

ISO/IEC JTC1 SC22/WG14 AND INCITS PL22.11
OCT 24 - 25, 2011, MEETING MINUTES

Meeting Location:

Information Technology Industry Council (ITIC)
1101 K Street, NW
Suite 610
Washington, DC 20005
USA

Host Contact information:

John Benito
E-mail: benito@bluepilot.com

Scheduled Meeting Dates: 24 - 28 Oct, 2011

Scheduled Meeting Times:

24 Oct 2011 09:30 am to 4:30 pm
25 Oct 2011 09:00 am to 4:30 pm
26 Oct 2011 09:00 am to 4:30 pm
27 Oct 2011 09:00 am to 4:30 pm
28 Oct 2011 09:00 am to 12:00 pm

1. OPENING ACTIVITIES

1.1 Opening Comments (N1572) (Spittle, Benito)

John Benito and Debbie Spittle welcomed us to the ITIC offices in Washington DC, and described the meeting facilities. Several local restaurants are within walking distance of the facility. Lunch break will be from 11:30 - 13:00. We are connected on WebEx to allow folks to call in.

1.2 Introduction of Participants/Roll Call

Name	Organization	NB	Comments
John Benito	Blue Pilot		WG14 Convener
Blaine Garst	Self	USA	
David Keaton	CMU/SEI/CERT	USA	PL22.11 Chair
Robert Secord	CMU/SEI/CERT	USA	
David Swvoda	CMU/SEI/CERT	USA	
Tana L. Plauger	Dinkumware, Ltd	USA	
P. J. Plauger	Dinkumware, Ltd	USA	
Jim Thomas	HP	USA	

Hans Boehm	HP	USA	
Rajan Bhakta	IBM	Canada	HOD - Canada
Michael Wong	IBM	Canada	
Clive Pygott	LDRA	USA	
Douglas Walls	Oracle	USA	HOD – USA, PL22.11 IR
Sheldon Lobo	Oracle	USA	
Raj Prakash	Oracle	USA	
Barry Hedquist	Perennial	USA	PL22.11 Secretary
Tom Plum	Plum Hall, Inc.	USA	
Fred Tydeman	Tydeman Consulting	USA	
Larry Jones	Siemens	USA	WG14 Project Editor
Bill Seymour	Seymour	USA	
Douglas Gwyn	Self	USA	
Nick Stoughton	Usenix	USA	POSIX Liaison
Chucky Ellison	University of Illinois	USA	
Roger Scott	Coverity	USA	
Martin Sebor	Cisco	USA	
Rex Jaeschke	Self	USA	SC22 Chair
Mark Batty	University of Cambridge	UK	
Peter Sewell	University of Cambridge	UK	

1.3 Procedures for this Meeting (Benito)

The Meeting Chair, John Benito, announced the procedures are as per normal. Everyone is encouraged to participate in 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 future projects for WG14.

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

1.4 Approval of Previous Minutes - London (N1550) (Hedquist)

Several comments for typos, etc.

Minutes were modified per editorial changes and approved.

Final London Minutes are **N1587**.

1.5 Review of Action Items and Resolutions (Hedquist)

ACTION – Editorial Review Group to meet on or about 4 April, 2011: JB, David Keaton, Blaine, Garst, Nick Stoughton, Douglas Walls, Jim Thomas, Larry Jones. Blaine can host. Comments can only be applied to Larry's edits; conforming to what we've voted on in this meeting. 1 PM at Apple. Expect to forward on or before 11 April. Allow no less than 6 months for another meeting.

DONE - The document was reviewed and forwarded.

ACTION – Convener forward the WP as revised in this meeting to the inquiry stage of ballot, DIS, 5 months, ITTF managed. If there are no "DISAPPROVE" votes, and no technical comments, WG14 should be able to publish. Even if an additional 2-month ballot is needed, the new C standard should be published in early in 2012.

DONE: ISO/IEC DIS 9899 was submitted to ITTF, and approved with no comments. The DIS will be forwarded to ITTF for final publication per ISO/IEC rules.

