WG14 N1763
INCITS PL22.11/13-002
Date: 2013-09-30
Reply To The Attention Of: Barry Hedquist
PL22.11 Secretary
Email: beh@peren.com

MINUTES
April 23-26, 2013
MEETING OF ISO/IEC JTC 1 SC 22/WG 14 AND INCITS PL22.11

Meeting Location:
Netherlands Standardization Institute
Vlinderweg 6
NL-2623 AX Delft, NL

Meeting information:
N1640

Local contact information:
Willem Wakker
ACE Consulting bv
De Ruyterkade 113, 1011 AB
Amsterdam, NL
Tel: +31 20 6646416
Mob: +31 625 026561
Fax: +31 20 6750389
E-Mail: willemw@ace.nl

Scheduled Meeting Times:
23 April 2013 09:00 – 12:00 13:30 – 16:00
24 April 2013 09:00 – 12:00 13:30 – 16:30
25 April 2013 09:00 – 12:00 13:30 – 16:30
26 April 2013 09:00 – 12:00

Teleconference information:

- Topic: WG 14 April 2013
  Date: Every day, from Tuesday, April 23, 2013 to Friday, April 26, 2013
  Time: 9:00 am, Europe Summer Time (Amsterdam, GMT+02:00)
  3:00 am, Eastern Daylight Time (New York, GMT-04:00)
  Meeting Number: *957 803 942*
  Meeting Password: *wg14*

  To join the online meeting (Now from mobile devices!)

- Go to ISO Meeting
- If requested, enter your name and email address.
- If a password is required, enter the meeting password: *wg14*
- Click *Join*.
- To view in other time zones or languages, please click the link: Other Time Zones and Languages

  ---

To join the audio conference only

To receive a call back, provide your phone number when you join the meeting, or call the number below and enter the access code.
Call-in toll-free number (UK): 0800-051-3810
Call-in toll number (UK): +44-203-478-5289
Global call-in numbers: Global Numbers
Toll-free dialing restrictions: Toll-Free
Access code: *957 803 942*

For assistance

- Go to ISO WebEx Help
- On the left navigation bar, click *Support*.
- Toll-free dialing restrictions: tollfree restrictions

Access code: *954 880 418*

For assistance:

1. go to iso meeting support
2. On the left navigation bar, click "Support".

To add this meeting to your calendar program (for example Microsoft Outlook), click this link: iso meeting to calendar

To check whether you have the appropriate players installed for UCF (Universal Communications Format) rich media files, go to iso meeting diagnostics

http://www.webex.com/

1. Opening Activities

1.1 Opening Comments (Wakker, Benito)

John Benito and Willem Wakker welcomed us to the Netherlands Standardization Institute in Delft, Netherlands and described the meeting facilities. Several local restaurants are within walking distance of the meeting. Lunch break will be from ***12:00-13:30.*** We are connected on WebEx to allow folks to call in. This meeting is hosted by the Netherlands Standardization Institute and ACE. Refreshments and coffee are available in the room. Lunch will be in the building.

1.2 Introduction of Participants/Roll Call

Name	***Organization***	***NB***	***Comments***
John Benito	**Blue Pilot**	**USA**	**WG14 Convener**
Jim Thomas	**Blue Pilot**	**USA**	
Martin Sebor	**Cisco**	**USA**	
David Keaton	**CERT/SEI/CMU**	**USA**	**PL22.11 Chair**
Daniel Plakosh	**CERT/SEI/CMU**	USA	
Robert Secord	**CERT/SEI/CMU**	**USA**	
Roger Scott	**Coverity**	**USA**	
Tana L. Plauger	**Dinkumware, Ltd**	**USA**	
P. J. Plauger	**Dinkumware, Ltd**	**USA**	

Blaine Garst	**Garst**	**USA**	
Rajan Bhakta	**IBM**	**Canada**	**HoD - Canada**
Clark Nelson	**Intel**	**USA**	
John Parks	**Intel**	**USA**	
Robert Geva	**Intel**	**USA**	
Clive Pygott	**LDRA**	**USA**	
Herb Sutter	**Microsoft**	**USA**	
Douglas Walls	**Oracle**	**USA**	**HoD – USA, PL22.11 IR**
Darryl Gove	**Oracle**	**USA**	
Barry Hedquist	**Perennial**	**USA**	**PL22.11 Secretary**
Tom Plum	**Plum Hall, Inc.**	**USA**	
Bill Seymour	**Seymour**	**USA**	
Fred Tydeman	**Tydeman Consulting**	**USA**	
Nick Stoughton	**Austin Group**	**USA**	**Austin Group Liaison**
Freek Wiedijk	**Radboud Univ. Nijmegen**	**Netherlands**	
Willem Wakker	**ACE**	**Netherlands**	**HoD - Netherlands**
Roberto Baghara	**Univ. of Parma**	**Italy**	**HoD - Italy**
Owen Shepherd	**Self**	**France**	

1.3 Procedures for this Meeting (Benito)

The Meeting Chair, John Benito, WG14 Convener, announced the procedures are as per normal. Everyone is encouraged to participate in the discussion and straw polls.

Straw polls are an informal mechanism used to determine if there is consensus within the meeting to pursue a particular technical approach or even drop a matter for lack of consensus. Participation by everyone is encouraged to allow for a discussion of diverse technical approaches. Straw polls are not formal votes, and do not in any way represent any National Body position. National Body positions are only established in accordance with the procedures established by each National Body.

INCITS PL22.11 members reviewed the INCITS Anti-Trust Guidelines at:

http://www.incits.org/inatrust.htm.

All 'N' document numbers in these minutes refer to JTC1 SC22/WG14 documents unless otherwise noted.

Emphasis for this meeting is to consider defect reports and future Technical Specifications for WG14.

Barry Hedquist, PL22.11 Secretary, is the Recording Secretary for the meeting.

1.4 Approval of Previous Minutes (Hedquist) (N 1643).

Several comments for typos, etc.

Minutes were modified per editorial changes and approved.

Final Portland Minutes are **N1692**

1.5 Review of Action Items and Resolutions (Hedquist)

ACTION: Clark Nelson to follow the disposition of N1634 in WG21 (N3433) and report such back to WG14.

ACTION: Robert Secord to make the agreed to changes to N1624 for the next mailing.

DONE

ACTION: A small editing group review the edits made to N1624.
The editing group will consist of:

John Benito
Blaine Garst
David Keaton
Dave Prosser
Clive Pygott
Robert Secord
Willem Wakker
Douglas Walls

DONE

ACTION: Convener to either send the reviewed document to SC22 for PDTS Ballot, or not, as appropriate.
Sent - DONE

ACTION: Clark Nelson to examine the adoption of the Proposed TC words for DR 402, and revise as needed.
DONE

ACTION: Convener to communicate discussion of submitted Defect Reports to the submitters.
DONE

1.6 Approval of Agenda (N 1688).

Revisions to Agenda: Last Updated 4/22/13

Added Items: None

Deleted Items: None

Agenda approved by unanimous consent.

