WG21 February 2023 Hybrid meeting
Record of Discussion
ISO/IEC JTC1 SC22 WG21 P2824— 2023-03-06
Nina Dinka Ranns, dinka.ranns_at_gmail.com

Chair: John Spicer

6-11 February 2023, Issaquah, WA USA

1. Opening activities

John Spicer opens the meeting at 09:02 AM GMT-8

1.1 Opening comments, welcome from host (INCITS C++)

John Spicer presents. Thank you to the sponsors : Standard C++ Foundation, WorldQuant, Edison Design Group.

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

1.2 Meeting guidelines

John Spicer presents.

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.

Agenda is on the wiki. There is a lot of other important information on the wiki, please make sure you have access to it. Talk to your neighbor if you need access to the wiki.

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

The INCITS Antitrust Guidelines (INCITS C++)
The INCITS Patent Policy (INCITS C++)
The ISO Code of Conduct
The INCITS Code of Conduct (INCITS C++)
The IEC Code of Conduct
The WG21 Practices and Procedures, and Code of Conduct

John Spicer presents the slides. They are also linked in the agenda.

If you have any questions or concerns about CoC issues, please approach a committee officer or a NB representative and bring it to their attention.

1.3 Membership, voting rights, and procedures for the meeting

John Spicer presents voting rights.

Rules are the same for US and other NBs. Every person registered in the global directory gets one vote.

Everyone can vote in subgroups. Please see best practice for voting.

If you are representing an organization that is considering formally joining INCITS/C++, or your organization is already a member and you wish to change your voting status, please inform an officer.

Everyone should register in the attendance doodle poll.

Nina Ranns : Subgroup chairs, please remind people to sign the attendance doodle poll.

John Spicer presents voting procedures during the plenary.

Herb Sutter : We have 19 national bodies present at this meeting.

1.4 INCITS/C++ Social Media Information (Linkedin, Twitter, Facebook)

Click here for the INCITS social media landing page.

1.5 Introductions

New members introduce themselves.
John Spicer welcomes new members.

1.6 Agenda review and approval (INCITS/C++ motion, WG21 poll)

John Spicer presents the agenda.

The meeting goals described above are derived from the schedule adopted in 2020 and described in: P1000R4.

The meeting goal is to finalize NB comments and proceed with the publication of the C++23 standard.

Agenda amendment - dates in the agenda are all off by one.
INCITS/C++ business.

WG21 motion to approve the meeting agenda.
The motion is unanimously approved by WG21.

1.7 Editor's reports, approval of working drafts

Document	Editor's report	Prospective WD
C++ Standard	N4929	N4928

WG21 motion to approve the above.
The motion is unanimously approved by WG21.

1.8 Approval of the minutes of the previous meetings (INCITS C++ motion, WG21 poll)

Meeting	Minutes
WG21 Kona	N4940
INCITS C++ Kona	pl22.16-2023-000001-0002
WG21 pre-Issaquah administrative telecon	N4942

INCITS/C++ business.

The amendments to the Kona minutes from the previous version are localized to the attendance list.

WG21 motion to approve the above.
The motion is unanimously approved by WG21.

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 discussion

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

(Clarify rooms available for evening sessions)
Jens Maurer presents.
Thank you to those who brought audio setup gear and all the volunteers who helped set it up. This helps the virtual attendance.

Jens Maurer presents the time schedule

Jens presents the subgroup room assignment.

Jens presents the evening sessions. The agenda is also printed out in the lobby.

Herb Sutter : evening sessions are normal subgroup sessions and can make decisions and progress.
Bryce Adelstein Lelbach : LEWG will have evening sessions, but we won't be making decisions at that session. It is only there for early feedback.

Jens presents food options and the hotel food policy.

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.
Chairs are encouraged to update the straw polls page as soon as they have papers ready.
Attendees are encouraged to keep track of the straw polls page and raise any concerns as early as possible.

7. Review of the meeting

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

SG1: Concurrency (Giroux)

Nothing to report

SG2: Modules (Stone)

Did not meet.

SG4: Networking (Snyder)

SG4 discussed two papers this week .

P2586R0 proposed a new networking API, with an implementation, but without any support for asynchrony. SG4 liked the design on the condition that an asynchronous API could be built around it. However, the author does not plan on pursuing the paper any further.

P2762R0 proposed a way of adapting the Networking TS to the senders/receivers API style that is currently favored by LEWG. SG4 unanimously encouraged further work on this approach.

SG5: Transactional memory (Boehm/Maurer)

Did not meet.

SG6: Numerics (Kretz/Lippincott/McFarlane)

SG6 met for two evening sessions and processed 5 papers. Most notably we forwarded P2075R2 "Philox as an extension of the C++ RNG engines" (it came back because authors preferred not to follow previous SG6 feedback), and P1383R1 "More constexpr for <cmath> and <complex>", informing LEWG about the numerics perspective.