1.6 Approval of Agenda (N1573) (Benito)

Revisions to Agenda:
Future meeting info

Added Items: None

Deleted Items: None

Agenda approved as modified.

1.7 Distribution of New Documents (Benito)

None

1.8 Identification of National Body Delegations (Benito)

US, Canada

1.9 Identification of PL22.11 Voting Members (Tydeman)

See PL22.11 Minutes, following these minutes. 12 of 16 members present.

2. LIAISON ACTIVITIES

2.1 SC22/WG14 & INCITS/PL22.11 C Language (Benito, Walls, Keaton)

DIS Ballot PASSED with no comments, and has been forwarded to ITTF for publication. Expect publication in 2011. Richard Cook is the contact at ITTF. US TAG meeting, Tuesday, 4PM. No US Positions on the table.

2.2 SC22/WG21 & INCITS/PL22.16, C++ Language (Plum, Benito)

The ISO/IEC FDIS 14882, C++, was forwarded to ITTF for final ballot following the WG21 meeting in Madrid, Spain, and has been approved. The new revision of C++ will be known as C++ 2011.

2.3 SC22 & INCITS/PL22, Programming Languages (Plum)

SC22 met in plenary session in September
PL22 meets twice a year via teleconference.

2.4 WG11

WG11 was dissolved at the SC22 plenary meeting.

2.6 WG23 (Benito)

A revision to their TR is underway. WG23 met in Madrid, Spain in March 2011.
For further information, contact John Benito. WG 23 Will meet in December, at INCITS, and in March, June, Sep, and Dec.

2.6 MISRA C (N1580) (Montgomery)

MISRA working on Rev 3 of their C Guidelines, based on C99, and correcting issues in prior versions. Public review version is now expected by the end of 2011. MISRA has opened pre-registration for the public review. Anyone wishing to participate in the review can find more information and an application on the MISRA C website. Go to <http://www.misra-c.com/> and click the News tab.

2.7 Floating-point Study Group (Thomas)

A paper, N1582, has been submitted and is on the agenda (See 5.5).

2.8 Secure Coding Study Group (Keaton)

Work is progressing. Three documents have been submitted and are on the agenda: N1577, N1578, N1579. (See 5.2 - 5.4).

2.9 Austin Group (Stoughton)

One item, a potential Defect Report on realloc(). Nick will provide. Plan on discussing Tuesday AM.

2.10 Other Liaison Reports - none

3. EDITOR REPORTS

3.1 Report of the Rationale Editor (Benito)

Benito expects to have a rationale ready soon.

3.2 Report of the Project Editor (Jones)

No news is good news. No feedback from ITTF.

3.3 Report on Editorial Meetings Held (Benito, Jones)

None held. DIS was in ballot.

4. FUTURE

4.1 Future Meeting Schedule

February 2012 (N1574) (Plum)

WG14 meeting is sponsored by Bloomberg LP, with local hosting arrangements by Plum Hall. The meeting will take place at the Royal Kona Resort, in Kailua-Kona, Hawaii, Monday Feb 13 to Friday Feb 17, 2012. WG21 will be meeting the week prior, Feb 6 - 10, at the same location. Hotel reservations and be made at: www.royalkona.com/bloomberg.

Fall 2012: Oct 22 – 26, Portland, OR 2012 (C++ prior week ?)

Spring 2013: Open. Either the week before or after the WG21 meeting (WG21, Cambridge ?)

Fall 2013: Chicago area, – Dates TBD (also w/C++)

4.2 Future Agenda Items

Per Normal

4.3 Future Mailings

Post Washington DC, 28 Nov, 2011

Pre Kona: 16 Jan, 2012

Post Kona: 18 April, 2012

5. Document Review

5.1 Possible Errors in Decimal Floating Point (Tydeman) (N1312) (N1550)

Fred Tydeman presented a list of possible issues with the current Technical Report for Decimal Floating Point. The issues fall into the following categories:

1. Existing items that may need to be changed.

2. Items missing from the TR.

It is not clear why these changes are being proposed, i.e. there's a lack of substantive technical rational.