1.7 Identify National Body Delegations
US, Canada, Netherlands, Italy,

1.8 Identify PL22.11 voting members

See PL22.11 TAG Minutes, following these minutes. 14 of 17 members present.

2. Reports on Liaison Activities

2.1 SC 22 (Benito, Plum)
No report.

2.2 PL22.11/WG 14 (Benito, Keaton, Walls)

Will retake the vote for the US Delegation. Secure coding rules completed ballot. FP Study Group now has a project number.

2.3 PL22.16/WG 21 (Plum)

WG21 met last week in Bristol, UK.

SC 22 conducted a ballot for the establishment of a new work item for WG21, Technical Specification, File System. And additional NP will be processed for an Amendment to C++, aka C++2014. This Amendment will consist of Technical Corrigenda, and several small new features C++. A larger set of new features is envisioned for a revision to the C++ Standard in 2017.

Further discussion lead to a conclusion that WG14 participants that normally attend WG21 meetings should be assigned specific areas to monitor for C/C++ compatibility issues.

Tom Plum, Plum Hall (US) - Core, Evolution, Modules (SG 2), Numerics (SG 6) and Transactional Memory (SG 5).

Bill and Tana Plauger, Dinkumware (US), Library, File System (SG 3).

Clark Nelson, Intel (US), Core, Concurrency.

Bill Seymour, self (US), Library, Database, Numerics (SG 6)

Barry Hedquist, Perennial (US), Evolution, Networking (SG 4)
Hedquist reported that SG 4, Networking, is planning to retain compatibility with several POSIX macros used for byte order conversion: htonl, htons, ntohl, and ntohs, and provided the WG21/SG 4 Chair with a contact point with The Austin Group – Nick Stoughton.

Sutter provided an overview of the WG21 activities last week. NWI's are being processed as documents are ready for forwarding to SC22.

Clark reported that Crowl's paper on memory allocation was accepted last week. There are about a half dozen items.
Signal handlers – in C we simple set a flag or exit. C++ wants to do more than that.
A subset of VLAs was accepted, compatible with C.
C++ accepted binary literals – do we have an interest?
Digit seperators were rejected.
Qualification conversions on pointers. To consider this, we need a paper, which is the same approach for all terms for us to consider.
WG14 interest in Cilk was not warmly greeted by WG21.

Sutter asked if C had a compatibility annex in the C Standard. We do not, and would probably need a LOT of help from C++ folks to do so.

Seymour reported that we may expect to see a paper on an exact width floating point types (N1703). Make sure that a copy gets sent to Jim Thomas.

Herb in interested in seeing a C/C++ compatibility annex added to the C Standard. The existing Annex in the C++ Standard could serve as a starting point. Tom does not see developing such an annex as a good use of people's time. The existing annex in C++ is not up to the technical quality of the standard. The annex in C++11 is not technically correct, and maybe should be deleted. Herb believes the information has value in the marketplace, and many find it useful. Clark pointed that people in the world are not transitioning from C++ to C, and that an annex usually has meaning only in that context. Rajan believes only one document should exist. David sees a continual switching back and forth between C and C++, and is something we really ought to have, that explains the overlap between the languages. One approach may be to develop a single separate document, perhaps as a TS, using experts from both Committees. Herb agrees that participation from both groups is needed. What do we want to do? Tom agreed to take an action to review the existing annex in C++ for technical accuracy. Tom believes that are compiler implementations that support named typedefs that are usable from either C or C++, although he is not aware that is true for atomics. Herb asked if mutexes are compatible. PJ pointed out they are of different types, but do the same things. They are not compatible, and for good reasons. Threads are another area of compatibility issues. David would like us to give greater thought about compatibility issues prior to making decisions about adopting new features. Benito pointed out that whatever vehicle is used will not work unless somebody owns it. Herb thinks he'll have volunteers in WG21. Tom emphasized that he

considers such an effort as a ISO document is a waste of effort, and will oppose it. The bureaucratic overhead alone makes it not worth the effort. But keeping such a document updated on a web page, such as on isocpp.org, would be easier to manage.

2.4 PL22 (Plum)

PL22 meets twice a year via teleconference. Next teleconference will be in June 2013. See Tom Plum, PL22 Chair, for details.

2.5 WG 23 (Benito)

TR 24772 is published soon with bindings to several languages, Meets in conjunction with SC22 plenary, and others. Looking for an Annex for C++. Also working on a Code Signing TS. Berlin in June.

2.6 MISRA C, N1697 (Montgomery)

Publication of MISRA C:2012 was announced on Feb 26, 2013, based on C99. The PDF version of the document was duly published, as announced, on 18 March 2013 and is available for purchase from http://www.misra.org.uk/shop/buy_now.php. The printed version can also be ordered from the same site but will not be available until week commencing 8 April.

2.7 Floating-point SG (N1681, N1689) (Thomas)

The Study Group continues to work on a C binding for IEEE 754-2008 / IEC 60559:2011, to be delivered as a five-part Technical Specification. Teleconferences were held in February and March. Two documents have been submitted, N1681, and N1689, containing notes from those teleconferences. A NWI has been assigned, and the Study Group is officially disbanded. The work is now part of WG14.

For further information on the group, contact Jim Thomas.

2.8 Other Liaison Reports

POSIX – N1694 (Stoughton)

Stoughton submitted a report from the Austin Group regarding LC_CTYPE, asking if the interpretation by The Austin Group regarding the use of LC_CTYPE is consistent with the C Standard. Nick will call in later today to discuss this item.

Does the C Locale support multibyte encodings? There is nothing in the C Standard that says it does, but there are a number of places where the implication is that multibyte characters are not supported. The C Standard lists that which is required, but does not prohibit a C locale from supporting additional characters. The behavior is whatever is appropriate for that case. 5.2.1;p3 is the source of concern. toupper and tolower apply only the defined character set. The Committee agreed that the interpretation of the Austin Group as presented by Nick was a correct interpretation.

3. Teleconference Meeting Reports

3.1 Report on any teleconference meetings held (Benito)

A teleconference was held 17-18 December, 2012, to discuss PDTR 17961. Went through the editorial issues, sent to SC22, and a PDTS ballot was held. One NO vote.

4. Future Meetings & Mailings

4.1 Future Meeting Schedule

- Fall 2013: Chicago, IL, USA, Sept 28 - Oct 3, 2013, with WG21.
- Spring 2014: Parma, Italy
- Fall 2014: St. Louis – Dates: 3-8 Nov C++, C last week of October? Bill Seymour will follow up.
- Spring 2015: Host needed.

