Features From C++26

Not supported or needs-revision

- P3022R0: A Boring Thread Attributes Interface
- P3001R0: std::hive and containers like it are not a good fit for the standard library
- D3024R0: Interface direction for std::simd
- P3014R0: Customizing std::expected's exception
- P2664R4: Proposal to extend std::simd with permutation API
- P2927R0: Observing exceptions stored in exception_ptr

Requires more review

- P2999R2: Sender Algorithm Customization (require LEWG wording review)
- P3019R2: Vocabulary Types for Composite Class Design (require LEWG wording review)
- P1028R5: SG14 status_code and standard error object
- P2855R0: Member customization points for Senders and Receivers
- P2170R0: Feedback on implementing the proposed std::error type (feedback should be discussed with the authors of P1028)
- P2781R3: std::constexpr_t (support for a language feature)

Info papers

- P2979R0: The Need for Design Policies in WG21
- P3011R0: Supporting document for Hive proposal #1: outreach for evidence of container-style use in industry
- P3012R0: Supporting document for Hive proposal #2: use of std::list in open source codebases
- P2980R0: A motivation, scope, and plan for a physical quantities and units library

Evening Sessions

LEWG was also organizing two evening sessions during Kona:

- Tuesday: SIMD API
- Thursday: P3023R0: C++ Should Be C++

The evening sessions encouraged discussion and were beneficial for increasing consensus and making informed decisions. LEWG will arrange additional evening sessions on similar topics when required (the next evening session is tentatively devoted to “Units and Quantities for the standard library”).

Summary

During Kona 2023, we've managed to go over and make significant progress on 35 large papers, and we will continue to make progress during our weekly telecons until Tokyo.

Core (Maurer)

We have 44 core issues as the first motion. We met jointly with EWG for issues processing to decide on direction. This was successful and we'll look into doing it again.

We have a number of papers on the plate for the polls.

Thank you for those who alerted me about the issue with the fifth agenda item - that needs to go back to drafting and we will see it in Tokyo again. This is already struck on the straw poll page.

Among the 44 issues I would like to point out issue 2823

- turns implicit undefined behavior into explicit undefined behavior
- here: dereferencing a null pointer or a past-the-end pointer is undefined behavior
- "null lvalues" don't exist
- contrast to C: int *p = nullptr; &*p; is ok there
- special carve-out for typeid

CWG will be meeting this afternoon from 1pm-4pm in ballroom 4
Thank you to the scribes.

CWG polls

1. Accept as a Defect Report and apply the proposed resolution of all issues in P3046R0 (Core Language Working Group "ready" Issues for the November, 2023 meeting) to the C++ Working Paper.

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

2. Accept as a Defect Report and apply the changes in P2308R1 (Template parameter initialization) to the C++ Working Paper.

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

3. Apply the changes in P2662R3 (Pack Indexing) to the C++ Working Paper.

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

4. Apply the changes in P2864R2 (Remove Deprecated Arithmetic Conversion on Enumerations From C++26) to the C++ Working Paper.

No Discussion
Objection in the room
Herb reminds the room of voting procedure.

In favour : 66 (54 in person + 12 online)
Opposed : 2 (1 in person + 1 online)
Abstain : 21 (7 in person + 14 online)

Motion passes.

~~**5. Apply the changes in P2748R3 (Disallow Binding a Returned Glvalue to a Temporary) to the C++ Working Paper.**~~

Library (Wakely)

LWG met all week, although I wasn't in Kona so thanks to Jeff and Dietmar for running the room, and the other for helping out when Jeff and Dietmar has to be elsewhere.

LWG will not be meeting after plenary, but we'll resume our weekly telecons soon, probably in a couple of weeks. We're discussing moving the time or day for the LWG telecon so that it's more favourable to some other time zones. Please let us know if you would be more likely to attend LWG telecons if we changed the time.

Some of the polls today are for proposals reviewed during telecons since Varna, but not many because most of the telecons were spent reviewing the Linear Algebra proposal, which is huge. We finally got to the end of it this week and it's in the polls. Thank you to the paper authors and especially to the tireless reviewers for getting to the end of another mammoth paper.