This proposed mod is to a C99 TR.

Tom – where's the beef. Fred ran into issues while writing tests.

1.2 – this area is touched on in the TR. Jim T disagrees with Fred's proposal. 1.0 is not less exact than 1.00000000, they are the same exactness.

1.3 – agree

1.4 – Jim T, Annex F of the C standard may already address this.

1.5 – What does correctly rounded mean? It depends on the rounding direction. Already covered.

1.6 – cannot change the name of a pragma. No

1.7 – Jim T thinks these words are needed. OK

1.8 – item was left out – OK

Concluded discussion at this point to address the overall approach we want to take for this paper. It looks like a new edition to the TR might be warranted. JB thinks the doc we have is OK, would rather see it tied to an IEEE 754 C binding. Jim T would prefer to keep it separate; an edition to the existing TR. Jim believes it would be a cleaner package. We need to decide whether we, WG14, want to do another edition to the DFP TR, and if so, how we want to do it. We have two options: 1) a stand-alone new edition, or 2) part of the IEEE 754 C Binding effort. Revisit this approach when we review the IEEE 754 material (See Item 5.5).

5.2 Rationale for C Secure Coding Analysis, Technical Specification (N1577) (Keaton)

N1577 provides a rationale for the development of a Technical Specification for a C language code analyzer for C secure coding rules.

There are several vendors of static analysis tools that look for bugs in general, and some security issues. Each has a different idea about what to look for. The goal is to identify an ‘at least this much’ model of what to check for if the tool is aimed at security issues. Is the market large enough? Yes. It’s seen as a growing market. There is a tendency for compilers to do a certain level of checking, and that is expected to expand. The decision is to do a Technical specification because 1) the work needs to be normative, but 2) it is too early to do a Standard (not enough is known). A base document is ready for processing as a New Work Item. The SG has been working on this for two years, and believes this is the next step in the process to carry this work forward.

Process issues, a recent change. If a NWI fails, i.e. lack of support, we are not supposed to bring it up again. Good idea to get five NBs to sign on prior to processing a NWI.

Raj opined that most programming errors are dynamic rather than static, and is concerned that such a tool coming from a Standards Committee would give folks a false sense of security. Roger disagrees. The number of problems found via static analysis is quite large, and everyone knows that static analysis is necessary, but not considered a catchall. Raj disagrees, and thinks most problems are found dynamically. Robert Secord pointed out that too many use only one tool, and get a false sense that their code is now secure. Roger Scott pointed out that there is no standard for a tool along these lines, and that anything would be a step in the right direction.

NIST has been running tests using static analysis tools that focus on security, but the process has been subjected to vendors arguing about what and what is not a security problem. So, results are inconclusive in terms of hard data. However, it is clear that exploitable issues can be found. Blain concerned that the rules set is in fact worthy of use.

Concern about the restricted nature of this document, if there is any. There is none. The 'boilerplate' ISO verbiage is unintentional, and will be removed.

Raj: Not every error found by such a tool is not necessarily a real error for every domain.

The way users actually use the tool is outside the scope of this project. The scope is confined to developers of such an analysis tool as a minimum set of check, not on which checks the users invoke.

Technical Specification v. Technical Report. A TR cannot contain normative language, thus a Technical Specification will better fit the purpose of the document.

ACTION: Convener to determine National Body interest in a Technical Specification for a security focused static analyzer, and if sufficient, to forward a New Work Item for a Technical Specification to SC22.

5.3 C Secure Coding Rules Editor's Report (N1578) (Keaton)

N1578 is the first Editor's Report for the draft Technical Specification, C Secure Coding Rules, and lists a number of issues remaining. David believes the issues listed can be resolved within the next 12 months.

5.4 C Secure Coding Rules, Draft Technical Specification, (N1579) (Keaton)

N1579 is the first draft of a Technical Specification for C Secure Coding Rules, and is based on "The CERT C Secure Coding Standard". It specifies rules for secure coding in the C programming language, and provides code examples of those rules.

