ISO/IEC JTC1 SC22 WG21 N4586 - 2016-03-30

Jonathan Wakely, cxx@kayari.org

February 29 - March 5, 2016 – Jacksonville, FL, USA

Chair: Clark Nelson

# WG21 2016-02 Jacksonville Minutes

## 1. Opening activities (Monday 9:00)

### 1.1 Opening comments, welcome from host

Hedquist welcomed everyone to the meeting and to Jacksonville. Maurer explained arrangements for the week.

### 1.2 Meeting guidelines

Every participant is responsible for understanding and abiding by the INCITS Antitrust Guidelines and Patent Policy and the ISO Code of Conduct.

### 1.3 Membership, voting rights, and procedures for the meeting

The chair explained the membership and voting rights for INCITS members, and voting for ISO global directory members.

### 1.4 Introductions

The attendees introduced themselves.

Those present included representatives of seven national bodies, Canada, Finland, Germany, Spain, Switzerland, UK, and USA.

### 1.5 Agenda review and approval

Agenda is in a revision of N4568, on the wiki.

Hedquist moved to adopt the agenda, Wakely seconded. Approved by unanimous consent.

### 1.6 Editor's reports, approval of working drafts

| Document | Editor's report | Prospective working draft |
|---|---|---|

| Document | Editor's report | Prospective working draft |
|---|---|---|
| C++ Standard | N4566 | N4567 |
| Library Fundamentals V2 TS | N4563 | N4562 |
| Ranges TS | N4561 | N4560 |
| Networking TS | N4576 | N4575 |
| Parallelism V2 TS | N4579 | N4578 |

Wakely explained that N4575 is very badly formatted, requested that editorial comments regarding non-technical content should be made against the latest sources not the draft in the mailing.

Halpern queried whether a revised Networking TS should be adopted at the end of the week instead of the poorly formatted one, but that was not deemed necessary.

Working papers adopted by unanimous consent.

Josuttis noted that N4578 was not in the mailing. Nelson and Sutter explained that the documents missed the mailing, but are in ISO LiveLink and on isocpp.org.

Voutilainen asked whether there is a work item for Parallelism v2. Sutter said he believed that was already done. Nelson asked for clarification of whether a second version of a TS requires a new work item. Sutter explained that there is a 1-month approval stage for a new work item, but no ballot needed to approve it.

### 1.7 Approval of the minutes of the previous meetings

| Meeting | Minutes |
|---|---|
| WG21 Kona | N4558 |
| PL22.16 Kona | N4559 |
| WG21 pre-Jacksonville administrative telecon | N4580 |

Telecon minutes were in the pre-meeting mailing. D4581 on the wiki has some corrections requested by Miller, to be published as N4581.

Tong queried the status of the Concepts telecon minutes adopted in Kona. The minutes approved in Kona were approved with corrections, and the revised minutes were published in the post-Kona mailing as N4557.

Hedquist moved to approve the minutes, Spicer seconded. Minutes approved by unanimous consent.

## 2. Liaison reports, and WG21 study group reports

See pre-meeting WG21 telecon minutes.

## 3. WG progress reports and work plans for the week (Core,

# Evolution, Library, Library Evolution)

Reports are in the telecon minutes.

Miller explained that Core would be prioritizing papers intended for inclusion in C++17. Requested that authors of such papers work with him to schedule discussion.

A number of issues that should be fixed before producing a CD in Oulu, will be working on those issues in Jacksonville.

At this meeting Core are moving tentatively ready issues from before Kona as well as

Revised version of Tentatively Ready issue lists, D0263r1, fixes some typos. Will be brought forward on Friday.

Voutilainen reported that Evolution plans to make C++17 feature-complete at this meeting, so that EWG will not send anything to Core in Oulu. Material for C++17 must be discussed this meeting.

Clow reported that Library are also concentrating on producing a CD. Have some high priority issues that still need to be resolved. There will be an issue processing evening session. All Ready and Tentatively Ready issues to be moved this meeting are in P0165 in the pre-meeting mailing. Requested any comments on those issues be brought to him before Friday.

Voutilainen asked whether LWG plans to handle NB comments on LFTSv2. Clow confirmed that some comments have been received and will be looked at, but not scheduled yet.

Yasskin reported that Library Evolution also plan to send all features for C++17 to LWG at this meeting. Prioritization of papers for C++17 will be discussed after Monday plenary, and in Friday plenary.

Spicer explained changes to the attendance records whereby sheets will not be passed around sub-groups each day. Those who require a record of their attendance should contact Spicer.

# 4. New business requiring action by the committee

Sutter repeated that the schedule for shipping C++17 requires a CD to be shipped at the end of Oulu. Aim to build consensus on C++17 candidates.

There are five proposals to add published TS content to C++17: Special Math, File System, Parallelism, Concepts, Library Fundamentals v1. Sutter sought feedback before the meeting regarding people's current dispositions on those proposals. Aim not to create consensus in Monday plenary, but to discover whether there is consensus, to guide discussions during the week. There would be a presentation of each proposal followed by a straw poll of those in the ISO global directory.

Sutter made some personal recommendations for questions to ask ourselves when considering material for inclusion in the IS. Are we 98% sure we won't want a breaking change and wish for a

time machine? And does the wording answer 98% of implementation questions? Spertus said that "will we wish we had a time machine?" is not a good benchmark, because all projects could always benefit from a time machine, and so that would lead to striving for perfection. Suggested "would your expected future happiness be lower if we put this in?" as an alternative question. Winters said that "are we likely to need a breaking change?" is a good question.

**Mathematical Special Functions**

Naumann gave a presentation on Special Functions. Started by saying these functions are not "special" they are very basic mathematics, widely used. They are basic ingredients for mathematical modelling, but non-trivial for regular users to compose. The content, usage, and wording are all known and stable, there would be no benefit to delaying. N3060 is the FDIS with the complete wording. Carruth asked whether the Boost implementation is good enough to form the basis of a standard library implementation, Naumann said yes. Plauger explained that these are simply the next set of widely-used and well-understood mathematical functions above the sine, cosine etc. functions in the C standard library. Halpern asked whether this has been proposed for C, Nelson confirmed that it's a TR for C but not in the C IS.

51 decided in favor, 0 decided against, 18 undecided.

**File System**

Dawes presented P0218R0, proposal to adopt the File System TS for C++17. More than a decade of implementation experience. The TS process worked, got implementor experience, with wording and specification issues reported by all implementors. Large userbases for Boost and Dinkumware implementations. API had already gone through lots of evolution in Boost before the TS. Tong asked whether there is any implementation experience on systems which do not have a hierarchical filesystem. There has not. Yasskin asked how to teach this for use on multi-user systems. Dawes said to do it the same way as for other languages with similar facilities. Filesystems are inherently racy. Crowl noted the API overloads most operations with exceptions and without and asked whether future evolution regarding alternative error-handling schemes would impact how the File System library should be specified. Asked whether we should delay the TS to wait for alternative schemes. Sutter said to discuss that in sub-groups. Plauger reported that the error-handling scheme is liked by his users, and that it is also in use on systems with flat filesystems.

