MINUTES FOR OCTOBER 2006 (DRAFT)
MEETING OF ISO/JTC1/SC22/WG14 AND INCITS J11
WG14/N1198

Meeting Dates and Times

23 October 2006 09:00-12:00 13:00-17:00
24 October 2006 09:00-12:00 13:00-17:00
25 October 2006 09:00-12:00 13:00-17:00
26 October 2006 09:00-12:00 13:00-17:00
27 October 2006 09:00-12:00

Meeting Location:

DoubleTree Hotel & Executive Meeting Center
Portland Lloyd Center
1000 NE Multnomah St.
Portland OR 97232
Phone: +1-800-996-0510
Fax: +1-503-284-8553
http://www.portlandlloydcenter.doubletree.com/

Host:
ANSI
Host Contact information:
Clark Nelson <mailto:clark.nelson@intel.com> +1-503-712-8433
2111 NE 25th Ave, JF1-13
Hillsboro OR 97124
USA
Hotel Information:
See document WG14/N1153

1. Opening activities

1.1 Opening Comments (Nelson, Benito)

Clark Nelson, Intel Corporation, welcomed us to the Executive Meeting Center at the Double Tree Hotel in Portland. Food and beverages will be available throughout the meeting, so breaks can be taken at any time. Small lockers with keys are available to store computers during lunch break. Keys are available at the EMC front desk.

1.2 Introduction of Participants/Roll Call

John Benito	WG14 Convener	USA
Randy Meyers	Silverhill Systems	USA

Douglas Walls	Sun Microsystems	USA	HOD
Fred Tydeman	Tydeman Consulting	USA	
Barry Hedquist	Perennial	USA	
Cecilia Galvan	Freescale	USA	
P. J. Plauger	Dinkumware, Ltd	USA	
Tana L. Plauger	Dinkumware, Ltd	USA	
Mark Terrel	Cisco	USA	
Nick Stoughton	Usenix	USA	
John Parks	Intel	USA	
Clark Nelson	Intel	USA	
Bill Seymour	self	USA	
Martyn Lovell	Microsoft	USA	
Tom Plum	Plum Hall	USA	
Christopher Walker	Dinkumware, Ltd	USA	
Ulrich Drepper	Red Hat	USA	
Doug Gwyn	US Army	USA	
Lawrence Crowl	Google	USA	
Rich Peterson	Hewlett Packard	USA	
Dan Gohman	Cray	USA	
Janis Johnson	IBM	USA	
David Keaton	Self	USA	
Edison Kwok	IBM	CANADA	HOD

1.3 Procedures for this Meeting (Benito)

The Chair, Benito, announced the procedures are as per normal. Everyone is encouraged to participate in straw polls. INCITS J11 members are reminded of the requirement to follow the INCITS Anti-Trust Guidelines which can be viewed at http://www.incits.org/inatrust.htm.

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

1.4 Approval of Previous Minutes (N1168) (Hedquist)

Comments from Fred Tydeman:

1.2: Column alignment could be improved.

9. TR 24732 (PDF page 10 of 24)

Nobu raised ...: "character encoding" -> "decimal FP encoding".

J11 TAG: Column alignment could be improved.

Minutes approved as modified.

1.5 Review of Action Items and Resolutions (Hedquist)

1. ACTION: Randy to write a paper with a proposed response for DR 314. OPEN

2. ACTION: Randy and Mark to work with Nick to write up a draft response to N1160. DONE

3. ACTION: Randy to add words to the rationale for TR 24731 to better clarify the issues raised in N1160. OPEN

4. ACTION: Edison to add words to the Decimal FP TR rationale to better explain why we need all the values provided for DEC_EVAL_METHOD DONE

5. ACTION: Fred to work with Edison to write up FLT_DEN with N1151. DONE

6. ACTION: Convenor to ask Sally Seitz (ANSI) to distribute the Rationale during Registration Ballot for TR 24732, Decimal Floating Point. DONE

7. ACTION: Convenor to provide a response to the submitter of N1152 explaining that we are not revising the Standard at this time. DONE

8. ACTION: PJ to write up a revised version of the text in C99: 7.12.1;p2, and for a proposed defect report. DONE

9. ACTION: Convenor to forward the draft of PDTR 24733, Special Math Functions, as revised by the editorial committee, to SC22 for Registration. DONE