A number of unpublished comments have been generated by those who have reviewed this document. The comments were discussed, and where applicable, adopted for inclusion in the draft. Details of adopted comments were recorded by Robert Secord.

There was considerable concern expressed over the issue of 'false positives'. How different are quality issues from security issues? i.e. false positive rates. Many security types may not care about having to sort through lots of false positives in order of find the one thing that will cause a real problem.

Roger: The list of library functions that required diagnosis needs to be cut down by a LOT, otherwise the false positive noise will be unbearable.

The basis of this specification comes from reports generated in the field, so new material in C2011, i.e. Atomics, has not been taken into consideration.

Revised document by pre-Kona mailing.

Expect the Word version of the document to become available for review on the WG14 Wiki.

ACTION: Robert Secord to make sure the Draft Technical Specification for C Secure Coding Rules is available to WG14 committee members prior to the pre-Kona meeting.

5.5 C Support for IEEE 754-2008 (N1582) (Thomas)

N1582 is a set of briefing charts that outlines recommendations by the WG14 C Floating Point Study Group for C language support of IEEE Standard 754 - 2008. That standard was adopted by ISO/IEC in 2011 as an update to IEC 60559, which the C Standard uses as a reference. It makes recommendations for Language Standards, but does not provide any bindings.

Earlier discussion of item 5.1, N1312, proposed inclusion of that material into the scope of the effort being addressed by this Study Group.

There are a number of new features in IEEE 754 that have not been seen by C in the past, although most of the features are already covered in C99. C1X does provide support for the original FP Standard, but not the newer one. Decimal Floating Point has been added to the scope of the Study Group. The plan is to write a Technical Specification (TS) that addresses updates needed for Decimal Floating Point (DFP) (N1582), as well as new Binary Floating Point (BFP) material in the new IEEE 754 (IEC 60559). The new TS would also replace Annex F in the new C1X. We will need to avoid creating any conflict between Annex F and the TS.

For simplicity, new functions are presented as the 'double' version. It follows that float, long double and type generic functions apply for each new function listed.

There is some question on how much of the new IEEE 754 Standard is really needed. It's huge, but at least on vendor, Intel, is implementing it. The resources available in the Study Group are limited, and we may want to consider doing a subset of this rather than trying to do the whole ball of wax at once. An approach using a Part I, Part II, etc., may be a viable approach to pursue.

5.6 Working Draft of 2nd Edition of TR 24772 (N1583) (Benito)

N1583 is based on ISO/IEC TR 24772 (Type 3), Guidance to Avoiding Vulnerabilities in Programming Languages through Language Selection and Use, developed by SC22/WG23. Of particular interest is the informative Annex C, Vulnerability descriptions for the language C.

Python, Ada, and Spark Ada Annexes are coming. The C Annex can be commented on by anyone on the Committee. It belongs to WG23, but WG23 will be receptive to any comments we have. The annex must be taken in context of the base document, Clauses 4 – 7. The codes used to identify vulnerabilities are unique to WG23.

5.7 Realloc with size zero problems, (N1586) (Stoughton)

There are at least three existing realloc behaviors when NULL is returned; the differences only occur for a size of 0 (for non-zero size, all three implementations set errno to ENOMEM when returning NULL, even though that is not required by C99).

Suggested Change

At 7.22.3, para 1, change:

If the size of the space requested is zero, the behavior is implementation-defined: either a null pointer is returned, or the behavior is as if the size were some nonzero value, except that the returned pointer shall not be used to access an object.

To

If the size of the space requested is zero, the behavior is implementation-defined: either a null pointer is returned and `errno` set to indicate the error, or the behavior is as if the size were some nonzero value, except that the returned pointer shall not be used to access an object.

Add a footnote to this sentence stating:

Note Memory allocated by these functions should be freed via a call to `free`, and not by means of a `realloc(p, 0)`.