39 decided in favor, 5 decided against, 28 undecided.

Nobody was aware of any National Body positions that would result in a No vote on C++17 if it included File System.

**Parallelism**

Hoberock presented overview of Parallelism TS. Multiple implementations exist. Getting these into C++17 is a first step that would enable later evolution regarding executors, distributed systems and more. Meredith asked whether adding the new overloads to namespace std is still open for technical discussion and Yasskin confirmed that is in scope. Vandevoorde asked whether anything in the

second version of the TS tweaks anything that is being proposed from the first version, but it does not. Halpern clarified that there might be a future ABI break if the parvec policy is made stateful. Brown asked how much user feedback has been received on the existing implementations and Hoberock said most of it was received in response to prior implementations and guided the design of the TS.

33 decided in favor, 3 decided against, 34 undecided.

**Concepts**

Sutton gave a presentation on Concepts TS, describing the advantages, history, and previous wording reviews. Feedback has been positive. New proposals are being written using Concepts. Yasskin said he had heard some parts are more contentious than others, asked whether taking a subset for C++17 would make sense, and what is the smallest subset that would make sense. Sutton said it doesn't make snse to split it up. Smith asked whether there has been consideration of adjusting syntax based on feedback to the TS. It was discussed in Kona and it was decided not to change anything. Ballman asked if there were any implementations created purely from the TS, based only on the final specification. There are not. Meredith asked whether changes could still be mode during this meeting and in Oulu, Voutilainen said fixes to features in the standard are always in scope. Yasskin asked whether any of the implementation requires greater-than-linear algorithms, Sutton said that using a large number of disjunctions in a concept may encounter the worst-case behavior, which is quadratic. Sutton stated a hope that implementors of the core language feature would also provide a set of core concepts for users to build on. Voutilainen reported implementation experience of using concepts to simplify implementation of existing standard library features. Sutter also noted that there is a Ranges TS underway using Concepts which is likely to ship around the time of C++17 which does include a number of basic concepts, so implementors and users would not need to invent their own. Dawes reported that his small experience of using the GCC implementation went well. Stressed that conceptifying the standard library is a major undertaking and must be done separately to the core feature, as a second step, rather than trying to do both together. Dos Reis asked for a comparison of experience using C++0x for iterator facade and using the Concepts TS. Dawes said that specifying it in C++11 was too difficult and the proposal was dropped. The new proposal using Concepts TS is no longer a heroic effort. The concern that there is no way to add definition checking later was raised and Sutton responded that the technical solution is known and documented, but that doesn't mean that we necessarily want definition checking.

27 in favor, 14 against, 32 undecided.

There is no National Body that would definitely vote No if C++17 included Concepts.

**Library Fundamentals**

Dawes presented P0220R0 and explained that a piecemeal approach to adopting pieces of the TS adds a lot of overhead, processing numerous proposals. It would also risk losing small but useful components. The suggested approach was to propose everything and let small pieces be struck out during the week to create a revised proposal. Spertus agreed that adding things piecemeal would make it very likely that things would get dropped. Wakely pointed out that unlike the other papers discussed Fundamentals TS is not a group of related, interlinked proposals, it is a grab bag of

orthogonal pieces. Sutter clarified that the poll is on taking all of LFTSv1.

13 decided in favor, 0 decided against, many undecided

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

Maurer clarified rooms available for evening sessions.

# 6. WG and SG sessions

The WG and SG chairs must arrange for any proposals to be written up in the form of a motion, and made available by 2:30 Friday.

# 7. Review of the meeting (Friday 2:30)

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

**SG5: Transactional memory (Wong)**

Wong reported that SG5 did not meet. There was a paper in the pre-meeting mailing but not asking for TM to go into the IS. Will encourage people to try the features and await feedback.

**SG6: Numerics (Crowl)**

Met on Tuesday for most of the whole day. Reviewed a number of papers and made some progress but didn't produce a draft of the TS. Trying to corrall the various proposals that keep coming in apparently with no knowledge of SG6's existence.

**SG7: Reflection (Carruth)**

Met Monday night, very productive evening session. 3 proposals presented representing diferent directions. Good discussion after the proposals, now pursuing a single direction. Likely to meet on Friday evening to have more discussion with the author. Hoping to bring something to LWG in Oulu.

**SG10: Feature test (Nelson)**

There was a draft of SD-6 in the pre-meeting mailing. Going to update the document on isocpp.org immediately with that, with one fix for an `owner_less` macro. There will be a load of new stuff to incorporate after the meeting.

Voutilainen said EWG didn't spend time coming up with macros. Nelson responded that it's probably OK for those discussions to happen in CWG and LWG.

**SG12: Undefined and unspecified behavior (Dos Reis)**

Planned to meet but didn't due to being busy with Evolution proposals.

**SG14: Games & low latency (Wong)**

A number of papers in the mailing, some treated through SG1. Big one on massive parallelism. Will hold an official SG14 meeting on Monday 14th March at 345 Spear Street in San Francisco, hosted by Google. The embedded group has now been folded into SG14.

Nelson asked how many people are involved. Wong said meetings are 10-15, about 100 on the mailing list, calls attended by about 30 people. Said that they do use the wiki, but many discussions happen on the mailing list.

Price thanked Wong for involving that community, to acclamation.

**SG13: I/O (Sutter)**

Sutter reported there was a meeting, with some feedback prepared for the author.

**SG1: Concurrency (Boehm)**

Met all week, started discussing some C++17 issues. Unanimous support within SG1 for move Parallelism TS v1 into C++17. No proposal to move Concurrency TS v1 into C++17. Also discussions on the destructor behavior of `std::thread`. SG1 initially decided not to change anything, but a larger evening session reversed that discussion. Paper not being dealt with at this meeting, but will be reviewed at the next meeting. Discussed `memory_order_consume` but nothing is going to happen for C++17, planning to carefully word things to suggest to users that they don't use it until it's fixed. Talked about signal handlers again, in order to move away from the wording that comes from C but makes less sense for C++.

Also talked about future Technical Specifications, Parallelism v2 and Concurrency v2, hoping to be feature complete at a Issaquah meeting, for a PDTS ballot for Kona. Talked about SIMD vector support, specifically adding an execution policy, made some progress. Centrepiece of Concurrency v2 is still expected to be executors. Discussion ongoing, slowly, with proposals converging, slowly. Also working on synchronic types and counters and queues. Atomic views and atomics for flotaing point likely to find heir way into Concurrency v2. Also discussed hazard pointers and RCU, which may or may not end up on that schedule. Will discuss issues Saturday morning.