10. ACTION: Convenor to forward the draft of PDTR 24732, Decimal Floating Point, as revised by the editorial committee, to SC22 for Registration. DONE

11. ACTION Convenor to forward the draft of TR 24731, Bounds Checking, to SC22 for DTR ballot. DONE

12. ACTION: Convenor to forward TR 18037, Support for Embedded Systems, to SC22 for publication, again. OPEN

1.6 Approval of Agenda (N1190)

New Doc on Wiki, N1193, add to agenda after Liaison Activities

2.4 WG21 Liaison Report. Add discussion on C++ Concurrency.

MOTION: Approve the agenda as modified. No objection.

1.7 Information on Next Meeting. (Benito)

Next meeting will be in London, 23-27 April 2007. Exact location is TBD. Meeting room at The Association of Chartered Accountants.

Fall 2008 will be in Kona. Douglas Walls will submit paperwork for invite to ANSI. Arrangement to stay after the meeting should be made now, since Ironman will follow the meeting.

1.8 Identification of National Bodies (Benito)

The following National Bodies are represented:

USA

Canada

1.9 Identification of J11 voting members (Tydeman)

J11 has 13 out of 18 possible voting members present. See attached J11 membership list for attendees.

2.0 Liaison Activities

2.1 J11 (Walls, Meyers)

Belated call for Chair has gone out, Randy responded. Randy gave a report on the INCITS officer symposium held in Washington DC, where he gave the J11 Annual Report. Lots of representation from Dept of Homeland Security.

Discussion is needed on CD Registration Ballot Comments

2.2 SC22 (N1192) (Benito)

Document is a report on the SC22 Plenary. Having a problem getting ITTF to republish TR18037, Embedded System Support.

TR 24731 Part I, is finished. To be sent out for publication by ITTF.

TR 24732 Decimal Floating Point - Passed registration Ballot, no comments

TR 23733 Special Math - in registration ballot at SC22

2.3 WG11 (Wakker)

No report.

2.4 J16/WG21 (Crowl) N1196, (Plum), (Nelson) TBS [WG21 papers as well – from Lawrence]

Lawrence Crowl gave a presentation on where C++ threads is going. C++ is trying to make threads correspond to operating system threads, POSIX and Windows, but not cover all of the functionality. They will be heavyweight, preemptive, and independent. Not a replacement for existing Standards. Presuming that all writes are available to all threads is not viable, and C++ wants to adopt a message memory model.

Sequencing has been redefined. Sequence points are gone. Relations are sequenced-before and indeterminately-sequenced. A write-to and read-from a location that are not sequence-before and not indeterminately-sequenced results in undefined behavior.

The sequence rules have been extended to concurrency. Relations are synchronizes-with and happens-before are based on acquire and release. Race conditions are defined. The existence of a race condition makes the program undefined.

A location is a non-bitfield primitive data object. Adjacent bitfields together constitute a single location, enables unsynchronized read-modify-write.

The C++ Committee decided there needs to be some compromised between the language and the hardware. Ulrich has an opinion that needs to be communicated back to WG21 via liaison. He has been talking with Hans Boehm in WG21 about this and prefers to have the POSIX type binding for threads (pthreads).

Optimizers are affected. Some speculative writes and reads are no longer legal. Loops without synchronization may be assumed to terminate.

All threads observe the same sequence of values for an atomic type. Atomic operations acquire, release, both, and neither. Atomic types are struct, but could be primitive type. Operators are type generic macros in C. The default assignment operator for both C and C++ is wrong, and cannot be disable in C99. It can be disable in C++98, but doing so breaks C/C++ compatibility. C++ has a template for making any type atomic.

The types are comprehensive over the important primitive types: atomic_flag, atomic_bool, atomic_integers, atomic_void_pointer. Atomics may be compiled by both C and C++. Atomic operations must be lock-free to be used in signals. A macro will indicate if a type is lock-fee. Atomic operations must be address-free if used between processes. If an operation is lock-free , it must also be address-free. Sequential consistency is not settled, and can be affected by hardware design. Example: How to program 'total store order'. Experts doubt whether mortals can program effectively without a total store order.