Discussion:

We discussed this issue in London, and basically decided to leave it alone. Different compiler vendors handle this in different ways. Some set `errno`, some do not. We don't want to break implementations that do not set `errno`. Nor do we want to specify what the choices need to be. There is no consensus to make a change, but we will accept this as a Defect Report and process it. DR 400. Will be on the agenda for Kona (Feb 2012).

5.8 Memory Model (N1584) (Mark Batty)

N1584 points out differences between the C and C++ memory models, and corrections needed to make them compatible. Some other problems in the current draft (DIS) and suggestions as to how they might be fixed are also identified. Mark Batty was available to teleconference for part of the DC meeting to discuss these items.

Changes were made to the C++11 memory model in the final iterations of the standardisation process to fix some problems that our work brought to light. We made a mathematical semantics of the memory model of the language and some tools to test small examples with. Although many important problems were fixed, there was not enough time before the finalisation of the standard to fix everything. Looking at N1569, it appears that many of the changes that were made were not carried over to C, so in the second part of this document we list the textual changes that are needed. First, we describe the new issues that we would like to discuss at the meeting in Washington DC next month.

--- New issues to discuss ---

- 1.1 Visible sequences of side effects are redundant
- 1.2 SC fences do not restrict modification order enough
- 1.3 Should locks provide intra-thread synchronisation?

--- Changes to bring over from C++11 ---

- 2.1 Happens before cannot be cyclic
- 2.2 Coherence
- 2.3 Malloc and free in the memory model
- 2.4 A joke fragment remains in a footnote
- 2.5 The mutex specification
- 2.6 Do conflicting un-sequenced accesses have undefined behavior?

--- **1.1 Visible sequences of side effects are redundant** ---

We have mathematically proved that a simplification can be made to the memory model as it is specified in the final draft of the C++11 standard. Essentially, the restriction defining 'visible sequence of side effects' (vsse) is redundant and can be removed with no ill effects. The main motivation for doing this is that the current restriction is misleading. 5.1.2.4p22 defines vsse's:

The visible sequence of side effects on an atomic object M, with respect to a value computation B of M, is a maximal contiguous sub-sequence of side effects in the modification order of M, where the first side effect is visible with respect to B, and for every subsequent side effect, it is not the case that B happens before it. The value of an atomic object M, as determined by evaluation B, shall be the value stored by some operation in the visible sequence of M with respect to B.

The wording of this paragraph makes it seem as if the vsse identifies the writes that an atomic read is allowed to read from, but this is not the case. There can be writes in the vsse that cannot be read due to the coherence requirements (to be included in C, 1.10p15 through 1.10p18 in C++ N3291, the latest version I have access to). Consequently this is even more confusing than it at first appears.

We propose changing 5.1.2.4p22 to the following:

The value of an atomic object M, as determined by evaluation B, shall be the value stored by some side effect A that modifies M, where B does not happen before A.

With a note to remind the reader of the coherence requirements:

NOTE The set of side effects that a given evaluation might take its value from is also restricted by the rest of the rules described here, and in particular, by the coherence requirements below.

If the committee is concerned about allowing a differing text from C++11, then a note could be added to assure the reader:

NOTE: Although the rules for multi-threaded executions differ here from those of C++11, the executions they allow are precisely the same. Visible sequences of side effects are a redundant restriction.

The above-proposed change would be contingent on accepting 2.2 below.

--- 1.2 SC fences do not restrict modification order enough ---

C1x seems to omit the restriction imposed in C++11 in 29.3p7 (from N3291):

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.

Furthermore, it seems that both C1x and C++11 are missing the following two derivatives of this rule:

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.

Need a diagram to understand this, which can be added.

--- 1.3 Should locks provide intra-thread synchronisation ---

Most of the C++ standard, synchronisation is used exclusively inter-thread, so in particular, synchronisation can't be used to avoid undefined behavior arising from conflicting un-sequenced memory accesses, e.g.:

(x = 1)==(x = 2)

Firstly, C does not define this sort of thing as undefined behavior. Is this intentional? Secondly, in C++ locks can currently be used to fix up such programs and avoid undefined behavior, e.g.:
(lock; x = 1; unlock; x) == (lock; x = 2; unlock; x)

The reason not to allow this sort of synchronisation in general is, I believe, because it disallows some single threaded compiler optimizations. Is intra-thread locking intended to be defined and usable?