4.2 Future Mailings

- Post Delft: 27-May-2013
- Pre Chicago: 02-Sept-2013

5. Document Review

Several of the documents listed here are proposed Defect Reports (DR). Documents that were accepted as Defect Reports were give a DR number. Further discussion of the document as a Defect Report can be found in Section 6, Defect Reports.

5.1 N1670 Proposed Defect Report: Appendix G.5.1: -yv and -x/v are ambiguous (Tydeman)

Discussion: Some of the formulas in the referenced tables seem ambiguous, should we disambiguate them? Clark says they are not ambiguous. The rules are known. If we make it a DR, we can answer it with a formal response.

Accepted as a Defect Report? YES

See DR 426 in Section 6 for further discussion.

5.2 N1671 Proposed Defect Report: Function Parameter and Return Value Assignments (Miller)

Discussion:

Accepted as a Defect Report? YES

See DR 427 in Section 6 for further discussion.

5.3 N1672 Proposed Defect Report: Runtime-constraint issue with sprintf family in Annex K (Walls)

Discussion:

Accepted as a Defect Report? YES

See DR 428 in Section 6 for further discussion.

5.4 N1673 Proposed Defect Report: Should get_s discard next input line when (s==NULL)? (Walls)

Discussion:

Accepted as a Defect Report? YES

See DR 429 in Section 6 for further discussion.

5.5 N1674 Proposed Defect Report: getenv_s maxsize should be allowed to be zero (Walls)

Discussion:

Accepted as a Defect Report? YES

See DR 430 in Section 6 for further discussion.

5.6 N1675 Proposed Defect Report: atomic_compare_exchange (Walls)

Discussion:

Accepted as a Defect Report? YES

See DR 431 in Section 6 for further discussion.

5.7 N1677 Proposed Defect Report: Is 0.0 a representable value? (Tydeman)

Discussion:

Accepted as a Defect Report? YES

See DR 432 in Section 6 for further discussion.

5.8 N1683 Proposed Defect Report: Constraint issues for wide character functions using RSIZE_MAX (Walls)

Discussion:

Accepted as a Defect Report? YES

See DR 433 in Section 6 for further discussion.

5.9 N1687 Proposed Technical Corrigenda: Thread Local Storage TSS Specification (Shepherd)

This paper contains Proposed Technical Corrigenda to for DR 416, and DR 424. However, those DRs may be determined as NOT being defects. Thus, N1687 may be treated as a proposal to modify the C Standard as an extension (TS), or as a future revision (ISO/IEC 9899:20XX). TBD

Introduction

The final release of the C11 international standard leaves many aspects regarding thread local storage unspecified. Specifically, the following aspects are unspecified:

- If or when destructors for thread specific storage ("tss") objects are invoked
- The ordering (or lack thereof) of destructor invocation
- The identity of the thread invoking the TSS destructors
- The number of times TSS destructors may be invoked (and the meaning of the TSS_DTOR_ITERATIONS constant)
- The behavior of TSS destructor invocation in the face of parallel modifications

This ambiguity leaves the utility of the TSS feature in a fully conforming application (i.e. one not relying on additional assertions by the implementation) greatly reduced. In particular, it prevents the usage of the thread specific storage feature for reliable resource cleanup

This proposal is submitted in relation to specification defect reports DR 416 (by the same author as this proposal) and DR 424. This proposal will look at existing implementations of thread specific storage and their behavior, and will then propose alterations to the C11 standard.

Delft Discussion: Douglas prefers now to treat this as a DR. JB disagrees. The paper is not in the correct from to handle as a DR, and we are not going to edit the paper. We can discuss this in the DR section, but not give it a DR number.

5.10 N1676 DTS 18661: Floating Point extensions for C - Part 1: Binary floating-point arithmetic (Thomas)

N1676 is an updated version of PDTS 18661, Part 1, and was presented by Jim Thomas.

Notes for N1676 presentation**: Look at N1702, the spreadsheet of comments.**

5.11 N1690 PDTS 18661: Floating Point Extensions for C - Part 2: Decimal floating-point arithmetic (Thomas)

N1690 is an updated version of N1657, a prior draft of a Technical Specification for Decimal Floating Point – Part 2, Decimal Floating-point arithmetic.

The goal is to prepare a document for WG14 review between now and the next meeting. Change bars indicate changes from the prior doc. There are about ten significant changes. Jim walked through each of them, and there were no significant comments.

5.12 N1691 PDTS 18661: Floating Point Extensions for C - Part 3: Interchange and Extended Types (Thomas)

N1691 is the Committee's first look at Part 3, which contains extensions to the C Standard as recommended by ISO/IEC/IEEE 60559:2011.

Jim Thomas walked through Part 3, PDTS 18861. The scheduling plan is to move along one meeting behind Part 2. Does C++ know we are working on this. Not yet. Bill Seymour will communicate this to WG21/SG6, Numerics.

Blain Garth wants a better understanding of the term 'interchange', as it is used in this document. It's that same meaning as in IEEE 60559. Do we need an explanation in a C Binding? The IEEE use of the term really refers to that which is nailed down, well defined, and thus 'interchangable'.

ACTION: Bill Seymour will communicate our work on PDTS 18661, Part 3, to WG21/SG6, Numerics.

5.13 N1665 Intel Cilk Plus Language Extension Specification (Nelson)

Note: Intel is a registered trademark of Intel Corporation. Cilk is a trademark of Intel Corporation.

Portland:
Clark led a discussion on this paper. Should allocations be clustered or divided up? Let implementations be smart about the way they do dynamic allocations.
Douglas: Oracle only likes the proposed change to 7.22.3, paragraph 2. They oppose the proposed change to 7.22.3;p1.
David K: Disagrees. The proposed change changes the meaning of what is intended.
Defer any action on this paper. We see no action to take at this time, and will wait until WG21 (C++), takes action.
ACTION: Clark to follow the disposition of N1634 in WG21 (N3433) and report such back to WG14.

Question: Does this paper stand alone, or are the other papers needed in order to implement a compiler that will support the features contained therein? Other papers cited are:

Part 1. The Intel® Cilk™ Plus Language Specification, document number 324396-002US.
Part 2. The Intel® Cilk™ Plus Application Binary Interface, document number 324512-001US.

5.14 N1682 Leveraging OpenMP for language level parallelization (Gove)

From the Introduction:

This proposal suggests how language level parallelisation can be achieved using the existing OpenMP infrastructure. OpenMP is a mature, well established and widely used specificationfor writing portable multi-threaded applications on a shared memory system. All leadingcompilers (Oracle, Intel, IBM, gcc, Microsoft, PGI, etc.) already support OpenMP.

OpenMP has been around for over 15 years, the first formal version was published in 1997, the 4.0 version of the specifications is due to be released in 2013. The 4.0 specification will also include support for cc-NUMA systems and heterogeneous hardware architectures, like DSP and GPU accelerators.