Thread-local storage has been implemented by at least five vendors. Some have slightly different syntax. A new thread duration has been defined. Dynamic initialization and destruction of static-duration variables can be tricky. Without implicit synchronization, there is the potential for data races. With implicit synchronization, there is the potential for deadlock.

The thread semantic model initiates a thread with a fork on a function call (semantically like pthreads). Join waits for the function to return. Some open questions remain. Is there a test for ready for join? Is there a join with a timeout? Does join return a function value?

Mutexes provide mutual exclusion. The Standard (C++) will have a least a simple mutex, and may have read-write and reentrant mutexes. Conditional variables enable the monitor paradigm. Threads may wait on a conditional variable.

Thread termination is voluntary. Return form outermost function, likely to have some form of cooperative termination, possible have some for of synchronous cancellation (exists in Windows) and there is strong opposition to asynchronous cancellation. Thread scheduling is limited.

Locks hold a mutex within a given scope.

Higher level primitives may be built on primitives such as tread pools, thread groups, parallel iterators, etc.

Further threads work is being prepared for inclusion in C++ TR2, which will follow the next revision of C++ scheduled for 2009.

Doug Gwyn suggested that we, WG14, may want to do something different from the approach being taken by C++.

ACTION - Ulrich and a small team to write a liaison report on threads to WG21, address Lawrence Crowl's presentation. JB, Nick, Tom Plum, and Bill Seymour to work on this with Ulrich.

Tom Plum Report #1: Macro Regions Proposal

http://wiki.dinkumware.com/twiki/bin/view/Wg14wg21/HybridSolution

This effort is essentially in the 'stalled' proposal list within the Evolution Working Group (EWG) in C++ (WG21/J16). Tom believes it solves real problems with the preprocessor. There is opposition within this committee to changing the C Standard. PJ and Doug believe we should not encourage further work in this area. Consensus is to do nothing.

Tom Plum Report #2, WG21/N1085 C++ Proposal to augment the interface of malloc, et al.

WG21/N1085 is a proposal by Howard Hinnant. Ulrich wants to add alignment considerations to this C++ proposal. POSIX is now using a function called posix_memalign() that seems suitable for the original proposal. That function came from an X/Open function named memalign(). Doug Gwyn believes a function of this sort would be useful for C as well. Tom pointed out that we are not in the business of revising the C Standard at this time. If C++ drops this, we could revisit. Ulrich expressed concern that C++ might put a different spin on this. PJ prefers that we not push this work on to C++. Sees it as simply adding to layers of complexity. It's not clear whether or not C++ will even ever adopt this. PJ sees it as low on the list of things to do.

Straw Poll: Do we want to reaffirm that WG21 consider N1085 with something along the line of posix_memalign included. 8-2-9. No consensus.

Straw Poll: Do we want to ask WG21 to consider adding something like posix_memalign to C++0X. 15-0-3.

Hedquist pointed out that asking WG21 to add something to C++0X will require submitting a proposal.

Straw poll: Do we want to ask WG21 to consider taking N1085, removed two functions, request_malloc, negotiate_alloc, and add posix_memalign like functionality. 13-1-4 This supercedes the second straw poll above, and will go into our WG21 Liaison Report.

Proposed Words for WG21 Liaison Report:

WG14 urges WG21 to incorporate into the next C++ revision ("C++0x"):

1. The functions sizeof_alloc and resize_alloc from WG14/N1085 ("Proposal to augment the interface of malloc/free/realloc/calloc", by Howard Hinnant); and

2. A function along the lines of posix_memalign - see www.opengroup.org/onlinepubs/000095399/functions/posix_memalign.html

Straw Poll: Include the above words in a Liaison Report to WG21. 15-2-1

Clark Nelson Report. WG21 Core Language issues #370 & #533.

There are a number of issues that CORE would like to have clarified.

What happens when an include directive is used with angle brackets (<>)? See WG21 CORE Issue List, numbers 370, and 533.

#370 Does the use of <…> vs "…" have any special meaning in C99?