This week we resolved a handful of issues, but mostly reviewed papers. Everything we spent review time on is being polled today except for P2300 (execution) and P1068 (vector API for RNG).
The review of P2300 will probably be the focus for every second telecon for some time, and we are going to try some new ways to review wording updates, so we work with diffs and pull requests against the actual text, instead of inaccurately describing the changes we want in English.

Alisdair Meredith : Do you think your backlog is steady, growing, or are you catching up ?
Jonathan Wakely : It grew a bit. We spent a lot of time on linear algebra. I'm not worried about it.

Herb Sutter : Roughly what percentage will be spent on P2300 ?
Jonathan Wakely : Probably 50% for the next few months.

LWG polls

1. Apply the changes for all Ready and Tentatively Ready issues in P3040R0 (C++ Standard Library Issues to be moved in Kona, Nov. 2023) to the C++ working paper.

Davis Herring : There is a merge conflict between this and poll 3.

Jonathan Wakely : This was done in an awareness of everyone. I will make it clear to the editors.

No objection to unanimous consent.
Motion passes.

2. Apply the changes in P0543R3 (Saturation arithmetic) to the C++ working paper.

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

3. Apply the changes in P2407R5 (Freestanding Library: Partial Classes) to the C++ working paper.

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

4. Apply the changes in P2546R5 (Debugging Support) to the C++ working paper.

No discussion
Objections in the room.

In favour : 62 (46 in person + 16 online)
Opposed : 2 (0 in person + 2 online)
Abstain : 25 (13 in person + 12 online)

Motion passes.

Jonathan Wakely : There has been discussion about this on reflector. Hubert has made some good suggestions for improving it. The wording has to be non normative. We can do this later. The concerns have been heard and we will address them, hopefully before C++26.

5. Accept as a Defect Report and apply the changes in P2905R2 (Runtime format strings) to the C++ working paper.

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

6. Apply the changes in P2918R2 (Runtime format strings II) to the C++ working paper.

Zhihao Yuan : was the option to introduce a new format option as opposed to a new type discussed ?
Jonathan Wakely : This would be a question for LEWG, we just reviewed the paper.
Fabio Fracassi : I don't think anyone brought this up.

Zhihao Yuan : It was brought up in the comments when doing electronic polling.
Inbal Levi : electronic polling has been using since the pandemic. During the electronic polling you can enter comments. These comments are not a source for the discussion in the room, they are meant for feedback for the authors. If something needs to be discussed by the group. please come to us directly.

Herb Sutter : Does the question apply to the previous poll?
Inbal Levi : No
Herb Sutter : is there a dependency between the two polls ?
Inbal Levi : No.

Daisy Hollman : We touched on a similar issue in the discussions this week when we spoke to another paper that does use types, and the room was fine with the direction. We didn't take polls.

Tim Song : from the Varna minutes, it looks like it was discussed there.

Objections in the room.

In favour : 62 (44 in person + 18 online)
Opposed : 1 (1 in person + 0 online)
Abstain : 29 (21 in person + 8 online)

Motion passes.

7. Accept as a Defect Report and apply the changes in P2909R4 (Fix formatting of code units as integers (Dude, where's my char?)) to the C++ working paper.

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

8. Apply the changes in P0952R2 (A new specification for `std::generate_canonical`) to the C++ working paper.

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

9. Apply the changes in P2447R6 (`std::span` over an initializer list) to the C++ working paper.

No discussion.
Objections in the room.

In favour : 65 (50 in person + 15 online)
Opposed : 2 (1 in person + 1 online)
Abstain : 26 (14 in person + 12 online)

Motion passes.

10. Apply the changes in P2821R5 (`span.at()`) to the C++ working paper.

No discussion.
Objections in the room.

In favour : 62 (46 in person + 16 online)
Opposed : 2 (2 in person + 0 online)
Abstain : 28 (14 in person + 14 online)

Motion passes.