OpenMP uses compiler directives to express the parallelism. However, the implementation of an OpenMP program relies on a library interface that can programmatically query the run-time system, alter settings and control how the parallelisation is undertaken.

This paper contains a proposal to leverage existing OpenMP infrastructure as part of a future language standard for parallelisation. In this proposal we have assumed some familiarity with OpenMP. A reader who is not familiar with OpenMP would be advised to read the brief overview contained in the appendix before reading the proposal.

Delft Discussion:

Darryl Gove from Oracle presented a slide presentation on OpenMP. One of the issues not addressed by OpenMP is one of resource management. There's a difference between using data parallelism for a given program, and task parallelism with multiple tasks running on the same platform. These typically require o/s level solutions, and are not dealt with at the program level, but eventually both problems need to be solved.

There may be common ground between OpenMP and the material presented in an earlier presented on Cilk that could help resolve the issues of tasking and data parallelism, and that a future next step may be a joint proposal based on a combination of OpenMP and Cilk Plus. Should we consider setting up a Study Group to start looking at options and possible solutions? We'll talk more about this later this week.

Where do we go from here with this material and with Cilk? Do we know what we can try to nail down in the near term, and leave for future resolution? Do we really think that whatever we try to nail down, if anything, will never change? Array sectioning may be a good initial candidate. Probably want to stay away from run-time dependencies. Cilk specifications are published, and available. Members want to take some time thinking about what to do next, but the idea of getting Cilk and OpenMP to work together looks promising. We can look at a refresh summary of Cilk tomorrow.

Robert Geva of Intel gave an overall review of previously presented material on Cilk Plus. He reviewed two keywords, cilk_spawn (_Cilk_spawn) and cilk_sync (_Cilk_sync), which are applied only to function calls. His presentation addressed specific comments received after the last presentation, and answered a number of questions raised by participants. Implementations of Cilk Plus include those by Intel and GCC.

Where do we want to go from here? We've listened to two approaches, OpenMP and Cilk. Wakker would like to see a unified proposal from OpenMP/Cilk that could serve as a basis for setting up a Study Group. Tom would like to see a SG created now so that a combined proposal can be worked out, along with input in several specific areas from WG14. Clark believes the harmonization between the two will have a very minor impact on existing proposals. However, taking what we already have and making it ready for WG14 will be the larger effort. Herb agrees with Clark, and has seen that process work well in WG21. Microsoft would also want to participate in creating a merged proposal. Blaine asked "How do we scope the work?" If a lot of specialized knowledge is needed to make use of parallelism, what can we really offer? Benito explained that our approach would be to create a Technical Specification prior to any attempt at standardization. The approaches by OpenMP and Cilk are not seen to be very far apart. In order to create a SG, we need someone who will volunteer to run it, and we need a Project Editor.

One approach:

- combine OpenMP/Cilk specs
- integrated with existing C11 (threads, atomics, etc)
- include a rationale – why we chose to go this way rather than another way

David Keaton suggested using a charter to 'study the problem of parallelism in C' will give the SG the flexibility to shift gears if needed. David also pointed out that a SG will provide the legal protection needed to allow competitors to discuss combining approaches without being accused of collusion. Clark Nelson is willing to Chair the SG. DONE.

5.15 N1669 PDTS 17961, C Secure Coding Rules (Secord) N1700, N1701

N1669 is an updated version of N1624.

Portland: N1636 is a set of comments on N1624, Draft TS 17961, C Secure Coding Rules, submitted by Douglas Walls. Most of the changes submitted were made to the draft.

Dave Prosser is concerned that the use of the term 'string' does not match the definition of string in the C Standard. Blaine Garth agrees. Dave Keaton will review the use of 'string'.

ACTION: Robert Secord to make the agreed to changes to N1624 for the next mailing.

ACTION: A small editing group review the edits made to N1624.
The editing group will consist of:

John Benito
Blaine Garth
David Keaton
Dave Prosser
Clive Pygott
Robert Secord
Willem Wakker
Douglas Walls

ACTION: Convener to either send the reviewed document to SC22 for PDTS Ballot, or not, as appropriate.

N1701 – Comments from Japan

5.16 N1695 An enhanced format output design for printf

N1637 is a paper that states the C Standard does not allow Turing complete implementations; evaluation semantics do not preserve typing, and that no strictly conforming programs exist (i.e. the C Standard cannot guarantee a C program will not crash.)

Has anybody studied this paper to a point where one is willing to champion the effort needed? Adoption would take a Technical Specification or a revision to the C Standard. The paper is complex, not easily understood, and we don't know what is really needed. We can't do anything with this paper as written. The author has been non-responsive to requests for clarification from the Convener. No further action is planned at this time.

5.17 **N1699** Results of PDTS 17961 Ballot

The results of the SC22 Ballot for PDTS 17961, one NO vote from The Netherlands.

5.18 **N1700** NEN Comments on PDTS 17961 Ballot

We do not have a document with proposed responses to review. Responses are being generated in real-time. However, the Netherlands seems satisfied with that process.

Willem – maybe this should be a TR type 3 – Information only, no conformance requirements, leave it to the market place. Rajan points out there are some decidable rules.

In preparation of the discussion this afternoon the Netherlands proposes the following words to be included in the conformance section:

> A conforming implementation should fully implement all decidable rules from this Technical Specification. For each of the undecidable rules, it should be documented what the quality achieved is through a set of representative examples for which the implementation can correctly establish violations of the rule, plus a set of representative examples of false positives (if they can occur) and false negatives (if they can occur).

For this to make sense, of course each rule in the document should be marked as "decidable" or "undecidable".
The Netherlands is willing to provide the list of "decidable" and "undecidable" rule; it is left to the editor to decide how to implement this change in the document.

If WG14 decides not to go this way, the only alternative we see is to change the designation of the specification from a Technical Specification to a Technical Report (type 3), the latter being an official ISO document intended to give information without conformity requirements.

It also seems that removing the conformance section resolves the issue.

MISRA 2012 rules are all tagged as either decidable or undecidable, and suggests the TS should do likewise.

Examples exist for every rule, however the examples are the simplest possible, and do not really reflect more realistic complexity.

The key concern here is that it will be very difficult to distinguish between analyzers in terms of 'conformance' because the conformance rules are generally weak.

We were not able to generate words that satisfied concerns of The Netherlands, but did generate new words in a number of places that others were satisfied with. Willem expressed appreciation of the Committee's effort to resolve the objection from The Netherlands.

John points out that the real value of an analyzer is whether or not it can tell you the problems it found are, in fact, problems you need to fix. No one expects an analyzer to find all problems, and those that have a high level of 'noise', i.e. problems detected, are almost worthless.