Proposed resolution to #370:

Change 16.2 cpp.include paragraph 7 from:

[*Example:* The most common uses of `#include` preprocessing directives are as in the following:

```
    #include <stdio.h>
    #include "myprog.h"
```

—*end example*]

to:

[*Note:* Although an implementation may provide a mechanism for making arbitrary source files available to the `< >` search, in general programmers should use the `< >` form for headers provided with the implementation, and the `" "` form for sources outside the control of the implementation. For instance:

```
    #include <stdio.h>
    #include <unistd.h>
    #include "usefullib.h"
    #include "myprog.h"
```

—*end note*]

#533 – Special treatment for C-style header names

Section: 16.2 cpp.include **Status:** review **Submitter:** Jens Maurer **Date:** 4 October 2005

In language imported directly from the C Standard, 16.2 cpp.include paragraph 5 says,

The implementation provides unique mappings for sequences consisting of one or more *nondigits* (2.10 lex.name) followed by a period (`.`) and a single *nondigit*.

This is clearly intended to support C header names like `stdio.h`. However, C++ has header names like `cstdio` that do not conform to this pattern but still presumably require "unique mappings."

Proposed resolution (April, 2006):

Change 16.2 cpp.include paragraph 5 as indicated:

The implementation provides unique mappings **between the delimited sequence and the external source file name** for sequences consisting of one or more nondigits or digits (2.10 lex.name)**, optionally** followed by a period (`.`) and a single nondigit...

2.5 FSG (Free Standards Group) (Stoughton)

No report for this meeting.

2.6 OWG: Vulnerability (Benito)

Group has met twice. A draft for a base document has started. Plans on having a draft ready for a December meeting in Washington, DC. Everything is public. Will meet again in April 2007, following WG14. We need a liaison from WG14. Tom Plum has volunteered to do so. Tom will be a two way liaison between WG14 and OWG Vulnerability.

2.7 Other Liaisons - none

3. Defect report status (N1191) (Benito)

N1191 is a linked listing of all CLOSED defect reports, categorized as either a Record of Response (RR), or as Technical Corrigenda (TC). JB wants to take the unpublished TC items, put them together as Technical Corrigenda to C99, and ask SC22 to let us republish the Standard.

4. Flexible array member as last member (N1189) (Wakker)

N1189 asks a question regarding a structure containing a flexible array member, and proposing a wording change to the C99 Rationale. Everyone agrees.

ACTION: Convenor to change the C99 Rationale as proposed in N1189.

5. Potential defect reports (N1178, N1181, N1183, N1184, N1186, N1187, WG21 CORE-370, WG21 CORE-533, N1194) (Jones, Plum, Tydeman,

Nelson, Gwyn)

N1178 - Yes, this is s defect. DR 328

N1181 - Yes, this is a defect. DR 329.

N1183 - Yes, this is a defect. DR 330.

N1184 - Yes, this is a defect. DR 331.

N1186 - Straw poll 2-6-9 Not a defect.

N1187 - This is Undefined Behavior. Not a defect.

WG21 CORE-370 - Straw Poll: Is this a defect? 2-9-7 Not a defect.

WG21 CORE-533 - This is a feature, not a defect.

N1194 – Proposes that gets is a defect. The presence of gets is not a defect, but it is worthy of discussion. DR 332.

6. Sequence Points. (N1188) (Plum)

N1188 is a paper that explains the C language sequence point model, and may be suitable as an addition to the C99 rationale. Typo on page 3. 2 to the N should be N factorial. One specific issue is the ordering of an expression such as b++ may be indeterminate. Are there other issues? The difficulty seems to be in determining all the possible orderings of sequence point, which makes teaching it difficult.

Making use of the model here is important in providing liaison input to C++ for their concurrency model. We want to make sure our underlying model does not get broken by C++ concurrency. We have plenty of time to get this in the C99 rationale, but there is more urgency in nailing down our model for C++ concurrency.

7. Decimal Float-point. (N1197) (Kwok)

N1197 is a collection of comments on N1176, DTR 24732, Decimal Floating Point.

GCC Comments
1) Function Return Value. Non issue, covered in C99.

2) xxx_DEN macros. Agree. Change to subnormal

3) Mixed operations between DFP and generic floating types. Suggests treating an implicit conversion between DFP and BFP as a constraint violation. Yes, agree. See section 8 Arithmetic Operations

4) Mixed operations between DFP types and integer type. Agree with suggested change.

5) Conversions between DFP and integer types:

5a) 6.3.1.4. When converting DPF -> integer and the resulting value cannot be represented, should be undefined? No, keep "invalid" exception but make the result unspecified.