We discussed P2681R1 "More Stats Functions" and P2746R0 "Deprecate and Replace Fenv Rounding Modes" and provided feedback for new revisions. If you have examples of users that use rounding modes, please contact Hans Boehm.

Finally, we looked at P2159R1 "An Unbounded Decimal Floating-Point Type", a late paper, and decided that its use cases describe a feature SG6 is not interested in.

SG7: Compile-time programming (Dusikova/Vandevoorde)

Did not meet.

SG9: Ranges (Levi/Carter)

SG9 met on Monday for two sessions, as well as had a joint LWG-SG9 session:

We have discussed the NB comment and the paper addressing it and reached two separate fixes which are planned for ranges for 26.
The solution in the paper was not accepted, but one of the fixes planned is aiming for fixing the way in which views interact with ranges algorithms, and is a major update for our model.
Thank you to the author for bringing this up, and for the great group of ranges' attendees who worked on this solution (including Tim Song, Barry Revzin, Eric Niebler, Louis Dionne, Yehezkel Bernat)

- US#523: US 46-107 25.5.7.1 (counted.iterator) Too many iterator increments (forwarded to LEWG with a weak consensus to have the solution in P2406 for C++26)
- P2406R3: Add `lazy_counted_iterator` Yehezkel Bernat, Yehuda Bernat (forwarded with weak consensus, an alternative design was encouraged by LEWG)
- P2655R2: common_reference_t of reference_wrapper Should Be a Reference Type Hui Xie, S. Levent Yilmaz (forwarded for C++23)
- P2769R0: get_element customization point object Ruslan Arutyunyan, Alexey Kukanov (discussed, waiting for a new revision)
- P2022R0: Rangified version of lexicographical_compare_three_way Ran Regev (discussed, waiting for a new revision)

SG9 will continue to have monthly telecons until Varna, which will focus on having new ranges' features for C++26 and finalizing the papers presented at Issaquah.

SG10: Feature test (Revzin/Wakely)

Did not meet.

SG12: Undefined and unspecified behavior (Ballman/Wakely)

Did not meet.

SG14: Games & low latency (Wong)

Did not meet.

SG19: Machine Learning (Wong/Reverdy)

Did not meet.

SG15: Tooling (Spencer/Boeckel)

The Tooling Study Group met for two evening sessions where we discussed how to build modules and what we would like to ship in the first C++ Ecosystem IS. We had between 11 and 18 people in the room.

For modules we discussed how to handle the case of tools that only observe the build via compiler command lines, such as a tool that observes all `cl.exe` process launches. These tools can't tell which options are required to be the same between modules and their importers, and which can be different. We decided that such tools will need additional metadata for these cases.

For the C++ Ecosystem IS we have decided we would like the first revision to include:

- Build System <==> PM Interop
- Minimum set of recognized file extensions
- [Introspection](http://wg21.link/p2717)
- Portable diagnostics format via [SARIF](https://sarifweb.azurewebsites.net/)
- Command line portability

SG18: LEWG Incubator (Baker/Liber)

Did not meet.

SG16: Unicode (Honermann/Brett)

Did not meet.

SG17: EWG Incubator (Ballo/Keane)

We met for the first time in several years.
We saw 9 papers and had an interesting conversation on object relocation.
We forwarded 3 papers to EWG and we gave feedback on the other 6.

SG20: Education (van Winkel)

Did not meet.

SG21: Contracts (Spicer/Doumler)

SG21 met for two afternoons (Tue + Wed); we had ~30 people attending.

As per our roadmap paper (P2695R1), our focus for this meeting was to agree on a design for how to handle side effects in contract annotations in the Contracts MVP targeting C++26. We have achieved this goal at this meeting. We discussed four papers in this space:

- P2461R1 "Closure-Based Syntax for Contracts" by Gašper Azman et al. (we only discussed only the parts of this paper that deal with side effect semantics, not the parts that deal with syntax)
- P2570R2 "Contract predicates that are not predicates" by Andrzej Krzemieński
- P2751R0 "Evaluation of Checked Contract-Checking Annotations" by Joshua Berne
- P2756R0 "Proposal of Simple Contracts Side Effect Semantics" by Andrew Tomazos

We had a very productive discussion, at the end of which we reached consensus on the following points:
1) A contract-checking predicate is evaluated zero or more times in eval_and_abort mode,
2) Certain useful rules on reordering/elision of contract-checking predicates in a sequence of contract annotations, and
3) When the evaluation of a contract-checking predicate exits the control flow other than by returning true/false or throwing (abort, longjmp), the behavior is as if it were not in a contract-checking predicate (you just get the abort/longjmp).