## Evolution (Voutilainen)

Jackson Ville reported that the major topics were to finish the design of important C++17 facilities, which he considers done. Gave guidance for Technical Specifications for Modules, Concepts and

Coroutines. Discussed two papers on coroutines but didn't take a direction poll in EWG. Approved cancelling the Arrays project. Looked at future material such as Contracts, approving the authors direction, but didn't decide where it will land.

Approved several proposals. Some had been approved before this meeting and went straight to Core. Plans to write an overview paper for the post-meeting mailing listing the things going into C++17.

Had a handful of rejected proposals. Some papers were not discussed because a lot of time was spent, particularly on Modules. Might reach those on Saturday.

Reviewed and postponed some proposals, such as multi-range `for` loop which might be superseded by other facilities, and structured bindings.

Considers work done for C++17. Future work is beginning. In good shape for C++17.

Gave humungous thanks to Andrew Pardoe for taking minutes all week, to acclamation.

Miller explained that usually Core would bring forward proposals for polls. In the case of Concepts CWG declined to propose the Concepts TS for inclusion in C++17, as Core were not comfortable with going forward with the proposal at this time. Based that decision on the TS being less than a year old, with one complete implementation. As a result of limited implementation and limited user experience Core felt it should not become part of the IS at this point. Unfortunate that the ship vehicles aligned in this way so that there was insufficient experience at the time when the decision needed to be made. Core decided not to propose it, but if the committee as a whole want to proceed with the proposal then Core will be OK with that and will integrate the wording and take the Concepts issues into the Core issues list and deal with them normally. But did not feel comfortable recommending it for inclusion.

Sutter explained that the intention is to go through each motion and come back to Concepts at the end for discussion.

## Library Evolution (Yasskin)

Had about 65 papers, nine not looked at yet. Forwarded 21 to LWG for C++17. Discussed 33 others. Rejected about two of those. Believes that LWG will try to process the forwarded papers for C++17, but may not have time. LEWG will be meeting again on Saturday. Also dealt with five issues from the LWG issues in LEWG status, sending them back to LWG.

## Core (Miller)

Busy week in Core. Essential did the entire week on processing proposals for C++17, doing no issue processing. Will do issue processing on Saturday. Some issues are in Ready status after Kona and some that are in Tentatively Ready, either in that status since Kona, or moved to that Status between meetings.

Motions for Ready issues that were decided by the full group during the meeting, and Ready issue

decided by a smaller group during telecons. Ready issues paper, P0167R2, is the same as the paper in the pre-meeting mailing modulo typos. Tentatively Ready issues paper, P0263R1, removed an erroneous SUPERSEDED marker on 1496, 2066 was resovled by 2109, and 2143 resolution was withdrawn after deciding no normative change was needed, so a note will be proposed for Oulu, and 2162 was withdrawn as it is superseded by P0018R3.

Will decide direction for unresolved Priority 1 issues on Saturday, review them by telecon, and produce Tentatively Ready resolutions for Oulu.

No new issues opened since Kona will be looked at until Oulu.

Proposals in today's motions include three attributes; extensions to aggregate initialization to allow classes with base specifiers (which makes more things static initialization, making more things constant, which can result in binary incompatibility); constexpr lambda; unary folds and empty parameter packs; extension to range-based `for`; lambda capture; `enum` initialization to allow enumeration types to act as opaque integer typdefs; hexadecimal floating-point literals; unified call syntax; and modules (requesting a new work item for a TS).

Still pending for Oulu, and still potentially reaching C++17, are constexpr if; default comparisons; coroutines; forward progress guarantees; `std::byte`; order of expression evaluation; dynamic allocation of over-aligned data; operator-dot.

Voutilainen suggested that LEWG should take a look at the default comparisons due to impact on library features. Meredith said that LWG would need to look at it for impact on the existing library as defined today.

Meredith queried the status of deprecating dynamic exception specifications. Miller apologized for forgetting to mention it, it is also on the list of proposals still pending for Oulu.

## CWG Motions

**Motion 1** Move to accept as Defect Reports the issues in P0167R2 (Core Language "ready" Issues) and apply their proposed resolutions to the C++ working paper.

Approved by unanimous consent.

**Motion 2** Move to accept as Defect Reports the issues in P0263R1 (Core Language "tentatively ready" Issues) and apply their proposed resolutions to the C++ working paper.

Approved by unanimous consent.

**Motion 3** Move to apply to the C++ working paper the proposed wording in P0188R1 ("Wording for [[fallthrough]] attribute").

Approved by unanimous consent.

**Motion 4** Move to apply to the C++ working paper the proposed wording in P0189R1 ("Wording for [[nodiscard]] attribute").

Miller summarized the purpose of the attribute.

Approved by unanimous consent.

**Motion 5** Move to apply to the C++ working paper the proposed wording in P0212R1 (“Wording for [[maybe_unused]] attribute”).

Approved by unanimous consent.

**Motion 6** Move to apply to the C++ working paper the proposed wording in P0017R1 (“Extension to aggregate initialization”).

Approved by unanimous consent.

**Motion 7** Move to apply to the C++ working paper the proposed wording in P0170R1 (“Wording for Constexpr Lambda”).

Approved by unanimous consent.

**Motion 8** Move to apply to the C++ working paper the proposed wording in P0036R0 (“Unary Folds and Empty Parameter Packs (revision 1)”).

Approved by unanimous consent.

**Motion 9** Move to apply to the C++ working paper the proposed wording in P0184R0 (“Generalizing the Range-Based For Loop”).

Approved by unanimous consent.

**Motion 10** Move to apply to the C++ working paper the proposed wording in P0018R3 (“Lambda Capture of `*this` by Value as `[=,*this]`”).

Miller described how lambda capture of members works, as it captures the `this` pointer. The proposal allows the entire object to be captured.

Crowl asks if there is any backwards compatibility problem. Maurer stated the new syntax was not previously valid. Voutilainen clarified that it is an extension not a change.

Approved by unanimous consent.

**Motion 11** Move to apply to the C++ working paper the proposed wording in P0138R2 (“Construction Rules for enum class Values”).

Lavavej asked if it only affects enumeration types with no enumerators. It was clarified that it affects types with enumerators.

Van Eerd asked if it changs existing code. Miller responded that it doesn’t because the conversions it allows were previously disallowed.

Approved by unanimous consent.

**Motion 12** Move to apply to the C++ working paper the proposed wording in P0245R1 ("Hexadecimal floating literals for C++").

Vandevoorde asked whether the rules are the same as for C. Miller said the wording is tweaked slightly but the intent is that it would be source compatible.