5.19 **N1701** JISC comments on PDTS 17961 Ballot

Reviewed comments to PDTS 17961 submitted by Japan. Each of the comments contains a proposed response generated by Robert Secord. Seventeen comments were submitted, one of which contained a proposed response from the Japanese.

JP2 – The document does have the inconsistency cited in this comment. The first sentence "All rules are meant to be statically enforceable" is actually incorrect. In a number of cases, choices were made to make the rule dynamically enforced or statically enforced depending on the nature of the rule. This is a fundamental problem with the specification, and will likely require a large effort to clarify whether static or dynamic checking is applicable. Bottom line however is that an analyzer is not expected to do dynamic analysis. Then, why would such a rule exist?

JP3 – Adds a new sentence.

JP6 – Douglas argues that this is NOT an editorial change, and does not want to see a change made.

JP13 – The response completely misses the point that the comment is referring to a 'pair of arguments', and malloc has only one argument.

5.20 **N1702** Comments and study group suggested responses to N1676

Jim led a discussion on the internal comments submitted. They were reviewed during the April teleconference. Most were accepted. Focus for this discussion is the non-editorial comments. Do we agree with the proposed changes? Yes. Next

step is to prepare a new draft incorporating these changes. Form a small editorial committee to check the changes to N1676. Do we want to send it to PDTS ballot after that? Yes. David Keaton, Fred Tydeman, John Benito agreed to serve on the committee.

ACTION: Small editorial committee to review the changes to N1676 made by the accepted comments contained in N1702.

ACTION: Convener to submit the revised version of N1676 to SC22 for PDTS Ballot. (90 day Ballot)

5.21 **N1704** New Complex Mathematical Functions for the C Language. (Reverdy)

This paper proposes that complex functions not yet included in C be included at some point in the future. Today, users are forced to switch to another computer language in order to use complex special functions. Particularly those defined in 60559-2011, functions cerf, cerfc, clgamma, and ctgamma.

Where do we want to go with this paper? Are they suitable for adding into our existing effort? We are not revising the C Standard and this work does not really fit into the current Study Group on Floating Point. Right now, we do not have a vehicle in place to carry out the work described in this paper.

6. Defect Reports

6.1 DRs in Review Status

The following DRs in REVIEW Status were either moved to Closed, or left in Review, as noted.

DR 400 - REVIEW

Proposed Technical Corrigenda from Feb, 2012, is adopted unchanged. Moved to CLOSED.

DR 402 – REVIEW
ACTION - Rajan and Blaine to write up words/diagrams for DR 402. There's a typo as well, p12 s/be para 22.

Portland:
We reviewed a diagram by Rajan. Clark was responsible for translating the math into English.

If the value of B is not a value provided by X, it must be possible to find a value for Y, and B gets its value from Y. Rajan sees an interpretation of the existing words where an implementer could use Y to provide a value for B, when in fact it should have come from X. Blaine believes the existing proposed words in the DR better represent, in industry standard terms, the intent than the current words in the Standard. Clark and Rajan seem to agree the DR interpretation was never intended.

Clark suggests that we take the Suggested TC in the DR, move the DR to Review, and spend some time between now and the next meeting working with C++ to find a way to further clarify them if we are able.

ACTION - Clark to examine the adoption of the Proposed TC words for DR 402, and tweak as needed.

Delft Discussion: Was this discussed by the C++ Committee last week in Bristol?
above 3rd note, in 3rd paragraph "if a value…" should read 'operation B that modifies M'

Committee Discussion, April 2013. Additional words are needed to address Rajan's issues noted above. Theere are documents on the Wiki.

Leave in REVIEW

DR 408 - REVIEW

The Proposed Committee Response from Portland, Oct 2012, is adopted unchanged. Move to CLOSED.

DR 410 - REVIEW

The Proposed Technical Corrigenda from Feb 2012 is adopted unchanged. Move to CLOSED.

DR 412 - REVIEW

The Proposed Technical Corrigenda from Feb 2012 is adopted unchanged. Move to CLOSED.

DR 414 - REVIEW

The Proposed Technical Corrigenda from Feb 2012 is adopted unchanged. Move to CLOSED.
Copy-paste edit of Suggested TC to Proposed TC.

DR 415 - REVIEW

The Proposed Technical Corrigenda from Oct 2012 is adopted unchanged. Move to CLOSED.
Copy-paste edit of Suggested TC to Proposed TC.

DR 417 - REVIEW

The proposed Suggested Technical Corrigenda from Feb 2012 is adopted unchanged. Move to CLOSED.
Copy-paste edit of Suggested TC to Proposed TC.

DR 418 - REVIEW

The Proposed Committee Response from Oct 2012 is adopted unchanged. Move to CLOSED.
Change "..but none is reported, and removing it.." in Committee Discussion to "..but removing it.."

DR 420 - REVIEW

The Proposed Committee Response from Oct 2012 is adopted unchanged. Move to CLOSED

6.2 DRs in Open Status

The following DRs in **OPEN** status were either moved to **REVIEW**, or left in **OPEN**, as indicated.

DR 405 – OPEN

The DR was discussed in Oct 2012, however, consensus for full wording for a Proposed Technical Corrigenda was not reached. We proposed to add words as indicated in the Committee Discussion for Oct 2012.

Add the following as 7.26.4 p1 and p2:

> For purposes of determining the existence of a data race, lock and unlock operations behave as atomic operations. All lock and unlock operations on a particular mutex occur in some particular total order.
>
> NOTE: This total order can be viewed as the modification order of the mutex.

Delft Discussion: Move discussion from Feb 2012 to proposed TC, move to REVIEW.

DR 406 - OPEN

Portland: This item has also become WG21 Core issue 1466. This issue has not yet been discussed (pre-Bristol) by WG21.

Delft Discussion: CWG did not get to this in Bristol. Leave OPEN.

DR 407 – OPEN

Portland Committee Discussion
This item has also become WG21 Library Issue 2130. Rajan believes we can do the first part of this DR as proposed for 29.3p7. If the first part is accepted, the second part may be redundant. Prosser disagrees, and is not comfortable with dropping the second part.

ACTION: Rajan to write up with words/diagram for DR 407.

Proposed Resolution:

After 7.17.3 paragraph 11 add the following:

For atomic operations A and B on an atomic object M, if there are memory_order_seq_cst fences X and Y such that A is sequenced before X, Y is sequenced before B, and X precedes Y in S, then B occurs later than A in the modification order of M.

Clark – This is a Library Issue for C++. We might want to liaison with C++ on this. Do we want the corollaries in the C Standard? Once we decide that, we should communicate that to C++.

The first part of the DR seems to be understood. Wait for C++ to make a decision on the issues they have already opened. Leave OPEN.

Straw Poll: (13-1-0) - YES
Adopt the words in the Proposed Resolution given above for the first part.