We failed to reach consensus on the following points: 1) What should we say about the case when the evaluation of a contract-checking predicate has undefined behavior? 2) What should happen when the evaluation of a contract-checking predicate throws?

We decided to postpone discussion on 1 for now, until new papers in this space come along; we further decided to re-discuss 2 in the context of contract violation handling, which will be our focus in Varna. To facilitate progress on this matter, we decided to switch the target dates for agreeing on a design for contract violation handling and for syntax in our roadmap, such that the contract violation handling comes first.

We also decided to conduct regular monthly telecons between now and Varna.

SG22: C/C++ Liason (Ballman)

Did not meet.

SG23: Safety/Security (Orr/Craig)

SG23 met for an afternoon and looked at:
1. P2723R1 - Zero-initialize objects of automatic storage duration
2. D2795R0 - Correct and incorrect code, and what it means
3. P2687R0 - Design Alternatives for Type-and-Resource Safe C++
4. P2759R1 - DG Opinion on Safety for ISO C++ (note: R1)

The first two papers were forwarded to EWG and the third was forwarded subject to a revision. We hope to hold monthly telecons between now and Varna.

ABI Group (Vandevoorde)

Did not meet.

Admin (Liber)

If you haven't marked the attendance sheet, please send an email to me (Nevin Liber) or Nina Ranns to get credit for being at the meeting.

We're looking to have the paperbot on Mattermost go through the isocpp.org papers so we can get things up to date.

Evolution (Bastien)

The summary is also on the top of the EWG page on wiki.
C++23 NB comments: one rejected, one forwarded to CWG with a proposed resolution.

- FR-025-017 — P2691R0 #1356 Allow referencing inline functions with internal linkage from outside their defining header unit, and D2808r0 #??? Internal linkage in the global module — Rejected. No consensus for a change.
- US 8-036 6.6p3.2 — D2788r0 #??? Modular constants, Forward D2788R0 to CWG as a resolution for US8-036 applied to C++23 and as a DR to C++20.

More C++23:

- P2593R1 #1251 Allowing static_assert(false) — Forward P2593R1 to CWG for C++23 and as a DR.

CWG issues: resolve 5, 1 to be seen again with proposed resolution, 2 need papers.

- CWG2518 Conformance requirements and #error/#warning — EWG agrees with the proposed resolution to CWG2518; additionally, this should be applied equally to failed static_assert
- CWG2538 Can standard attributes be syntactically ignored? P2552R1 #1212 On the ignorability of standard attributes, Timur Doumler — Clarify alignment with C23 with an editorial note, handled by CWG2695
- CWG2692 Static and explicit object member functions with the same parameter-type-lists — D2797R0 #??? provided feedback to the authors on the proposed resolution, send to CWG
- CWG2521 User-defined literals and reserved identifiers — Suggest that LEWG amend the proposal for CWG2521 to reserve UDL declarations with double underscores anywhere for the implementation, then forward to CWG
- CWG2678 std::source_location::current is unimplementable — EWG approves of the approach for CWG2678 of changing the ODR to make use of source_location in a way that causes an inline function/function template/etc to 'be different' be an ODR violation (as proposed by Jason) as a DR.

- CWG2563 Initialization of coroutine result object — EWG agrees with Gor and Lewis' proposed solution to CWG2563, and instructs them to enpanel a CWG expert to assist in turning this into wording, and bring it back to EWG
- CWG2463 Conditions for trivially copyable classes — Needs paper, Ville volunteering
- CWG2536 Zero-initialization in addition to constant initialization and stability of padding bits — Needs a paper

Matching C23: EWG matches C23 in 2 instances.

- P1967R10 #700 #embed - a simple, scannable preprocessor-based resource acquisition method — EWG agrees to match C23, sent to CWG
- P2537R2 #1200 Relax va_start Requirements to Match C — Forward P2537R2 "Alternative 2" (change define, remove all but 1st sentence) to LWG/CWG for inclusion in C++26.

New C++26 features: encourage further work on 16 topics, and has no consensus to continue work on 2.