Approved by unanimous consent.

**Motion 13** Move to apply to the C++ working paper the proposed wording in P0301R0 ("Wording for Unified Call Syntax":).

Spicer said he would have preferred a different approach. If code relies on calling a free function it could now be broken by adding a member function that is a much poorer match.

Van Eerd asked if this interacts with operator-dot. Carruth stated it would.

Winters agreed with Spicer that this is not the worst thing that would happen to maintainability and understandability of code, but was concerning.

Carruth said this makes name lookup and overload resolution more complicated.

In favor: 24 Opposed: 24 Abstained: 21

The motion failed.

Stroustrup commented that discussions have happened for some time in Evolution with positive results, but negative votes were only seen at plenary. That leads him to believe that things should receive a lengthy description of the pros for the feature. Winters agreed that sometimes at plenary only cons are heard, which might say votes.

Voutilainen was asked how strong the consensus in EWG was and what he planned to do with the proposal. He said there was clear consensus but future directions would depend on the author. Sutter said he thought that the authors had removed the controversial part of the paper, but clearly people still had objections. Spicer said it's always the case that bringing proposals to a larger audience can always bring surprising results.

Nelson suggested holding further discussion of this at the end and consider another poll. Sutter suggested that there is not consensus now, that we could hold further discussion tomorrow and take the poll tomorrow.

Voutilainen requested that objections be brought to Evolution.

Yasskin suggested that the poll was taken, and that further discussion should wait for Saturday.

Van Eerd suggested this could have ben run past LEWG and LWG.

Stroustrup expressed surprise at the suggestion that proposal authors should make supporting

arguments, as he had heard no new objections and that it would have contradicted the longstanding tradition to avoid length technical discussion in plenary.

Sutter began a technical discussion by summarizing that the proposal and the objections were not new. Stroustrup explained that non-member and member call syntax being different required a conscious and intentional decision to be made when designing an API. Originally the proposal would allow x.f() to find non-members and f(x) to find members, but in each case the old meaning would be preferred. There was a lot of concern about x.f() finding members, so the authors decided that having unity in one direction was better than none, so proposed only allow f(x) to find x.f(). A later tweak was made to handle cases where the new rule introduced an ambiguity, so that the old style would be preferred, in order to avoid breaking code. This eases the definition and interface specification of templates, without breaking code. Some people wanted additional opt-in mechanisms to enable the new behavior, which was discussed in great detail, but rejected. The proposal adds a feature available in other language. Allows defining helper functions more conveniently. Sutter added that all the concerns raised previously were considered already.

Spicer liked the goal of the proposal, but wanted opt-in which would then consider members and non-members equally in an overload set.

Finkel had no doubt this would simplify things for library developers, but would make things harder for novices. Stroustrup strongly objected to that characterization.

Halpern asked whether it had been implemented and tested on a sizable codebase. Stroustrup said yes. Sutter said that since it didn't break code, the major objection he was aware of was for future maintenance, not breaking old code.

Meredith said this enables putting a stop to the creeping additions of free functions to the library such as `size`, `data`, `begin` etc.

Smith said that if you're comfortable with ADL then this is not more surprising. Spicer said the difference between this and ADL is that this allows something that is a much worse match, because the functions are considered in separate overload sets.

Yasskin said he'd feel more comfortable being able to see the feature in use for a long period of time, and would feel more comfortable seeing it in a TS.

Sutter took a poll to determine consensus on the proposal

| SF | F | N | A | SA |
|---|---|---|---|---|
| 13 | 11 | 18 | 21 | 5 |

**Motion 14** Move to direct the convener to request a new work item for a Technical Specification on C++ Extensions for Modules and create a working paper with P0143R2 ("Wording for Modules").as its initial content.

Approved by unanimous consent.

## Library (Clow)

Very full week, didn't get to all papers. A dozen pending for Oulu. Will be having some telecons for issue processing. Processed some issues this week. There had been a large uptick in LWG issues mostly because the File System issues list was merged into the LWG list. Reviewed 19 papers, adopted 14. Resolved all priority 1 issues except one. Thanked the scribes.

## LWG Motions

**Motion 1** Move we apply the resolutions of the following issues in "Ready" status from P0165R1 to the C++ Working Paper

Issue numbers: 2276, 2523, 2537

Clow clarified that the paper has two more issues, but as they refer to the Arrays project they are no longer needed. Maurer suggested an updated paper without those two would have been useful.

Approved by unanimous consent.

**Motion 2** Move we apply the resolutions of the following issues in "Tentatively Ready" status from P0165R1 to the C++ Working Paper:

Issue numbers: 2192, 2450, 2520, 2545, 2557, 2559, 2560, 2565, 2566, 2571, 2572, 2576, 2577, 2579, 2581, 2582, 2583, 2585, 2586, 2590

Approved by unanimous consent.

**Motion 3** Move we apply the resolutions of the following issues in "Tentatively Ready" status from P0165R1 to the Library Fundamentals Working Paper:

Issue numbers: 2522, 2539, 2558, 2574, 2575

Approved by unanimous consent.

**Motion 4** Move we apply to the C++ Working Paper the Proposed Wording from P0024R2, The Parallelism TS Should be Standardized.

Carruth asked how long the Parallelism TS had been published. It was clarified that it was under a year. Winters said less than a year doesn't seem like enough time to gain experience. Sutter said there is no fixed rule or arbitrary time period that must pass. Yasskin said the purpose of a TS was to let us get more comfortable with things, and experience is just one way we get comfortable.

Smith said there are six partial or complete implementations, which is more than he can remember for any proposal, but he doesn't know how much user experience there is.

Finkel said the TS and the functionality it represents was very important to the national labs, but he would be voting no because getting more experience is even more important.

Garland said the paper lists some experience. The oldest implementation is NVIDIA's Thrust which is almost 10 years old, and has been shipping for five years. They consider it to have significant experience.

Spicer said that moving something into the working draft makes it much harder to make breaking changes and so he would like to know the interfaces are not going to need changes in six months. Garland said that almost everything in the TS is an overload of an existing STL algorithm, so the interfaces are based on those. They haven't needed to change the interfaces of their implementation in the ten years they've been using it. Lelbach echoed that they do not believe any changes will be needed.

Winters asked for "person years" of experience. Garland said they have many users, they surveyed users and had 1500 responses.

Sankel said he heard that the answers to "will we need to change this" is no.

Halpern said the functions are very basic. It's very solid stuff that isn't going to change other than adding more overloads.

Stroustrup said we should not be delaying things due to fear of breaking things. Users don't have infinite time. Thinks there is not enough trust in the sub-groups in the committee.

Edwards said SG1 discussed need for future breaking changes and had already stripped those things out before it became a TS.