5b) Annex F. Align with IEEE 754 for DFP types? No change, leave as is.

5c) Other 754R changes affect all floating types. How to handle? Leave as is. Consider adding a Note that addresses our intent to follow 754R, but it is a moving target a/o Oct 2006.

6) Precision of Formatted Output Specifiers. The precision is encoded in the representation. Should it be used when the precision is not given? The default is 6. The encoded precision may be more than folks really want? Support for recommended changes? 1-15-3. No support, leave as is.

7) Internal representation of TTDT. Suggests removing "recommended practice" to use _Decimal128. Leave it to the implementation. No support to removing the statement? Support for suggested change? 1-9-10 No support, leave as is.

HP Comments

I Document Structure. Rich agrees that the existing structure makes sense, given that the types are not going to change. Leave as is.

II Design Rationale

Rich sees no need for all of the new types used in this TR. Tom pointed out that we discussed this in detail in Kona, but may not have captured the reasoning in the rationale. Agreed to add text to the rationale to explain this.

III Detailed Comments on the text by section (HP)

2.1 Agree to add words showing deferring to IEEE 754R.

3: No change. Leave as is.

6.1 [1a] sentence 3: make it unspecified behavior.

6.1 [2a]: No change.

6.1 [2a], near: There is no overflow for 'near'. Text needs to be changed to make that clear.

6.1 [2a], negative infinity: Change as suggested.

6.1 [2a], near: No change, however, words are needed for the rationale.

6.2: Same set of comments for 6.1 are applicable to 6.2. Agree

7: Agree. Will add words to the rationale.

7.1 If we have TTDT, it should be exact. Otherwise, remove it. No support for change.

9.2: May be right, but no change for now. Once 754R is firmed up, we will know.

9.3: OK – no action required. Should we add more?

9.5 – 9.6: It's not clear what could be added here that would be of any real use. No Change.

ACTION: Rich Peterson to submit paper for additional comments on TR 24732.

Tydeman Comments

1. Section 5 Characteristics of decimal floating types <float.h> (page 10 of 36), [3]: "The values..." should be "The integer values..."

Agree

2. frexp(): Name of function means Fraction and Exponent. For DFP arguments, should it still be done in terms of 2**N, or should it be done in terms of 10**N? If kept in terms of 2**N, the function is nearly useless for DFP.

Agree

3. remquo(): Is quotient modulo 2**N OR 10**N?

There is some use to leaving it as 2 based. Not clear that a 10 based version is needed. Remove it from the TR.

4. Should add to rationale some words about why no general case designations: type domain, radix, precision, e.g., _Flt_2_24, _Cmplx_10_7, _Imag_16_6.

Agree.

ACTION: Edison Kwok and Editorial Group (PJ, Fred, Jim Thomas, Janis Johnson) to revise DTR 24732, and forward to Convenor.

ACTION: Convenor to forward DTR 24732, as revised by the Edison Kwok editorial group, to SC22 for final ballot.

8. TR24731, Part 2, Dynamic Allocation Functions. N1193 (Stoughton)

This document is a proposed Part 2 to TR24731, using dynamic allocation functions to avoid buffer overruns. There is no invention here. All of the functions already existing implementations, particularly in POSIX. The document is organized the same way as Part I. The intent of this document is to provide programmers with an additional mechanism to address issues of buffer overflow, etc., as does Part I. A key difference is that the mechanisms are via dynamic memory allocation, and heavily dependent on malloc. There are some minor changes that need to be made, the document is still a working draft, but it has sufficient substance to be ready for CD registration. Small editorial group to go over the document. Nick Stoughton, JB, PJ, Randy, Martyn, and Ulrich.

ACTION: Stoughton Editorial Group (Stoughton, Benito, PJ Plauger, Randy Meyers, Martyn Lovell, Ulrich Draper) to edit N1193, TR 24731-2, as needed, and forward the revised document to the Convenor.

ACTION: Convenor to forward TR24731-2, as revised by the Stoughton Editorial Group, to SC22 for CD Registration.