- P2169R3 #878 A Nice Placeholder With No Name — EWG saw issues with the implementation, would want them fixed, and would then want experience on varied C++ codebases to ensure that no breakage occurs
- P2662R0 #1329 Pack Indexing — EWG encourages further work.
- P2640R2 #1306 Modules: Inner-scope Namespace Entities: Exported or Not? — Gave feedback
- P2600R0 #1264 A minimal ADL restriction to avoid ill-formed template instantiation — EWG would like to see more work in the direction of P2600R0, as we are motivated by the problem it is trying to solve.
- P2477R3 #1133 Allow programmer to control and detect coroutine elision by static constexpr bool should_elide(), —EWG is interested in a solution to the problem raised by P2477R3 of coroutine elision, in some form. EWG prefers the approach suggested by Gor, where the programmer has an option to prohibit elision, or to have it be an error when the optimizer fails to elide a coroutine.
- P2670R1 #1336 Non-transient constexpr allocation — EWG encourages more work in the direction of 'propconst' as a specifier akin to 'mutable', and would like to see this again.
- constexpr void* — P2738R0 #1431 and P2747R0 #1438 — Forward D2738R1 to CWG for inclusion in C++26. Authors of P2738 and P2747 should work together and come back with reviewed wording/implementation experience.
- static_assert message generation — P2741R0 #1434 and P2758R0 #1445 and N4433 — EWG would like to encourage more work on P2741R0 (static_assert-with-constexpr-message), solicit feedback from SG16, and see it again with core expert reviewed wording.
- Pointer zap — P2188R1 #895 and P1726R5 #553 and P2414r1 #1084 and P2434r0 #1364 and P2318r1 #1011 — Guidance was given to the authors, and we expect to see an omnibus solution to the problem.
- P2677R2 #1343 Reconsidering concepts in-place syntax— Knowing that our time is limited, EWG would like to see more work in the direction of P2677: consensus.
- P2641R2 #1307 Checking if a union alternative is active — Return P2641R2 to LEWG with our blessing and recommendation that this be included in C++26.
- P2481R1 #1137 Forwarding reference to specific type/template — More work, come back.

- Initializing stack variables
    - D2795R0 #1460 Correct and incorrect code, and what it means — EWG is interested in adding the concept of erroneous behavior as an alternative to undefined behavior, as proposed in D2795R0. Feedback was given to the author on how to increase consensus.
    - P2754R0 #1461 Deconstructing the Avoidance of Uninitialized Reads of Auto Variables — Discussed, and used to prepare for the next paper.
    - P2723R1 #1401 Zero-initialize objects of automatic storage duration — EWG believes zero initialization, as presented in P2723R1, is the approach we wish to take to automatic stack variable initialization (with opt out).
- P2410R0 #1080 Type-and-resource safety in modern C++ — We encourage further work on “profiles” as presented in P2410R0.
- P2748R0 #1439 Disallow Binding a Returned glvalue to a Temporary — EWG encourages more work on P2748R0, and would like to see it again with improved wording to improve library/unevaluated context concerns.
- D2806R0 #1462 do statement-expressions — EWG encourages more work in the direction of do-statement-expressions as presented in D2806R0.
- P2370R2 #1056 Stacktrace from exception — No Consensus.
- P2672R0 #1338 Exploring the Design Space for a Pipeline Operator — No Consensus.

Library Evolution (Adelstein Lelbach/Fracassi/Craig)

At 2023-02 Issaquah

C++23

- Senders & Networking (P2300, P2762)
- Linear Algebra (P1673 forwarded, P1385)
- SIMD (P1928, P2638, P2663)
- Unicode (P2728, P2729)
- mdspan (P2630, P1684)
- fiber_context (P0876)
- Tuples (P2141, P2580, P2726)

Future Plans

- Focus on big papers at F2F meetings
- Policy (P1656, P2148)
- SIMD (Forward P1928)
- Unicode (P2728, P2729)
- mdspan (Forward P1684)
- fiber_context (Forward P0876)

At the next meeting we're planning to have a policy discussion, including the P1656R0 ‘"Throws: Nothing" should be noexcept’ paper.

Core (Maurer)

Core reviewed the remaining NB issues that came from EWG and all C++26 papers that were in our queue. Some C++26 papers seem ready and some need another round. We will see all of them again to make sure they are still accurate once we have the final C++23 document.
We reviewed a large amount of core issues. There are 34 core issues in ready status that will be moved today, most of them are in the first motion. Second motion is talking about core issues that are not DRs.

Motion talking about issue 2518 covers both #error and #warning as well as static_assert(false) changes.

Issue 2521 contains user defined literals and includes the deprecation of literal operator ids that contain space between the string literal and the ud-suffix.

Issue 2678 is about std::source_location::current. It fixes it in a more general manner by getting an ODR violation if you do anything in an ODR sensitive context that gets you different values at compile time. We expect we will hit that problem more when we get reflection so this is a forward compatible fix.

Issue 2659 is a repair to get the feature test macro for the range-for loop change.

Straw poll 7 - we erroneously voted P2647R1 as a defect report in Kona. We are not undoing this paper, it only undoes the DR status of this paper as it refers to feature that went in after C++20. This is a no-op poll in terms of wording in the standard

We have a bunch of papers for NB comments.