Straw Poll: (3-10-3) - NO
Adopt the following derivatives following 7.17.3

Note that the following derivations do fall through the statement above, and as such, do not need to be stated explicitly:
For atomic modifications A and B of an atomic object M, if there is a memory_order_seq_cst fence X such that A is sequenced before X, and X precedes B in S, then B occurs later than A in the modification order of M.
For atomic modifications A and B of an atomic object M, if there is a memory_order_seq_cst fence Y such that Y is sequenced before B, and A precedes Y in S, then B occurs later than A in the modification order of M.

Delft Discussion: The Library Working Group of WG21 has developed a Proposed Resolution for this issue (2130), which was in REVIEW status going into the meeting last week. Was any action taken by WG21 in Bristol last week?

Move to REVIEW

DR 409 - OPEN

DR 409 has a Proposed Committee Response added in Oct 2012. Unless there is a need for further discussion, this item should be moved to REVIEW. Moved to REVIEW.

DR 413 - OPEN

Feb 2012 Meeting

Committee Discussion

- *It was noted that this is basically the same issue as dr 253.*
- *The following was proposed, but there was no consensus for adoption.*

The initialization shall occur in initializer list order, each initializer provided for a particular subobject overriding any previously listed initializer for the same subobject [151]. Subsequently, all subobjects that are not initialized explicitly previously shall be initialized implicitly the same as objects that have static storage duration.

Oct 2012 Meeting

Committee Discussion

Blaine believes the Standard is clear as written. Does anyone else read it differently? Clark believes there is some ambiguity. The intent for the value given in the example is 42. Rajan expected 0. It seems that we did not nail down words in the beginning to make it clear. Does anybody depend on '0' – not likely? This is really an edge case. If we make a clarification, it's not likely we will get anybody upset. Adding an example would be non-normative. It's also likely that the proposed new text can also be misread. GCC also gives '0'.

ACTION: David Keaton to draft words for DR 413.
DONE – N1659

From David's example, what does 'maximally-enclosing' mean? It's not a defined term in the Standard?

Make the 'new footnote' part of the normative proposed text in place of the 'maximally-enclosing'.

ACTION – Dave K to update N1659 (not yet published). - DONE

Dave Prosser does not see how the new proposed text clarifies the initial question asked – which sub-object is being initialized? Back to the kitchen.

Updated N1659:

Committee Discussion-Delft

6.7.9 paragraphs 17-18 specify that each designator list affects only the smallest subobject to which the designator list refers. As a result, the second clause of paragraph 19 occurs once for the greater object as a whole, filling in only those parts of the whole object that were never initialized explicitly.

There was ongoing discussion of the Proposed TC, which will result in more changes. Will not solve today, table for now.

Leave OPEN.

Delft Discussion: Willem says there are two conflicting directives in the Standard**.** At least seven compilers give the result as 0, six from IBM, and GCC. Rajan pointed out there would be some reluctance in making a change. However, those who are doing it 'right' need to be able to answer those who claim the 'right' implementations are 'wrong'.

ACTION – David Keaton to write a Proposed TC for DR 413.

DR 416 - OPEN

Oct 2012 Discussion:

This paper was discussed in conjunction with DR424 which covers one additional related issue.

The key observation is that these papers are about the deliberate underspecification of threads, since that allows the greatest opportunity for implementation on a variety of operating systems.

Pete Becker, an implementer of the C11 threads library, was asked about these papers and replied in SC22/WG14 message 12813.

Based on that response the committee is concerned that there could be subtleties in adopting the Proposed Technical Corrigendum and that, as such, these changes are substantial enough to warrant a proposal and to not be considered a defect report.

Delft: This topic along with DR 424 is now treated as a proposal, and moved there for discussion. See 5.9. This DR 416, and DR 424 should have a Proposed Committee Response, and moved to REVIEW???? Douglas now believes the original thinking behind this as a DR was that the proposed changes constituted sufficient changes to make it a proposal. However, he now believes we really underspecified threads to the point where it is unusable, and these changes are needs simply to make threads work, and should thus be treated as a DR. Tom pointed out that not posting this as a DR will deny others the opportunity to look at what we are doing. The Committee is considering whether or not the capabilities requested constitute items for future consideration as a revision to the Standard or to handle as a Defect. Link to the new paper.

We eventually need to decide which way to go with this: DR or Proposal. Leave OPEN

DR 419 - OPEN

Committee Discussion

Suggested Technical Corrigendum

7.17.1 add a new paragraph after paragraph 5:

> It is unspecified whether any generic function declared in stdatomic.h is a macro or an identifier declared with external linkage. If a macro definition is suppressed in order to access an actual function, or a program defines an external identifier with the name of a generic function, the behavior is undefined.

J.2 add:

> The macro definition of a generic function is suppressed in order to access an actual function (7.17.1)

These words allow it to work. Jim – why not just call them macros? Would that confuse them with 'generic macros'? 'type generic macros'? The proposed words allow for a lot of flexibility. However, they do not really define anything. Adopt Suggested as Proposed, keep OPEN.

Delft discussion: There were no comments submitted to the proposed TC. Move to REVIEW.

DR 421 - OPEN (N1648)

Committee Discussion (Portland)

Clark believes the intent of the Standard is exactly what it says, because different architectures have different conventions. It is not an oversight, as the DR suggests. It could be argued that this is a proposal, not a DR.

Delft: We need a Proposed Committee Response. Draw from first bullet item for that response.

"The standard deliberately does not specify values the clear and set states of atomic_flag objects in order to support the widest possible set of architectures."

Leave OPEN.

DR 422 - OPEN (N1649)

Committee Discussion

The current version of the standard doesn't specify to which value an atomic object should be initialized to if it is initialized by default.

Are atomic qualified ints treated the same as normal ints, i.e. all are initialized to '0', atomic qualified or not ?

The DR is requesting a change that has a big impact on the model. It's really a proposal.

First three bullet items in CD Oct 2012 as Proposed Committee Response.

Leave OPEN.

DR 423 - OPEN (N1650)

Issue: The dealing of rvalues with qualified types is largely underspecified in all versions of the C standard. This didn't surface as a problem until C11, since until then the type of an expression was not observable but only its value.

Committee Discussion

This paper is new enough that a thorough examination of its contents has not been made. It's not clear whether it's a DR or a proposal. If implementers don't know what to do, it's a defect. We really need more time to examine this. Dave Prosser – there are inherent problems with what the Standard says now. Handling of the atomic type qualifier may be the most likely defect, if there is one.

Leave OPEN.

Delft Discussion: The Standard is not exactly clear. We know what it is supposed to say, but it does not seem to say it. Rajan knows of one group who has interpreted the Standard different from what we would expect. Clark believes that the intent of the Standard is stated as Proposal 5 in this DR. Does Proposal 5, Drop all values of rvalues, cover all we would want to say about this. Tom agrees.