9. Defect Reports

We have a number of closed defect reports that are listed as either 'record of response' (RR), or technical corrigenda (TC). We plan to sweep up the RR defect reports, publish a Record of Response document, and collect the remaining TC list at publish a Technical Corrigenda to C99.
A DR Editorial group will go over these reports for accuracy. (Benito, Doug, Fred, Larry, David).

Review of DRs moved to REVIEW in Berlin

DR 322 REVIEW This DR has a proposed TC, and was discussed in Berlin. Austin Group is satisfied with the proposed TC, so is PJ. Move to CLOSED

DR 323 REVIEW Move to CLOSED

DR 324 REVIEW Moved to CLOSED

Review of DRs moved to CLOSED in Berlin

DR 219 CLOSED - RR

DR 236 CLOSED - RR

DR 304 CLOSED - TC

DR 312 CLOSED - TC

DR 317 CLOSED - RR

DR 319 CLOSED - RR

DR 320 CLOSED - TC

DR 321 CLOSED - TC Needs a new macro name to list of macro names in 6.10.8;p2. Notify WG 21.

ACTION: Nick Stoughton to write up a DR 333 to add the macro __STDC_MB_MIGHT_NEQ_WC__ to 6.10.8;p2 See DR 321

ACTION: Tom Plum to notify WG21 as a liaison of a new macro, __STDC_MB_MIGHT_NEQ_WC_ , being added to C99: 6.10.8;p2.

DRs in OPEN Status

DR 298 – Has a proposed TC developed in Berlin. Moved to REVIEW

DR 311 – Discussed in Berlin, has a proposed TC. Doug Gwyn has an issue with the term 'declarator derivation'. Rich wrote this response, but does not remember that term. It is in the Standard. Change 'declarator derivation' to 'declarator type derivation' in proposed TC, last sentence. Moved to REVIEW.

DR 314 – Randy had an action item to write a paper on this DR. He believes the proposed response is wrong. Not done. There is a lack of agreement as to whether or not question 2 is undefined behavior. There does not seem to be specific words in the Standard that make it undefined behavior, even though that may be what we intended. Remains OPEN, Randy's action item is carried over.

DR 315 – This DR was moved from REVIEW to OPEN at the last meeting (Berlin). See WG14 message: http://www.open-std.org/jtc1/sc22/wg14/11100. Our use of the term 'type' does not seem to be consistent enough to answer the question. Promotion rules take the width into account for types that are required to be supported. There is a certain amount of what is required. I-D types do not have any applicable rules spelled out, and the semantics are I-D as well.

Tom Plum proposed the following new response:

Committee Discussion:

Question 1: Must bit-fields of type char nevertheless have the same signedness as ordinary objects of type char, and similarly for those of types short (or short int), long (or long int), long long (or long long int)?

These would all be implementation-defined, determined by the implementation-defined behavior specified in 6.7.2.1#4.

Question 2: But what should sizeof(x.a + x.b) evaluate to, when (x.a + x.b) has such a bit-field type which does not occupy an integer number of bytes?

In the example presented above, this would be implementation-defined, determined by the implementation-defined behavior specified in 6.7.2.1#4.

Suggested Technical Corrigendum:

Add the words "(and bit-field width)" as follows:

6.3.1.1 Boolean, characters, and integers (add after paragraph 1).

2 The following may be used in an expression wherever an int or unsigned int may be used:

- An object or expression with an integer type whose integer conversion rank is less than or equal to the rank of int and unsigned int.

- A bit-field of type _Bool, int, signed int, or unsigned int.

If an int can represent all values of the original type (and bit-field width),

----- end of Tom's proposed response -----

Further discussion on Tom's input. Intent is to include 'width' as part of the promotion rule. PJ suggested a change to the last sentence to read:

If an int can represent all values of the original type (as restricted by the width, for a bit-field)

Everyone accepted, leave OPEN.

DR 325 – In Berlin, the committee decided that there was no need to force the change requested on all environments, and that doing could be viewed as adding a new requirement. Doug Gwyn believes the response is too terse.
ACTION: Doug Gwyn to propose a modified response to DR 325. DONE

Proposed response:
The intention is to allow implementation to decide what form of message is appropriate. There is no consensus for imposing additional requirements.

Moved to REVIEW.

DR 326 – Moved to REVIEW (TC). Editorial fix for singular/plural consistency needed.