Issue 2692 is fixing holes in explicit object member functions. The changes are comparatively large in terms of wording touched.

Core will meet this afternoon from 1:30-4pm in Denton room.

CWG polls

1. Accept as a Defect Report and apply the proposed resolution of all issues except issues 2518, 2521, 2659, 2674, 2678, and 2691 in P2796R0 (Core Language Working Group "ready" Issues for the February, 2023 meeting) to the C++ Working Paper.

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

2. Apply the proposed resolution of issues 2674 and 2691 in P2796R0 (Core Language Working Group "ready" Issues for the February, 2023 meeting) to the C++ Working Paper.

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

3. Accept as a Defect Report and apply the proposed resolution of issue 2518 (Conformance requirements and #error/#warning) in P2796R0 (Core Language Working Group "ready" Issues for the February, 2023 meeting) to the C++ Working Paper.

Ville Voutilainen : I find it unfortunate that we keep backporting breaking changes.
John Spicer : EWG, was this considered ?
Erich Keane : yes.

Objections in the room.
John Spicer reminds the room of the voting rules

In favor : 82 (50 in person + 32 online)
Opposed : 3 (2 in person + 1 online)
Abstain : 12 (6 in person + 6 online)

Motion passes.

4. Accept as a Defect Report and apply the proposed resolution of issue 2521 (User-defined literals and reserved identifiers) in P2796R0 (Core Language Working Group "ready" Issues for the February, 2023 meeting) to the C++ Working Paper.

No discussion.

Objections in the room.

In favor : 76 (45 in person + 31 online)
Opposed : 1 (1 in person + 0 online)
Abstain : 15 (13 in person + 2 online)

Motion passes

5. Accept as a Defect Report and apply the proposed resolution of issue 2678 (std::source_location::current is unimplementable) in P2796R0 (Core Language Working Group "ready" Issues for the February, 2023 meeting) to the C++ Working Paper.

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

6. Apply the proposed resolution of issue 2659 (Missing feature-test macro for lifetime extension in range-for loop) in P2796R0 (Core Language Working Group "ready" Issues for the February, 2023 meeting) to the C++ Working Paper, resolving NB comment DE-038.

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

7. Specify that P2647R1 (Permitting static constexpr variables in constexpr functions) (applied in November, 2022) is no longer a Defect Report.

Daveed Vandevoorde : Did core consider making the relevant feature a DR ?
Jens Maurer : We did consider it when we saw the original feature. I didn't feel people were in favor of DRing this change.
Herb Sutter : That would be a change that EWG would need to approve.
Ville Voutilainen : I was the author of the feature. I didn't intend the feature to be backported as a DR

No objection to unanimous consent.
Motion passes.

8. Apply the changes in P2736R2 (Referencing The Unicode Standard) to the C++ Working Paper, resolving NB comments FR-010-133 and FR-021-013.

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

9. Accept as a Defect Report and apply the changes in P2788R0 (Linkage for modular constants) to the C++ Working Paper, resolving NB comment US 8-036.

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

10. Apply the changes in P2797R0 (Proposed resolution for CWG2692 Static and explicit object member functions with the same parameter-type-lists) to the C++ Working Paper.

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

Library (Wakely)

LWG met all week, we had no evening session.
We processed all C++23 work and had a couple of sessions on C++26 material.
We had two sessions on P2300. We’re making good progress.
Apart from one paper, everything that was forwarded to us got reviewed, approved and is in the straw polls.
Most polls are addressing NB comments. The other ones are the usual DR issue fixes.

Concurrency TS poll is to put one more thing in the TS. That will conclude what was in the plan that LEWG had for TS concurrency content. After the library polls, there will be a poll to publish Concurrency TS v2.

LWG polls

Concurrency TS v2 polls

1. Apply the changes in P0290R4 (synchronized value) to the Concurrency TS v2 working paper.

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

C++23 polls

2. Apply the changes for all Ready and Tentatively Ready issues in P2789R0 (C++ Standard Library Issues to be moved in Issaquah, Feb. 2023) to the C++ working paper.

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

3. Apply the changes for all Immediate issues except 3441 in P2790R0 (C++ Standard Library Immediate Issues to be moved in Issaquah, Feb. 2023) to the C++ working paper.

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

4. Apply the changes for the Immediate issue 3441 in P2790R0 (C++ Standard Library Immediate Issues to be moved in Issaquah, Feb. 2023) to the C++ working paper.

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

5. Apply the changes in P2770R0 (Stashing stashing iterators for proper flattening) to the C++ working paper. This addresses ballot comment US 61-126 .

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

6. Apply the changes in P2164R9 (views::enumerate) to the C++ working paper. This addresses ballot comments FR 14-021 and US 48-108 .

No discussion.
Objections in the room.