ACTION: Clark Nelson to review the applicability of Proposal 5, and write a Propose Technical Corrigenda. – DONE, see below.

Clark submitted the following:

Resolving DR 423

At this point I'm just assuming the submitter has in fact identified all the issues where the standard is insufficiently clear that non-lvalues do not have qualified type; I haven't had time to search for search for others.

Following are the submitter's proposed edits for what he calls Proposal 5, each followed by my commentary.

6.5.1.1, modify as follows:
EXAMPLE The cbrt type-generic macro could be implemented as follows. Here the prefix operator + in the selection expression ensures that lvalue conversion on arithmetic types is performed such that e.g lvalues of type float const select cbrtf and not the default cbrt.

```
#define cbrt(X) _Generic(+(X), \
                         long double: cbrtl,          \
                         default: cbrt,               \
                         float: cbrtf                 \
                         )(X)
```

This edit is not necessary. The controlling expression of a generic selection was very carefully not added to the list of contexts in which lvalue conversion is not done and type qualification is discarded; see 6.3.2.1p2. So the controlling expression of a generic selection can not have qualified type. So, for example, by a careful reading of the standard, a qualified type name in a generic association can never be matched.

The submitter believes that “The intention is clearly ... to distinguish all 8 different forms of qualifications of a type”, and that it is important to be able to do so. However, he is simply mistaken about WG14's intention. (Only time will tell whether this is in fact important.)

6.5.4, add after p2: The type of a cast expression of a qualified scalar type is the scalar type without any qualifiers.

The effect of this proposed edit is desirable. My proposed edit to achieve that effect is instead to change 6.5.4p5:

Preceding an expression by a parenthesized type name converts the value of the expression to the unqualified version of the named type. This construction is called a cast. A cast that specifies no conversion has no effect on the type or value of an expression.

Footnote 104 should also be deleted.

6.5.2.2, add after p1: The type of a function call is the return type of the function without any qualifiers.

6.7.63, change p15, first sentence: For two function types to be compatible, the unqualified versions of both return types shall be compatible.

The combined effect of these edits is desirable: a qualified function return type doesn't make sense. However, I think it would be simpler just to remove any qualifier from the declared return type of a function; that would make the submitter's changes unnecessary. To do that, I propose to change paragraph 5:

…and the type specified for ident in the declaration “T D” is “derived-declarator-type-list T”, then the type specified for ident is “derived-declarator-type-list function returning the unqualified version of T”.

Discussion: The Committee reviewed Clark’s submission, and made several suggestions to changing it.

Clark points out...

This is about the return type of a function, example

```
const int foo(int); // what is the return type?
        // int or const int
        // I believe the answer is int
```

Proposal 5: Drop all qualifiers of rvalues. This matches what I (Clark) believe is the answer
But NOTE: This solution seems the one that is chosen by clang. It is probably the easiest to specify.
As mentioned above it has the disadvantage that the two very similar expressions (int const){0}
and (int const)0 have different types.

Clark will try to come up with proposed TC words: See 423_resolution.htm
Discussion lead to suggested change to 6.5.5.1 saying we mean unqualified type. Change to 6.5.4p5, leave firstsentence of footnote 104. Place the words into the committee discussion. No Proposed TC yet.

Leave Open.

DR 424 - OPEN (N1651)

Section 7.26.6 “Thread-specific storage functions” of C11 is severely underspecified since it uses terms that are not introduced (so far) in the context of C. This is really a pity, since POSIX also has pthread_key_t that is completely feature equivalent and for which the specification is much more complete.

Committee Discussion

Tied to DR 416 – and 5.9 See discussion there.

Leave OPEN

DR 425 - OPEN (N1653)

Section 6.2.4 in p4 and p5 requires implementation defined behavior for accessing objects with thread local or automatic storage from different threads than where they are defined. No such mention is done for objects with *temporary lifetime* in p8. Can they be accessed by other threads? Is this property handled similar to the property for automatic storage duration? Or should this simply be forbidden?

Committee Discussion

There is a Proposed Committee response: Moved to REVIEW

6.3 New DRs established at this meeting.

DR 426 - **N1670 Proposed Defect Report: Appendix G.5.1: -yv and -x/v are ambiguous (Tydeman)**

Discussion: Some of the formulas in the referenced tables seem ambiguous, should we disambiguate them? Clark says they are not ambiguous. The rules are known. If we make it a DR, we can answer it with a formal response.

Delft Discussion: Fred will provide more details to explain why this is ambiguous. Others disagree. How can it be ambiguous? We'll await Fred's details.

DR 427 N1671 Proposed Defect Report: Function Parameter and Return Value Assignments (Miller)

Discussion: There is no example.

ACTION: Douglas to contact the submitter for clarification. Leave OPEN.

DR 428 (N1672) Proposed Defect Report: Runtime-constraint issue with sprintf family in Annex K (Walls)

Delft Discussion: Martin has found additional places where the wording needs to be corrected. We may want to do a comprehensive review of Annex K. A quick review of Martin's findings confirms there are additional cases that need to be addressed. The general direction provided as a Suggested TC looks correct, and should be applied to all applicable use cases. Douglas and Martin will take

Do we agree with the Suggested TC? Yes (17-0-0). Make the Suggested TC a Proposed TC.
Leave OPEN

DR 429 N1673 gets_s

Summary:

"If there is a runtime-constraint violation, s[0] is set to the null character, and characters are read and discarded from stdin until a new-line character is read, or end-of-file or a read error occurs."

The runtime-constraint violation here can be caused by a null "s" pointer. Should we discard the next input line even if (s == NULL) ?

The way it is written, it looks like the answer is yes. However it is not clear to us that that was the intent. Note also that s[0] cannot be set to the null character when s==NULL.

Delft Discussion, Annex citation should be Annex K.3.5.4.1p3 vice p2. What do other implementations do? We need more data.

DR430 (N1674) - getenv_s, maxsize should be allowed to be zero.

If it cannot be zero, why does the 2nd sentence mention the condition that (maxsize != 0) ?
Delft Discussion: The Suggested TC looks good. The words in the Standard look editorially awkward. The rationale seems to support the suggested change.
Straw Poll: Adopt Suggested TC as Proposed TC 9-2-4
Approved. Leave OPEN

DR 431 (N1675) atomic_compare_exchange: What does it mean to say two structs compare equal?

Delft Discussion: The intent is not clear, but there is a Note using memcmp. 7.17.1p5 says 'one of the atomic types' which would include structures. The suggested TC is not really a suggested TC as it offers several choices. The Standard does not seem to specifically include structs as being able to compare equal. What was the original intent? The Suggested TC reads:

Either define equality of objects of struct type, add a restriction disallowing use of atomic structs as arguments for the atomic_compare_exchange generic functions, or note that atomic_compare_exchange generic functions for objects of atomic struct type results in undefined behavior.