It's undefined behavior, per rules that go back to C90.

Two locks unsequenced on the same thread undefined behavior? Affects the sequence point model, which Clark Nelson needs to look at.

All of the items above (1.1 – 1.3) will be treated as Defect Reports, and processed accordingly.

--- 2.1 Happens before cannot be cyclic ---

C++11 forbids happens before from being cyclic, but this change has not made its way into C1x. In order to fix this, the following sentence (taken from C++ N3291, 1.10p12) should be added to 5.1.2.4p18:

The implementation shall ensure that no program execution demonstrates a cycle in the "happens before" relation.

NOTE This cycle would otherwise be possible only through the use of consume operations.

--- 2.2 Coherence ---

The memory model described in N1569 matches an older version of the C++0x memory model, one that allowed executions that were not intended by the designers. We recommend matching the C++11 text by removing the sentence starting 'Furthermore' in 5.1.2.4p22, and including the following paragraphs in section 5.1.2.4 (Taken from C++ N3291, 1.10p15 through 18):

If an operation A that modifies an atomic object M happens before an operation B that modifies M, then A shall be earlier than B in the modification order of M.

NOTE The requirement above is known as write-write coherence. If a value computation A of an atomic object M happens before a value computation B of M, and A takes its value from a side effect X on M, then the value computed by B shall either be the value stored by X or the value stored by a side effect Y on M, where Y follows X in the modification order of M.

NOTE: The requirement above is known as read-read coherence. If a value computation A of an atomic object M happens before an operation B on M, then A shall take its value from a side effect X on M, where X precedes B in the modification order of M.

NOTE: The requirement above is known as read-write coherence. If a side effect X on an atomic object M happens before a value computation B of M, then the evaluation B shall take its value from X or from a side effect Y that follows X in the modification order of M.

NOTE: The requirement above is known as write-read coherence.

--- 2.3 Malloc and free in the memory model ---

The synchronisation afforded to malloc and free is missing some vital ordering, and as the definition stands it makes little sense and creates cycles in happens before. C++11 includes a total order over the allocation and deallocation calls, which fixes the problem and seems to give a sensible semantics. From 18.6.1.4p1 in N3291:

Calls to these functions that allocate or deallocate a particular unit of storage shall occur in a single total order, and each such deallocation call shall happen before the next allocation (if any) in this order.

Unfortunately, there is a typo here. Happens before edges are not transitively closed in to the happens before relation, but the edges here are meant to be. Instead the sentence above should create a synchronizes with edge. In light of this, we suggest removing the last two sentences from 7.22.3p2 and replacing them with:

Calls to these functions that allocate or deallocate a particular region of memory shall occur in a single total order, and each such deallocation call shall synchronize with the next allocation (if any) in this order.

--- 2.4 A joke fragment remains in a footnote ---

C1x seems to have inherited part of a joke from C++, which ought to be removed or made whole and annotated as such. Originally, C++0x had the footnotes:

"Atomic objects are neither active nor radioactive" and "Among other implications, atomic variables shall not decay".

The first is pretty clearly a joke, but it's not obvious that the second doesn't have some technical meaning, and that is the one that remains in C1x in 7.17.3p13.

--- 2.5 The mutex specification ---

The C1x specification of mutexes is missing the total order over all the calls on a particular mutex. This is present in C++11. The following is from 30.4.1.2p5 in N3291:

For purposes of determining the existence of a data race, these behave as atomic operations (1.10). The lock and unlock operations on a single mutex shall appear to occur in a single total order. [Note: this can be viewed as the modification order (1.10) of the mutex. End note]

The synchronisation in 7.26.4 is defined in terms of some order over these calls, even though none is specified, for instance 7.26.4.4p2 reads:

Prior calls to mtx_unlock on the same mutex shall synchronize with this operation.