11. Apply the changes in P2868R3 (Remove Deprecated `std::allocator` Typedef From C++26) to the C++ working paper.

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

12. Apply the changes in P2870R3 (Remove `basic_string::reserve()` From C++26) to the C++ working paper.

No discussion.
Objections in the room.

In favour : 65 (50 in person + 15 online)
Opposed : 2 (1 in person + 1 online)
Abstain : 23 (13 in person + 10 online)

Motion passes.

13. Apply the changes in P2871R3 (Remove Deprecated Unicode Conversion Facets from C++26) to the C++ working paper.

No discussion.
Objections in the room.

In favour : 57 (46 in person + 11 online)
Opposed : 2 (0 in person + 2 online)
Abstain : 29 (15 in person + 14 online)

Motion passes.

14. Apply the changes in P2819R2 (Add tuple protocol to `complex`) to the C++ working paper.

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

15. Apply the changes in P2937R0 (Freestanding: Remove `strtok`) to the C++ working paper.

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

16. Apply the changes in P2833R2 (Freestanding Library: inout expected span) to the C++ working paper.

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

17. Accept as a Defect Report and apply the changes in P2836R1 (`std::basic_const_iterator` should follow its underlying type's convertibility) to the C++ working paper.

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

18. Apply the changes in P2264R7 (Make `assert()` macro user friendly for C and C++) to the C++ working paper.

No discussion.
Objections in the room.

In favour : 66 (52 in person + 14 online)
Opposed : 1 (1 in person + 0 online)
Abstain : 27 (12 in person + 15 online)

Motion passes.

19. Apply the changes in P1673R13 (A free function linear algebra interface based on the BLAS) to the C++ working paper.

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

Direction Group (Stroustrup)

No report.

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.

2023-12-16: Post-Kona
2024-02-15: Pre-Tokyo

8.3 Plans for the future

No discussion.

8.4 Next and following meetings

- 2024-03-18/23: Tokyo, Japan (N4961)
- 2024-06-24/29: St. Louis, MO, USA (Nxxxx)

Herb Sutter : The next meeting is in Tokyo.
JF Bastien : Read the paper for more information about the meeting.

Herb Sutter : Thank you to Bill Seymour for hosting us in St Louis.
Bill Seymor : The official invitation will be in the next mailing. If you want to print the invitation, send me the email and I'll send you the url for the original html paper.

Herb Sutter: After St Louis we will be meeting in October or November in Poland. We will have the dates shortly. No announcement for 2025, but we have three European hosts volunteering.

9. Adjournment

Meeting adjourned at 9:44 AM GMT-10.

10. Attendance

Attendee	NB
Adams, Michael	SCC
Adelstein Lelbach, Bryce	ANSI

Alday, Juan	ANSI
Amini, Parsa	ANSI
Arutyunyan, Ruslan	ANSI
Ash, Bill	ANSI
Ažman, Gašper	BSI
Baker, Billy	ANSI
Baker, Lewis	ANSI
Balog, Pal	ANSI
Basith, Shanawaz	ANSI
Bastien, Jean-Francois	SCC
Berne, Joshua	ANSI
Bi, Brian	ANSI
Birbacher, Frank	ANSI
Boeckel, Ben	ANSI
Boehm, Hans	ANSI
Brock, Michael	ANSI
Brown, Bret	ANSI
Brown, Jorg	Guest
Brown, Walter E	Guest
Butler, Matthew	ANSI
Büttner, Sebastian	ANSI
Caves, Jonathan	ANSI
Chen, Yuxuan	ANSI
Childers, Wyatt	ANSI
Chorazewicz, Igor	ANSI
Coe, Jonathan Brian	BSI
Craig, Benjamin	ANSI
Craig, Philip	BSI
Cranmer, Joshua	ANSI
D'Angelo, Giuseppe	ANSI
Dave, Jagrut	ANSI
de Wever, Mark	ANSI
Dionne, Louis	SCC
Dos Reis, Gabriel	AFNOR
Douglas, Niall	NSAI