In favor : 64 (42 in person + 22 online)
Opposed : 2 (0 in person + 2 online)
Abstain : 27 (14 in person + 13 online)

Motion passes.

7. Apply the changes in P2711R1 (Making multi-param constructors of views explicit) to the C++ working paper.

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

8. Apply the changes in P2609R3 (Relaxing Ranges Just A Smidge) to the C++ working paper. This addresses ballot comment US 39-099 .

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

9. Apply the changes in P2713R1 (Escaping improvements in std::format) to the C++ working paper. This addresses ballot comments US38-098 and FR005-134 .

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

10. Apply the changes in P2675R1 (format's width estimation is too approximate and not forward compatible) to the C++ working paper. This addresses ballot comment FR-007-012 .

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

11. Apply the changes in P2572R1 (std::format fill character allowances) to the C++ working paper.

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

12. Apply the changes in P2693R1 (Formatting thread::id and stacktrace) to the C++ working paper. This addresses ballot comment FR-012-023 .

No discussion.
Objections in the room.

In favor : 73 (47 in person + 26 online)
Opposed : 1 (0 in person + 1 online)
Abstain : 20 (9 in person + 11 online)

Motion passes.

13. Apply the changes in P2679R2 (Fixing std::start_lifetime_as for arrays) to the C++ working paper. This addresses ballot comment CA-086.

No discussion.
Objections in the room.

In favor : 64 (41 in person + 23 online)
Opposed : 0 (0 in person + 0 online)
Abstain : 29 (15 in person + 14 online)

Motion passes.

14. Apply the changes in P2674R1 (A trait for implicit lifetime types) to the C++ working paper. This addresses ballot comment GB-089.

No discussion.
Objections in the room.

In favor : 58 (41 in person + 17 online)
Opposed : 1 (0 in person + 1 online)
Abstain : 29 (12 in person + 17 online)

Motion passes.

15. Apply the changes in P2655R3 (common_reference_t of reference_wrapper Should Be a Reference Type) to the C++ working paper.

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

16. Apply the changes in P2652R2 (Disallow User Specialization of allocator_traits) to the C++ working paper. This addresses ballot comment US 33-077.

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

17. Apply the changes in P2787R1 (pmr::generator - Promise Types are not Values) to the C++ working paper. This addresses ballot comment US 53-116.

No discussion.
Objections in the room.

In favor : 48 (32 in person + 16 online)
Opposed : 3 (0 in person + 3 online)
Abstain : 34 (19 in person + 15 online)

Motion passes.

18. Apply the changes in P2614R2 (Deprecate numeric_limits::has_denorm) to the C++ working paper. This addresses ballot comment DE-079.

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

19. Apply the changes in P2588R3 (barrier's phase completion guarantees) to the C++ working paper. This addresses ballot comment DE-135 and US 63-131.

No discussion.
Objections in the room.

In favor : 60 (43 in person + 17 online)
Opposed : 1 (0 in person + 1 online)
Abstain : 28 (11 in person + 17 online)

Motion passes.

20. Apply the changes in P2763R1 (layout_stride static extents default constructor fix) to the C++ working paper.

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

WG21 polls

Poll 1
Appoint a review committee composed of Jonathan Wakely, Daniel Krügler, and Bryan St. Amour to approve the correctness of the Working Paper for C++ Extensions for Concurrency, version 2 as modified by the polls approved at this meeting, and direct the Convener to transmit the approved updated Working Paper for PDTS ballot.

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

Poll 2
Appoint a review committee composed of Daniel Krügler, Nina Ranns, Michael Hava, and Jonathan Wakely to approve the correctness of the C++ working paper as modified by the polls approved at this meeting, and direct the Convener to transmit the approved updated working paper for DIS ballot.

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

Direction Group (Stroustrup)

Nothing to report

8. Closing activities

Herb Sutter : I have asked ISO for formal clarification on whether transcript and closed captioning that the ISO zoom account allows you to turn on conflict with the ban on recording the meetings. They will consider it and let us know. The interim answer from our parent committee manager and our technical program manager is to consider it as a recording until we have a final answer. Don't save the transcript, but you can use the captions. We understand the captions are very useful.

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.

- 2023-02-15: Post-Issaquah
- 2023-05-15: Pre-Varna

9. INCITS/C++ Agenda Items

9.1 INCITS C++ Motions, if any

9.1.1 Do you approve the following? CIB Resolution 2023-04: Use of Paragraph Numbers for DIS 14882 SC 22 approves the use of paragraph numbers for the DIS 14882 ballot.

INCITS/C++ business.

9.1.2 Do you approve the following? CIB Resolution 2023-03: Use of Paragraph Numbers for DTS 19568 SC 22 approves the use of paragraph numbers for the DTS 19568 ballots.