Carruth expressed some concern as an implementor. Existing implementations are not standard library implementations, they are add-ons that sit on top of it. There might be problems when adding these to a full standard library. Existing implementations are also not portable ones, so we don't know what portability or ABI stability problems might arise when adding these features to portable standard libraries shipping today. Would like the time to implement, integrate and deploy this. Sutter asked for an expectation of how long it would take to implement. Carruth said not ten years, but might take six months.

Meredith said he was concerned at the speed we're adopting TS's in general, but that isn't a concern with this one. Noted there was concern being shown now than on Monday, would be nice if that was brought to the chairs in advance since it was known this was going to be a motion.

Van Eerd asked why this material went into a TS in the first place rather than proposing it for the IS immediately. Halpern said that SG1 used a TS for everything they were working on, not because of lack of readiness.

Orr said he was pleased this was almost unchanged in the six months since publication. Wakely echoed that this was brought to LWG in almost its final state, didn't change much during review, and had not changed significantly since.

In favor: 53 Opposed: 2 Abstained: 16

The motion passed.

**Motion 5** Move we apply to the C++ Working Paper the Proposed Wording from P0226R1, Mathematical Special Functions for C++17, v5.

In favor: 62 Opposed: 1 Abstained: 5

The motion passed.

**Motion 6** Move we apply to the C++ Working Paper the Proposed Wording from P0220R1, Adopt Library Fundamentals V1 TS Components for C++17 (R1).

Dawes explained that some components had been dropped since the P0220R0 paper. Clow clarified that the proposal is to take the wording from V2, but these are all components that existed in V1.

Meredith explained that the pieces of V1 missing are invocation traits, changes to uses-allocator construction and changes to make `function`, `promise` and `packaged_task` to use type-erased allocators.

Approved by unanimous consent.

**Motion 7** Move we apply to the C++ Working Paper the Proposed Wording from P0218R0, Adopt the File System TS for C++17.

Clow reported that LWG had not had time to review this wording during the meeting. Clow said we could drop the motion, we could try again in Oulu, or LWG could do the review on Saturday morning and do the motion on Saturday.

Halpern asked for clarification that there was no urgency to move the proposal at this meeting, and it could still be done in Oulu. Sutter confirmed that but said it was good to get it off our plates while it's fresh in our minds.

Crowl asked whether it was an optional feature or required of all hosted systems, Clow confirmed it was required of all hosted systems.

In favor: 52 Opposed: 4 Abstained: 15

The motion passed.

**Motion 8** Move we apply to the C++ Working Paper the Proposed Wording from P0033R1, Re-enabling shared_from_this (revision 1).

Approved by unanimous consent.

**Motion 9** Move we apply to the C++ Working Paper the Proposed Wording from P0005R4, Adopt `not_fn` from Library Fundamentals 2 for C++17.

Clow gave a reminder that this was originally proposed in Kona, but following objections to removal without deprecation it was withdrawn and re-proposed with deprecation instead of removal.

Approved by unanimous consent.

**Motion 10** Move we apply to the C++ Working Paper the Proposed Wording from P0152R1, constexpr atomic::is_always_lock_free.

Approved by unanimous consent.

**Motion 11** Move we apply to the C++ Working Paper the Proposed Wording from P0185R1, Adding [nothrow-]swappable traits, revision 3.

Approved by unanimous consent.

**Motion 12** Move we apply to the C++ Working Paper the Proposed Wording from P0253R1, Fixing a design mistake in the searchers interface.

Van Eerd asked whether it breaks code. Meredith confirmed it would only break people using searchers directly. Voutilainen pointed out that since the fix is only being applied in the IS not TS V2 it didn't break anything,

Approved by unanimous consent.

**Motion 13** Move we apply to the C++ Working Paper the Proposed Wording from P0025R0, An algorithm to "clamp" a value between a pair of boundary values.

Approved by unanimous consent.

**Motion 14** Move we apply to the C++ Working Paper the Proposed Wording from P0154R1, constexpr std::hardware*{constructive,destructive}*interference_size.

Approved by unanimous consent.

**Motion 15** Move we apply to the C++ Working Paper the Proposed Wording from option #2 in P0030R1, Proposal to Introduce a 3-Argument Overload to std::hypot.

Lavavej asked whether the "sufficient additional overloads" applies to this new function. Clow said that rule applies globally so includes this new function. The proposal states that option #2 means that wording applies to the new function.

Approved by unanimous consent.

**Motion 16** Move we apply to the C++ Working Paper the Proposed Wording from P0031R0, A Proposal to Add Constexpr Modifiers to reverse_iterator, move_iterator, array and Range Access and apply the resolution of LWG Issue 2296 - std::addressof should be constexpr. This issue is published in P0304R0.

Approved by unanimous consent.

**Motion 17** Move we apply to the C++ Working Paper the Proposed Wording from P0272R1, Give 'std::string' a non-const '.data()' member function.

Approved by unanimous consent.

Lavavej noted a typo in the non-normative Annex C addition, to be corrected editorially.

**Motion 18** Move we apply to the C++ Working Paper the Proposed Wording from P0077R2, is_callable, the missing INVOKE related trait.

Approved by unanimous consent.

## EWG Motions

**Motion 1** Move to adopt the ISO/IEC TS 19717:2015 Technical Specification for Concepts into the C++17 working paper.

Sutton explained that it has been implemented and is shipping and is in use and popular. Voutilainen echoed the sentiment.

Maurer noted that the motion is for the published TS, not the amended draft after Kona, but Sutter assumed corrections would be made editorially. Maurer said he would vote against as he thought it needed about a year more experience.

Crowl asked for user feedback, who was using it, and how heavily. Niebler said that he and Carter were using it. They considered it a fabulous success compared to trying to use ISO C++. Writing algorithms was easier and diagnostic were better and would only get better. Crowl asked how many outside the committee.

Coe asked whether some of the controversial syntaxes that had caused concern could still be removed if this is added to the TS. Sutter confirmed that NB comments can request fixes to things in the draft or removal of features (but not additions). Voutilainen said that removing syntax would cause a problem for people using those forms.

Dawes reported that he used Concepts for the iterator facade proposal in the pre-meeting mailing, using the pre-release GCC for about a month. Was tremendously impressed. The error messages were not up to scratch yet. Would vote against it because it's not in a released compiler yet.

Honermann said it's great and easy to use but still unclear how many people are using it. Experience with the compiler suggest that it would not be possible for someone to create a really significant body of work.