What about unions and fixed arrays? Yes, VLAs? Yes. Others as well. Should the list go beyond the list contained in the table in the Standard? TBD

ACTION – Blain to write a Proposed Technical Corrigenda for DR 431

Leave OPEN

DR 432 (N1677) Is 0.0 required to be a representable value?

There is nobody who does not believe 0.0 is not a representable value. Is anybody getting this wrong? Do we need to say something, or is it implicit? Make Fred's suggested change with "if x !=0" removed as the Proposed TC (11-2-2). Leave OPEN.

DR 433 (N1683) Issue with constraints for wide character function arguments involving RSIZE_MAX

What is the real purpose of RSIZE_MAX? Is it to limit buffer sizes? David says the real purpose is to make it easier for the system to detect errors. It is intended to be the number of bytes in an object, as in R(sizeof)_MAX. So, it seems that we got it wrong for wide characters. We need a Suggested TC.

ACTION – Douglas Walls to write a Suggested Technical Corrigenda for DR 433

Discussion of other items:

Subject: memcpy is defined using characters

7.24.1 describes memcpy as "... function copies n characters ...".

3.7 defines character as either a single byte (3.7.1) or multibyte character (3.7.2) (circular definition)

Note that given character could also mean the alphabet type character, there are at least 3 different definitions of characters.

Applies to memcmp as well (7.24.4.1).

Suggested fix:

Change "The memcpy function copies n characters from the object pointed to by s2 into the object pointed to by s1." to "The memcpy function copies n *bytes* from the object pointed to by s2 into the object pointed to by s1."

ACTION – Blain Garth to submit the material contained above as a Defect Report.

7. Resolutions

7.1 Review of Decisions Reached - NA

7.2 Review of Action Items

ACTION: Bill Seymour will communicate the work of WG14 on PDTS 18661, Part 3, to WG21/SG6, Numerics.

ACTION: Small editorial committee (Tydeman, Benito, Thomas, Keyton) to review the changes to N1676 made by the accepted comments contained in N1702.

ACTION: Convener to submit the revised version of PDTS 18661, Part 1, (N1676) to SC22 for PDTS Ballot. (90 day Ballot)

ACTION: Blain Garst to submit the material discussed in item 5, Defect Reports, Discussion of Other Items, memcpy, as a Defect Report.

ACTION: Douglas Walls to write a Suggested Technical Corrigenda for DR 433

ACTION: Blain Garst to write a Proposed Technical Corrigenda for DR 431

ACTION: Douglas Walls to contact the submitter of N1671 for clarification.

ACTION: David Keaton to write a Proposed Technical Corrigenda for DR 413.

ACTION: Convener to forward PDTS 17961, Secure Coding, to DTS Ballot after review by a small editorial committee (Roberto, Clive, Willem, Douglas, Benito, Secord, Keaton)

8. Thanks to Host

The Committee expressed its thanks to Willem Wakker, ACE, and Netherlands Standardization Institute for hosting this meeting in Delft, Netherlands.

9. Adjournment

Adjourned at 1530, local time, Thursday, 4/25/2013

INCITS PL22.11/2013-00001
Reply To The Attention Of: Barry Hedquist
PL22.11 Secretary
Email: beh@peren.com

PL22.11 TAG Meeting Minutes (Draft)
24 April, 2013
Netherlands Standardization Institute
Delft, Netherlands

Meeting convened on April 24, 2013, at 16:15 pm by PL22.11 Chair, David Keaton.

Attendees:

Voting Members:		
Name:	**Organization:** **P – Primary, A - Alternate**	**Comments**
John Benito	Blue Pilot - P	
Jim Thomas	Blue Pilot – A	
David Keaton	CERT/SEI/CMU-P	PL22.11 Chair
Daniel Plakosh	CERT/SEI/CMU-A	
Robert Secord	CERT/SEI/CMU-A	
Roger Scott	Coverity - P	
Tana Plauger	Dinkumware, Ltd – A	
P. J. Plauger	Dinkumware, Ltd – P	
Blaine Garst	Garst - P	
Rajan Bhakta	IBM - P	
Clark Nelson	Intel - A	
John Parks	Intel - P	
Clive Pygott	LDRA - P	
Herb Sutter	Microsoft - P	
Douglas Walls	Oracle - P	PL22.11 IR
Barry Hedquist	Perennial – P	PL22.11 Secretary
Tom Plum	Plum Hall, Inc. – P	
Bill Seymour	Seymour - P	
Fred Tydeman	Tydeman Consulting – P	PL22.11 Vice Chair
Blaine Garst	Self	
Non-Voting:		
Owen Shepherd	Self	France

1. Approval of Agenda

Revisions to Agenda: None

Added Items: None

Deleted Items: None

Agenda approved by unanimous consent. (Garth/Tydeman)

2. Approval of Previous Minutes (PL22.11/12-002)

Minutes were modified per editorial changes and approved by unanimous consent.
(Hedquist/Benito)

3. Selection and Review of US Delegation.

Motion: (Walls/Benito)

The US delegation for SC 22/WG14 meetings during 2013 will be all Principals and Alternates for PL22.11 including new members and representatives that join the committee in-between PL22.11 meetings and before the next US delegation vote.

Approved (Unanimous Consent)

4. INCITS Anti-Trust Guidelines

We viewed the slides located on the INCITS web site.
http://www.incits.org/inatrust.htm

5. INCITS official designated member/alternate information.

Be sure to let INCITS know if your designated member or alternate changes, or if their email address changes. Send contact info to Lynn Barra at ITI, lbarra@itic.org.

6. Identification of PL22.11 Voting Members (Tydeman)

See attendance list above.
15 PL22.11 voting members participated out of 17.

6.1 PL22.11 Members Attaining Voting Rights at this Meeting

Microsoft

6.2 Prospective PL22.11 Members Attending Their First Meeting

None

7. Members in Jeopardy

7.1 Members in jeopardy due to failure to return Letter Ballots.

Bloomberg, Cisco, Coverity, Seymour

7.2 Members in jeopardy due to failure to attend Meetings.

HP

7.3 Members who lost voting rights attending this meeting.

Microsoft

8. New Business - None

9. Next Meetings: Chicago with WG14

- Fall 2013: Chicago, IL, USA, Sept 28 - Oct 3, 2013, with WG21.
- Spring 2014: Parma, Italy
- Fall 2014: St. Louis –Dates: 3-8 Nov C++, C last week of October? Bill Seymour will follow up.
- Spring 2015: Host needed.

10. Adjournment

There being no further business, the meeting was adjourned at 4:35 PM local, October 23, 2012 (Garst/Benito) - Unanimous Consent.