INCITS/C++ business.

9.1.3 Plans for the future (INCITS C++)

No discussion.

9.1.4 Next and following meetings

2023-06-12/17: Varna, Bulgaria (N4935)

Vasil Vassilev presents. See N4935 for more details.

Pre-WG21 Events. Sofia
Chaos and VMware are planning an event on Friday June 9 in Sofia preceding the WG21 meeting in Varna.

We would be happy to welcome WG21 members in Sofia for an onsite meeting with representatives of the C++ community, professionals and students in programming from Bulgaria.
In case you are traveling to Varna via Sofia and you are interested in participating, please contact maya.traykova@chaos.com for more details.

Pre-WG21 Events. Varna
We plan a presentation by a C++ committee member at TU Varna. Details still being sorted out.

Organizing Your Trip
Consider pre-booking a taxi for the night flights as sometimes it takes more than half an hour for a taxi car to come to Golden Sands
Check Ins(outs) work 24 hours. If you would like breakfast to go, contact reception the day before.
The venue offers shuttle service from/to the airport: has to be organized

couple days in advance (don’t forget to mention your flight number)

2023-11-6/11: Kona, HI, USA (N4936)

Herb Sutter : We don’t have firm plans for 2024, but we hope to have some papers in the near future.

(tentative) 2024-winter/spring: Japan
(tentative) 2024-summer: Stockholm, Sweden
(tentative) 2024-autumn: Wrocław, Poland

Gabriel Dos Reis : Does that mean we have no US based meetings in 2024 ?
Herb Sutter : We will have 3 out of 4 meetings in the US before 2024. It balances out.

10. Adjournment (INCITS C++ motion)

Roger Orr : Move to thank the scribes.
Motion unanimously approved.

John Spicer : Thank you to everyone who helped with the equipment for virtual meetings. It’s working well. Big thank you to Jens Maurer who does a lot behind the scenes to put everything together.

Thank you to the sponsors of the meeting

J. Daniel Garcia : We have managed to approve a standard every 3 years since 2011. This is a major success. Thank you everybody.

INCITS/C++ motion to adjourn.

WG21 Motion to adjourn.
No objection.

Meeting adjourned at 10:00 AM GMT-8.

11. Attendance

Attendee	Affiliation
Aaron Ballman	ANSI (United States)
ADAM David Alan Martin	
Alex Voicu	ANSI (United States)
Alexander Riegler	ANSI (United States)
Andreas Weis	DIN (Germany)
Andrei Zissu	SII (Israel)
Andrew Pardoe	
Anthony Williams	BSI (United Kingdom)
Arash Partow	
Arthur Helwig	
Arthur O'Dwyer	ANSI (United States)
Aschwin Marsman	NEN (Netherlands)
Balog Pal	ANSI (United States)
Barry Hedquist	ANSI (United States)
Barry Revzin	ANSI (United States)
Ben Boeckel	
Ben Craig	ANSI (United States)
Bengt Gustafsson	SIS (Sweden)
Benjamin Brock	ANSI (United States)
Bernhard Manfred Gruber	SNV (Switzerland)
Bill Seymour	ANSI (United States)
Billy Baker	ANSI (United States)
Bjarne Stroustrup	ANSI (United States)
Bob Steagall	ANSI (United States)
Bogusław Cyganek	ANSI (United States)
Brian Bi	ANSI (United States)
Bryan St. Amour	SCC (Canada)
Bryce Adelstein Lelbach	ANSI (United States)
Chris Ryan	ANSI (United States)
Christian Trott	ANSI (United States)

Christof Meerwald	ASI (Austria)
Chuanqi Xu	SAC (China)
Colin MacLean	ANSI (United States)
Corentin Jabot	AFNOR (France)
Damien Lebrun-Grandie	ANSI (United States)
Dan Raviv	ANSI (United States)
Daniela Engert	ANSI (United States)
Darius Neațu	ANSI (United States)
Daveed Vandevoorde	ANSI (United States)
David Friberg	SIS (Sweden)
David Goldblatt	ANSI (United States)
David Olsen	ANSI (United States)
David Sankel	ANSI (United States)
David Vitek	ANSI (United States)
Davis Herring	ANSI (United States)
Detlef Vollmann	SNV (Switzerland)
Dietmar Kühl	ANSI (United States)
Eddie Nolan	ANSI (United States)
Elias Kosunen	SFS (Finland)
Eric Niebler	ANSI (United States)
Erich Keane	ANSI (United States)
Fabio Fracassi	DIN (Germany)
Frank Birbacher	ANSI (United States)
Fraser Gordon	SCC (Canada)
Gabriel Dos Reis	AFNOR (France)
Gašper Ažman	BSI (United Kingdom)
Gonzalo Brito	ANSI (United States)
Gor Nishanov	ANSI (United States)
Guy Davidson	BSI (United Kingdom)
Hana Dusíková	UNMZ (Czech Republic)
Hans Boehm	ANSI (United States)
Herb Sutter	ISO/IEC JTC 1/SC 22/WG 21
Howard Hinnant	ANSI (United States)
Hubert Tong	SCC (Canada)
Iain Sandoe	BSI (United Kingdom)