Douglas, Robert	ANSI
Doumler, Timur	BSI
Dusikova, Hana	UNMZ
Engert, Daniela	ANSI
Fertig, Andreas	DIN
Fevold, Jake	ANSI
Floyd, Paul	Guest
Foco, Marco	UNI
Fracassi, Fabio	DIN
García Sánchez, José Daniel	UNE
Garland, Jeff	ANSI
Giroux, Olivier	ANSI
Goldblatt, David	ANSI
Goodspeed, Nathaniel	ANSI
Gordon, Fraser	SCC
Gozillon, Andrew	ANSI
Gruber, Bernhard Manfred	SNV
Hagins, Jody	ANSI
Halpern, Pablo	ANSI
Hauswedell, Hannes	IST
Hava, Michael Florian	ASI
Herrick, Ellen	ANSI
Herrick, Michael	ANSI
Herring, Davis	ANSI
Hoemmen, Mark	ANSI
Hollman, Daisy	ANSI
Honermann, Tom	ANSI
Jabot, Corentin	AFNOR
Waterloo, Jarrad	ANSI
Jevnik, Joseph	ANSI
Jha, Dheeraj	BIS
Jort, Ari	ANSI
Kaan, Erdogmus	Guest
Kalb, Jon	ANSI
Katz, Dan	ANSI

Keane, Erich	ANSI
Khlebnikov, Rostislav	ANSI
Klauser, Nikolas	ANSI
Koeppe, Thomas	ANSI
Kosunen, Elias	SFS
Kretz, Matthias	DIN
Krzemienski, Andrzej	PKN
Kuhl, Dietmar	ANSI
Kulczycki, Peter	ASI
Laine, Timothy	ANSI
Lakos, John	ANSI
Leahy, Robert	ANSI
Lebrun-Grandie, Damien	ANSI
Levi, Inbal	SII
Li, Yihe	Guest
Liber, Nevin	ANSI
Lippincott, Lisa	Guest
Loser, Joseph	Guest
MacLean, Colin	ANSI
Maness, Wesley	ANSI
Marr, Greg	Guest
Marsman, A.	NEN
Maurer, Jens	ANSI
McKenney, Paul	ANSI
Meerwald, Christof	ASI
Meneide, JeanHeyd	NEN
Meredith, Alisdair	ANSI
Merrill, Jason	ANSI
Miller, William	ANSI
Morales, Nicolas	ANSI
Moschovakos, Paris	SNV
Müller, Jonathan	DIN
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
Persson, Jonas	SIS
Preney, Paul	SCC
Prince, Tim	ANSI
Pusz, Mateusz	PKN
Ranns, Nina Dinka	BSI
Ratzloff, Phil	ANSI
Revzin, Barry	ANSI
Rigault, Jean-Paul	AFNOR
Rivera Morell, René Ferdinand	ANSI
Rodgers, Thomas	ANSI
Ronkainen, Jari	SFS
Roy, Patrice	SCC
Ruoso, Daniel	ANSI
Ryan, Christopher	ANSI
Sajjad, Hassan	Guest
Sandoe, Iain	BSI
Sankel, David	ANSI
Satle, Ankur	BIS
Schurr, Stephen	ANSI
Seacord, Robert	ANSI
Seymour, William	ANSI
Shoop, Kirk	ANSI
Snyder, Jeff	BSI
Sommerlad, Peter	SNV
Song, Tim	ANSI
Spencer, Michael	ANSI
Spicer, John	ANSI
St. Amour, Bryan	SCC
Stroustrup, Bjarne	ANSI

Sutter, Herb	ANSI
Talbot, Alan	Guest
Taylor, Matthew	Guest
Teeple, Doug	ANSI
Tenty, David	SCC
Tong, Hubert	SCC
Touton, James	ANSI
Towner, Daniel	ANSI
Trott, Christian	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
Wei, Weile	ANSI
Weis, Andreas	DIN
Williamson, Gerald	ANSI
Wong, Michael	SCC
Xie, Hui	BSI
xu, chuanqi	SAC
Yaghmour, Shafik	ANSI
Yuan, Zhihao	ANSI
Zeren, Mark	ANSI
Zissu, Andrei	SII
Zverovich, Victor	ANSI