Stroustrup reported a dozen or two dozen students downloading and using it to do work they would have been unable to do without concepts, they were amazed. Allows shorter code and less error-prone code. Recalled when function types were very controversial and making them controversial was considered outrageous and would break code, which it did because most code had bugs in. The additional syntaxes came following user experience. Writing something like mergeable needs constraints that are longer than the algorithm, without the terse syntax. People prefer the shorter form. People who are not serious C++ programmers want the shorter form. Regarding definition checking, it's been talked about for years, but it's not clear it's desirable. The plan was to do the work in stages, and this is the first stage. C++ is in a very competitive environment for mindshare, would hate for C++17 to be looked at as a few small features and libraries. This proposal is the best thing for showing things that are brand new. Without a big change in 2017 we would be back on a 9 year cycle.

Brown agreed with Stroustrup but would still vote against. His experience is that he loves it. He has been using GCC trunk exclusively for some time. His experience is mixed. He loves `requires` clauses, make life much easier. Has tried a private implementation of the standard library. More than two thirds of the time he can map the specification in the standard to code almost verbatim, and when it doesn't work it tends to be because the spec is confused. That said, chosen after experimentation to avoid all use of short forms. Tried to write some papers to address some concerns. Strong supporter of concepts, but voting against today.

Van Eerd said it was a TS voted out in October. We are holding it to a higher standard than some TS's, but it deserves to be held to a higher standard.

Orr reported than some BSI members have used it and given positive feedback. But unclear how much in-depth experience there is. Very new, the official BSI document was oinly received a week ago, dated 22 February.

Meredith said ideally it would be an older document we had more experience with, but that's not the world we live in. In favor of accepting it, and trying to fix some things for C++17. The alternative is that it has to wait for C++20, by which time more and more people learn the template hackery and put more of that into their code. The committee would also waste more time adding SFINAE in places to work around the lack of concepts. Needs to be accepted today to give us more time with it in the draft, putting it in after Oulu would be too long.

Voutilainen said he didn't think the design needed changing, he thinks it's fine.

Ballman said he loved the feature, but as an implementor he didn't think the specification was clear enough to produce an independent implementation. Really wants the feature but didn't see it as feasible to implement in a compiler.

Spicer agreed, though that in ten years this will be how people write and use templates, but it's far too soon to put into the standard. For a feature this complicated we need broader implementation and usage. In a year's time there would be two or more additional implementations. Worried about concerns with making breaking changes when it's still only a TS, need to be able to make breaking changes while it's a TS. Constexpr had about 70 issues, lambdas had 50 issues, only had about 30 so far for concepts and would expect about 150 before we're finished with it. From a project management point of view, putting in a huge feature that we don't have complete confidence in is a huge risk. Agree we need to take some risks, but also need to remember mistakes of the past, export templates being one example, C++0x concepts being another. For some features it takes time for problems to play out, for example rvalue references seemed like a small feature, but exception safety issues arose, and the notion of value categories had to be introduced. Disagreed with a comment from Dawes on Monday saying it would not be practical to release language support for concepts with library support simultaneously. Need to wait for the new TS process to mature and wait for things to come out of the pipeline. Much easier to add something to the IS than to remove it, things should be fixed first because taking them out later is very, very hard. Regarding definition checking he thinks it's not essential, without definition checking in C++0x we probably would have had them in the standard. It's a hard problem and should not be holding this up.

Nelson requested skipping repetitive comments.

Vandevoorde agreed with many points. Pointed out a paradox that if it was in the standard he believed implementations would take longer to come out than if it was in a TS. EDG have been at the leading edge of many features, and have learned from mistakes. They have been proactive with coroutines, because they had been happy to warn customers that it's experimental and can break between releases. They can start implementing and shipping TS things sooner because they don't have to commit to finished, complete support.

Halpern thinks if it goes into the IS we can get NB comments. If we have to yank it out and it delays the next standard, that would not bother him too much. People should know that they might be signing up to cause delays if they vote this in.

Smith says his personal bias is that he values the experience of C++ professionals more than learners and students. For such people he has heard that they need to be able to know if something is a template or not, so some breaking syntax changes would need to be made.

Fracassi said definition checking was the sticking point for them. They had been told it would be possible, if they waited. Now they are being told it might not be desirable and they might not get it at all.

Dos Reis responded to the statement that it's easier to put something in the IS than to remove it. Said it's much harder to get stuff through Core than to remove it. Today, in C++14 constexpr is one of the things people love, and it was delayed for a long time due to issues needing to be addressed. The same thing is happening with concepts. Looking at the successful languages, they are all corporate owned. In C++ we have convinced ourselves we need to wait 5 or 10 years to wait and fix all bugs. We're not helping the community by being extremely conservative.

Myers requested a show of hands of people who said they loved concepts but thought it was too soon, would they be willing to wait for C++18? Stroustrup said shipping in 2017 was very important. Myers asked whether he would rather have C++17 without concepts or C++18 with. Stroustrup said he'd rather have C++17 without, which he really did not want.

Crowl expressed confusion over when we let users use something. Sounds like people have different expectations over when something should be put in a shipping compiler and we should ask for people to use it. Would like people to suggest when they would want TS-level features to be in users' hands.

Ballo said he had no doubt the concepts designers had great ideas for definition checking, but that the devil is in the details and that the risk of big issues coming up that would require breaking changes. Would prefer not to do something now that requires breaking changes later.

Van Eerd made a comment on the major/minor release train not working. Prefers to think about release trains not major/minor.

Tong said he was not worried about people using it before it's ready to go into the next standard. Suggested a different analogy to release trains, airport security. Would be happy if this was the first thing we added to the draft after C++17, people would try it out, but we'd have time to correct it. If we drop it in now we don't have time to correct it before the IS.

Stroustrup said they had promised to look into definition checking, but after doing the interface part. EWG voted that they do want to see it, so it's not going to go away. There are theoretical models showing how it would work. Referred to Meredith's point about concepts being able to stop the proliferation of template hacks to work around the lack of it.

Finkel pointed out that the implementation in GCC trunk would get three orders of magnitude more users in a month or two when it is released. Waiting would get more feedback.

Wong said there's no doubt there will be bugs, but still voting for the feature. At some point you have to have faith that the people working on it know what they're doing.

Spicer said in his experience you really want to make sure a big feature like this is in really good shape to what you finally want in the IS, because it's hard to fix. Says that as somebody who spent two years of his life trying to get something out of C++98, failed, spent two years implementing it, then it got taken out. Letting the TS process play out is important to be sure we got something right.

Brown said if after hearing all of this you are still undecided, bear in mind that the core group have declined to bring this motion forward. If you're not sure about the technical merits bear that in mind.

Sutter thanked the committee for allowing him to persuade them to switch to a release train model. Each release has been smaller, but due to the frequency of releases more has happened in the last five years.

In favor: 25 Opposed: 31 Abstained: 8

The motion failed.