Inbal Levi	SII (Israel)
Ivan Čukić	ANSI (United States)
Jason Merrill	ANSI (United States)
Jean-Paul Rigault	AFNOR (France)
Jeff Garland	ANSI (United States)
Jeff Snyder	BSI (United Kingdom)
Jens Maurer	ANSI (United States)
Jerry Williamson	
JF Bastien	SCC (Canada)
John Lakos	ANSI (United States)
John Spicer	ANSI (United States)
Jonas Persson	SIS (Sweden)
Jonathan Caves	ANSI (United States)
Jonathan Wakely	BSI (United Kingdom)
Jorg Brown	ANSI (United States)
Jose Daniel Garcia Sanchez	UNE (Spain)
Joshua Berne	ANSI (United States)
Joshua Cranmer	ANSI (United States)
Juan Alday	ANSI (United States)
Kaan Erdogmus	ANSI (United States)
Kelly Walker	ANSI (United States)
Kirk Shoop	ANSI (United States)
Konstantin Varlamov	ANSI (United States)
Kulczycki Peter	ASI (Austria)
Lawrence Crowl	
Lewis Baker	ANSI (United States)
Loïc Joly	AFNOR (France)
Louis Dionne	SCC (Canada)
Lucian Teodorescu	ASRO (Romania)
Marco Foco	UNI (Italy)
Mark de Wever	ANSI (United States)
Mark Hoemmen	ANSI (United States)
Mark Zeren	ANSI (United States)
Mateusz Pusz	PKN (Poland)
Matt Calabrese	ANSI (United States)

Matthew Butler	ANSI (United States)
Matthias Kretz	DIN (Germany)
Michael Adams	SCC (Canada)
Michael Garland	ANSI (United States)
Michael Hava	ASI (Austria)
Michael L. Brock	ANSI (United States)
Michael Park	SCC (Canada)
Michael Price	ANSI (United States)
Michael Spencer	ANSI (United States)
Michael Wong	SCC (Canada)
Michał Dominiak	PKN (Poland)
Mike Spertus	ANSI (United States)
Mike Voss	ANSI (United States)
Mungo Gill	NSAI (Ireland)
Nat Goodspeed	ANSI (United States)
Nathan Myers	
Nathan Owen	ANSI (United States)
Nemanja Boric	ANSI (United States)
Nevin ":-)" Liber	ANSI (United States)
Nicolas Morales	ANSI (United States)
Nina Ranns	ANSI (United States)
Olivier Giroux	ANSI (United States)
Pablo Halpern	ANSI (United States)
Parsa Amini	ANSI (United States)
Patrice Roy	SCC (Canada)
Paul E. McKenney	ANSI (United States)
Paul Preney	SCC (Canada)
Peter Brett	BSI (United Kingdom)
Peter Sommerlad	SNV (Switzerland)
Philip Craig	BSI (United Kingdom)
Ran Regev	SII (Israel)
René Ferdinand Rivera Morell	ANSI (United States)
Robert Douglas	ANSI (United States)
Robert J. Simpson	ANSI (United States)
Robert Leahy	SCC (Canada)

Robert Seacord	ANSI (United States)
Roger Orr	BSI (United Kingdom)
Scott McMillan	ANSI (United States)
Scott Schurr	ANSI (United States)
Sean R. Spillane	ANSI (United States)
Sebastian Büttner	ANSI (United States)
Shafik Yaghmour	ANSI (United States)
Steve Downey	ANSI (United States)
Steven Vormwald	ANSI (United States)
Thomas Köppe	ANSI (United States)
Thomas Rodgers	ANSI (United States)
Tim Prince	ANSI (United States)
Tim Song	ANSI (United States)
Timur Doumler	BSI (United Kingdom)
Tom Deakin	BSI (United Kingdom)
Tom Honermann	ANSI (United States)
Tomasz Kamiński	AFNOR (France)
Vassil Vassilev	BDS (Bulgaria)
Ville Voutilainen	SFS (Finland)
Vincent Reverdy	AFNOR (France)
Walter E Brown	
William M. Miller	ANSI (United States)
Wyatt Childers	ANSI (United States)
Xavier Bonaventura	DIN (Germany)
Zach Laine	
Zhihao Yuan	ANSI (United States)