This seems like simple omission. We suggest adding a new paragraph to 7.26.4 that matches C++11:

For purposes of determining the existence of a data race, mutex calls behave as atomic operations. The lock and unlock operations on a single mutex shall appear to occur in a single total order.

NOTE This can be viewed as the modification order of the mutex.

--- 2.6 Do conflicting un-sequenced accesses have undefined behavior? ---

See 1.3.

Section 2 items need to be handled as Defect Reports, and processed accordingly, except for 2.6 which is already covered.

6. OTHER BUSINESS

6.1 C1X Schedule

The DIS 9899 ballot was approved with no comments and has been forwarded to ITTF for publication.

7. RESOLUTIONS

7.1 Review of Decisions Reached NONE

7.2 Review of Action Items

Carry Over - NONE

New

ACTION: Convener to determine National Body interest in a security focused static analyzer, and if sufficient, to forward a NWI for a Technical Specification to SC22.

ACTION: Robert Secord to make sure the Draft Technical Specification for C Secure Coding Rules is available to WG14 committee members prior to the pre-Kona mailing.

ACTION: Blain Garth to work with Mark Batty to clarify DR 408.

8. THANKS TO HOST

The Committee expresses its great appreciation and thanks to Dinkumware for the years of hosting the WG14 Wiki since it was created.

The Committee also expresses its great appreciation and thanks to Keld Simolsen for taking over the hosting the WG14 Wiki.

Thanks to EDG for hosting the FP Study Group Wiki, and providing the link to the WG14 Wiki

Thanks also to our host INCITS & ITI.

9. ADJOURNMENT

Meeting adjourned at 2:36 pm, Tuesday, Oct 25, 2011.

PL22.11 Meeting 25 October 2011

Meeting convened at 2:20 PM, Oct 25, 2011, by PL22.11 Chair, David Keaton.

Attendees:

<u>Voting Members:</u>		
Name:	Organization: P – Primary, A - Alternate	Comments
John Benito	Blue Pilot - P	
Martin Sebor	Cisco - P	
David Keaton	CMU/SEI/CERT - P	PL22.11 Chair
Robert Secord	CMU/SEI/CERT - A	
David Swvoda	CMU/SEI/CERT - A	
P. J. Plauger	Dinkumware, Ltd – P	
Tana L. Plauger	Dinkumware, Ltd – A	
Jim Thomas	HP – P	
Rajan Bhakta	IBM - P	
Clive Pygott	LDRA - P	
Douglas Walls	Oracle - P	PL22.11 IR
Sheldon Lobo	Oracle - A	
Barry Hedquist	Perennial – P	PL22.11 Secretary
Tom Plum	Plum Hall – P	
Bill Seymour	Seymour - P	
Fred Tydeman	Tydeman Consulting – P	PL22.11 Vice Chair
Douglas Gwyn	Self	Member Emeritus
<i>Prospective New Members</i>		
Blaine Garst	Self	
Roger Scott	Coverity	

1. Approval of Agenda

The agenda, as modified, was approved by unanimous consent.

2. Approval of Previous Minutes (PL22.11/11-001)

Minutes were modified per editorial changes and approved by unanimous consent.

3. Selection and Review of US Delegation.

All members of PL22.11, who are not JTC1 Officers, are members of the US delegation to WG14.

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.
12 PL22.11 voting members participated out of 16.
- 6.1 **PL22.11 Members Attaining Voting Rights at this Meeting**
Cisco, Martin Sebor.
- 6.2 **Prospective PL22.11 Members Attending Their First Meeting**
Blain Garst, Self
Roger Scott, Coverity
7. **Members in Jeopardy**
- 7.1 **Members in jeopardy due to failure to return Letter Ballots.**
None
- 7.2 **Members in jeopardy due to failure to attend Meetings.**
Intel
8. **New Business**
None
9. **Adjournment**
Adjourned at 2:31 PM local, Oct 25, 2011