Sutter asked "does anyone know of a national body that would be likely to vote No on C++17 itself because concepts are included? (not asking for an individual position, but a national position)" US doesn't know what it's position would be. Germany are likely to vote No on an IS with Concepts. UK would vote No if the feature were in an IS unchanged, without resolving some issues. Canada does not know if it would vote No.

Sutter asked if people would be willing to consider adopting concepts into the draft immediately after after C++17. There were objections to that poll and it was not taken.

**Motion 2** Move to direct the convener to inform SC22 we will not produce an Arrays TS and close the Arrays TS work item.

Sutter announced that the wording of the motion had been revised in a more formal manner.

Lelbach asked for confirmation that the authors of some array-related proposals are targeting the Library Fundamentals TS instead. It was confirmed that there was no consensus on the content of an Arrays TS, but that would not stop related work.

Approved by unanimous consent.

## Other polls

Sutter explained that Evolution proposed coroutines (P0057R2) proceed for C++17, but others had suggested a TS.

Sutter wanted to see whether the IS or TS had stronger consensus.

Winters asked how many developers have been using the implementation, for how long, and when was the last breaking interface change. In use since 2012, and had two lexical changes since then: dropping `resumable` from the function declaration, and changing the spelling of the keywods in Kona. Those were changes for libraries using the features. Huge applications using the feature. Know of at least 20 users outside Microsoft. Implementation shipping since 2014, partial independent implementation from EDG. HPX using it satisfactorily.

Vandevoorde implemented it, following revisions in the spec. Implemented it independently, based on the spec only. Recommended a TS, in order to be able to make further breaking changes if needed.

Riegel said there are a number of difficult trade offs to be made in the design to get good performance. This requires non-trivial work involving many interactions. Putting it in a TS enables us to gain experience. Please consider whether you can confidently put this into an IS without risking breaking changes. There have been significant changes to the proposal since it was first proposed to WG21, it still seems like a moving target. Sutter asked whether Red Hat would produce an implementation, and whether a couple of years would be time to get the needed implementation experience. Riegel confirmed yes.

Smith said he had done a partial implementation and had done some work on the spec, so not entirely independently. He thought the wording was good and ready to go into some document, but he thought that should be a TS.

Yasskin said Chrome would want to use it, but would want to be confident the feature was ready for the 20 year lifetime over which they would be using it, so would like it in a TS so others have time to improve it.

Stroustrup said this has been reviewed and is what Evolution want, it's unbeatable performance for some use cases. Want this for speed, not elegance.

Sutter asked if adding this to the IS blocked adding other coroutines later. Riegel said it doesn't prevent anything else, but it locks people into one interface that doesn't easily allow adapting code to a different interface.

Voutilainen said this feature fails the time machine test, in that if Riegel is right that the same advantages can be achieved differently then we would definitely regret adopting this proposal now.

Sutter took a poll to see who would be in favor of coroutines targeting the C++17 IS

In favor: 9 Against: 36 Abstained: 19

Poll to recommend coroutines targeting a new TS, involving a new work item

In favor: 57 Against: 0 Abstained: 7

Nelson moved to thank the host.

Bastien moved to thank Space-X for their rocket launch.

Brown moved to thank to working group chairs, convener and scribes.

# 8. WG and SG sessions continue (Saturday 8:30)

# 9. Closing activities (Saturday 1:30)

## 9.1 Confirm WG21 consensus to adopt proposals ("consent agenda", approved without discussion if no new information)

Clow reported that LWG had reviewed the P0218R0 proposal, making only one significant change, in the front matter which should have replaced `std::experimental::filesystem` with `std::filesystem` instead striking the entire paragraph. LWG recommend the adoption of the proposal.

Smith requested clarification whether the restored text was an instruction to the editor or to readers of the standard. It was clarified that it was the latter.

Proposals adopted based on consent agenda.

Meredith gave a presentation showing advantages of Unified Call Syntax for the standard library. The new `begin` and `end` functions were added in C++11, then in C++14 additional functions such as `cbegin` and `crend` were added, then corrected. The same functions are being modified again in C++17. With unified call syntax these would be unnecessary because the member functions would be found.

Sutter requested people to think about the feature before the Oulu meeting and talk about it. There might be an evening session in Oulu to discuss it again and see if there is better consensus.

Van Eerd requested information on both sides of the story. Meredith to write a paper containing that information.

Voutilainen pointed out that the new functions also add implementation burden for facilities that very few people use.

Van Eerd asked whether we had rewritten `make_pair` yet for this standard revision. Voutilainen said we'd changed the whole of `pair` and `tuple`.

Myers said there has been a suggestion of an open-access journal that publishes proposals accepted by the committee, which would be reviewed by an editorial board of academic members of the

committee. Sutter explained that a number of academics spend their time writing WG21 proposals which they cannot publish in peer-reviewed journals, sometimes at the expense of writing other papers which would help their career more. Allowing them to get credit would be great.

Yasskin reported that LEWG reviewed `joining_thread` and approved it for C++17. LEWG also looked at `variant` again and decided to try proposing it for C++17, based on a poll at the start of the Oulu meeting. Voutilainen asked the LWG chair whether a paper with new wording for the `joining_thread` would be considered by LWG in Oulu. Clow said they would be busy but he would try. Voutilainen recalled that this addresses NB comments on C++14.

Meredith reported that LEWG would like to propose a new namespace name reserved for future standardization, and requested comments.

Voutilainen asked the convener what the ballot status of Library Fundamentals v2 was. Sutter said it would close on Tuesday 8th March and asked the library chairs whether they planned to have ballot resolution telecons. Clow said he planned to have issue resolution telecons, not necessarily ballot resolution telecons.

### 9.2 PL22.16 motions, if any

None.

### 9.3 Issues delayed until today

None.

## 10. Plans for the future

### 10.1 Next and following meetings

2016-06-20/25 Oulu, FI (N4570) 2016-11-07/12 Issaquah, WA, US (N4571) 2017-02-27/03-04 Kona, HI, US (N4573)

Likely to be a meeting in Toronto, July 10-15th 2017.

Yasskin reported an offer for a Library meeting in Chicago, in August or September and asked for a show of hands to gauge interest.

### 10.2 Mailings

Spicer announced some changes to the mailing process. Requested that documents clearly state the audience, and an abstract for non-trivial papers, and a revision history.

The post-meeting mailing deadlines will be the Monday three weeks after the meeting at 1400 UTC. The pre-meeting mailing deadlines will be the third Monday before the meeting, to give people more time to review papers in the mailing.

For Oulu the deadline will be May 30th.

Liber thanked Spicer for moving the pre-meeting mailing earlier.

Halpern asked what the audience should be for a revision of a paper that has moved from one group to another. The answer was to put the next group, or groups, to look at it.

# 11. Adjournment

Hedquist moved to adjourn, Clow seconded. Passed by acclamation.

# 12. Attendance

The column “WG21” designates official PL22.16 or WG21 status (“P”, “A”, “E”, “M”)

The column “PL22.16” indicates organizations eligible to vote by “V”.

### PL22.16 members

| Company / Organization | NB | Representative | WG21 | PL22.16 |
|---|---|---|---|---|
| AMD | | Ben Sander | A | V |
| AMD | | Tony Tye | A | |
| Apple | | Duncan Exon Smith | | |
| Argonne National Lab | | Hal Finkel | P | V |
| Bloomberg | | John Lakos | P | V |
| Bloomberg | UK | Alisdair Meredith | A | |
| Bloomberg | UK | Dietmar Kühl | A | |
| Bloomberg | UK | Mathias Gaunard | A | |
| Bloomberg | | Nathan Myers | | |
| Bloomberg | | Graham Bleanex | | |
| Brown | | Walter E. Brown | E | |
| CERT Coordination Center | | Aaron Ballman | P | V |
| Cisco Systems | | Lars Gullik Bjønnes | P | V |
| Dinkumware | | P.J. Plauger | P | V |
| Dinkumware | | Tana Plauger | A | |
| Edison Design Group | | John H. Spicer | P | V |
| Edison Design Group | | Daveed Vandevoorde | A | |
| Edison Design Group | | Jens Maurer | A | |
| Edison Design Group | | Mike Herrick | A | |
| Edison Design Group | | William M. Miller | A | |
| FlightSafety International | | Billy Baker | P | V |
| Google | | Chandler Carruth | A | V |

| Company / Organization | NB | Representative | WG21 | PL22.16 |
|---|---|---|---|---|
| Google | | Geoffrey Romer | A | |
| Google | | Hans Boehm | A | |
| Google | | James Dennett | A | |
| Google | | Jeffrey Yasskin | A | |
| Google | CA | JF Bastien | A | |
| Google | UK | Richard Smith | A | |
| Google | | Thomas Koeppe | A | |
| Google | | Titus Winters | A | |
| Google | | Greg Miller | A | |
| GreenWireSoft | | Juan Alday | P | V |
| IBM | CA | Michael Wong | P | V |
| IB | | Paul E. McKenney | A | |
| IBM | CA | Hubert Tong | | |
| IBM | | Maged Michael | | |
| Indiana University | | Marcin Zalewski | A | V |
| Intel | | Clark Nelson | P | V |
| Intel | | Pablo Halpern | A | |
| KCG Holdings | | Robert Douglas | P | V |
| Lawrence Berkeley | | Bryce Adelstein-Lelback | P | V |
| Lawrence Livermore | | James Frederick Reus | P | V |
| Los Alamos National Laboratory | | Li-Ta Lo | P | V |
| Los Alamos National Laboratory | | Christopher Sewel | A | |
| Los Alamos National Laboratory | | S. Davis Herring | A | |
| Louisiana State University | | Hartmut Kaiser | P | V |
| Microsoft | | Jonathan Caves | P | V |
| Microsoft | | Gabriel Dos Reis | A | |
| Microsoft | | Herb Sutter | A | |
| Microsoft | | Stephan T. Lavavej | A | |
| Microsoft | | Gor Nishanov | | |
| Microsoft | | Andrew Pardoe | | |
| Microsoft | | Neil Mackintosh | | |
| Microsoft | | Casey Carter | | |
| Morgan Stanley | | Bjarne Stroustrup | P | V |
| NVidia | | Jared Hoberock | A | V |
| NVidia | | Michael Garland | A | |
| NVidia | | Olivier Giroux | A | |
| NVidia | | Boris Fomitchev | A | |
| NVidia | | Sergie Nikolaev | A | |
| Oracle | | Paolo Carlini | P | V |

| Company / Organization | NB | Representative | WG21 | PL22.16 |
|---|---|---|---|---|
| Oracle | | Fedor Sergeev | A | |
| Oracle | | Maxim Kartashev | A | |
| Perennial | | Barry Hedquist | P | V |
| Perennial | | Beman G. Dawes | A | |
| Perennial | | Lawrence Crowl | A | |
| Plum Hall | | Thomas Plum | P | V |
| Plum Hall | FI | Ville Voutilainen | A | |
| Programming Research Group | | Christof Meerwald | A | V |
| Qualcomm | | Marshall Clow | P | V |
| Red Hat | | Jason Merrill | P | V |
| Red Hat | UK | Jonathan Wakely | A | |
| Red Hat | | Torvald Riegel | A | |
| Ripple Labs | | Howard Hinnant | A | V |
| Sandia National Labs | | Carter Edwards | P | V |
| Seymour | | Bill Seymour | P | V |
| Sony Computer Entertainment | | Sunil Srivastava | P | V |
| Sony Computer Entertainment | | Michael Spencer | | |
| Stellar Science | | David Sankel | P | V |
| Symantec | | Mike Spertus | P | V |

## Other WG21 members

| Company / Organization | NB | Representative | WG21 |
|---|---|---|---|
| Mozilla | CA | Botond Ballo | M |
| Christie Digital | CA | Tony Van Eerd | M |
| CERN | CH | Axel Naumann | M |
| Vollmann Engineering | CH | Detlef Vollmann | M |
| | DE | Fabio Fracassi | |
| University Carlos III | ES | J. Daniel Garcia | M |
| CryptoTec | FI | Mikael Kilpeläinen | M |
| PDT Partners | UK | Jeff Snyder | M |
| | UK | Jonathan Coe | M |
| | UK | Roger Orr | M |

## Participating non-members

| Company / Organization | NB | Representative |
|---|---|---|
| Autodesk, Inc. | | Adam Helps |
| LTK Engineering | | Alan Talbot |
| University of Akron | | Andrew Sutton |

| Company / Organization | NB | Representative |
|---|---|---|
| | | Arash Partow |
| Blizzard Entertainment | | Brian Fitzgerald |
| Facebook | | Eric Niebler |
| | | Faisal Vali |
| | | Guy Sandberg |
| Blizzard | | James Touton |
| Facebook | | Lee Howes |
| Xalnix Corp. | | Les Potter |
| Facebook | | Louis Brandy |
| | | Matt Calabrese |
| Kitware | | Matthew Woehlke |
| Frankfurt Inst. for Adv. Studies | | Matthias Kretz |
| Bob Taco Industries | | Michael McLaughlin |
| Coverity by Synopsys | | Michael Price |
| | | Nicolai Josuttis |
| Rancher Labs | | Oleg Smolsky |
| SAS Institute | | Phil Ratzloff |
| Coverity by Synopsys | | Tom Honermann |
| University of Illinois | | Vincent Reverdy |
| Schonfeld | | Wesley Maness |
| ARM | | Will Deacon |
| Yander | | Yegor Devevenets |