DR 327 – This DR was generated to respond to issues that came up in the discussion of DR 311. Make the Suggested TC the Proposed TC deleting the portion referred to by 'it might also be desirable to change…' . Moved to REVIEW.

DRs new to this meeting.

DR 328 – N1178 String literals in compound literal initialization. A compound literal has static storage duration. Clark wants 6.5.2.5;p3 removed completely rather than adopting the suggested TC, because it is redundant. Do we want to remove the written constraint from the text even if it is redundant? A suggestion is to remove the paragraph, and add a footnote pointing out that constraints applicable elsewhere (6.5.2.5;p2, 3, &7) are applicable here. Clark also pointed out that 6.5.2.5;p2 is also redundant. Doug Gwyn believes that the 'fewest' changes is the best policy.

Straw poll:
1) Remove the redundancies and add a place holder -10
2) Take the TC as suggested - 5
Abstain 2

If we do 1), Nick suggests adding a constraint by adding para 7 to Constraints but with the word "semantic" removed. Modify para 7, in place, by removing "constraints".

Straw poll: Adopt Nick's suggestion: 16-0-2

Replace para 2 & 3 with the words "All the constraints for initializer lists in 6.7.8 are applicable to compound literals."

Reword para 7: "All the semantics rules for initializer lists in 6.7.8 are applicable to compound literals."

Footnote 82 (to para 7) remains in place.
Status remains OPEN

DR 329 – N1181 Math functions and direct rounding. Footnote 204, 7.12.10.2;p2, is wrong. The corrections proposed look like new requirements. Fred agrees. Moving the proposed changes to Annex F might satisfy the matter. The footnote needs to be fixed. The sentence "Thus, the remainder is always exact." should be removed from the footnote. The remaining changes should be added to Annex F.

ACTION: Fred Tydeman to modify the Suggested TC for DR 329. DONE

DR 330 – N1183 Externally visible exception conditions. The text of 7.12.1;p1 seems to imply that errno cannot be altered by any math function, nor can any of the floating-point exceptions: invalid, divide-by-zero, overflow, underflow. Suggested TC: 7.12.1;p1, last sentence: insert spurious , so it reads "… without generating any spurious externally visible exceptional conditions." No objection to this change. Moved to REVIEW.

DR 331 – N1184 Permit FE_DIVBYZERO when errno says EDOM. According to 7.12.1;p2, when errno and FP exceptions are used, and a domain error occurs, errno gets set to EDOM and the FP Exception is set to FP_INVALID. This DR proposes to add FE_DIVBYZERO as an acceptable return. The DR needs a Suggested TC. PJ volunteered to write one up.

ACTION: PJ to write up a Suggested TC for DR 331 (FE_DIVBYZERO). DONE

After further review, PJ concluded that this is not a defect, and concluded that no TC needs to be written. Leave OPEN

DR 332 – N1194 gets. gets has known problems with buffer overflow, and its existence is a continuous source of criticism for the C Standard. Doug Gwyn proposes to add a change to how the buffer is handled for gets. Other possible solutions include coming up with a recommended practice, deprecating gets, etc., etc. Changing the BUFFSIZ behavior means changing source code. Martyn suggests deprecating. Ulrich does not want a change to how BUFSIZ works.

Further, he believes it is time to consider revising the C Standard. Don't try to 'fix' the problem, that's unworkable. PJ prefers to see a TC that deprecates gets, with strong words telling folks to never use this function. POSIX is making gets "obsolete". An implementation must provide it, but applications are forbidden from using it. Doug Gwyn proposed adding gets to a list of Deprecated Functions in the Standard. JB proposed adding gets to 7.26, Future Library as a deprecated function. PJ suggests making it obsolescent as well. ISO/IEC Directives support our desire to deprecate gets.

ACTION: Doug Gwyn to draft a Suggested TC to deprecate gets for DR 332 DONE

Discussion: The committee thinks that the programming community would be better served by flagging the gets function as deprecated.

Proposed Technical Corrigenda
Add to subclause to 7.26.9:
The gets function is obsolescent, and is deprecated.
[Note: Rationale wording might be useful.]
[Editorial Note: Add a forward reference to this from gets 7.19.7.7]

Moved to REVIEW

DR 333 – Missing Predefined Macro Name. This DR was generated from the discussion on DR 321, where we realized that the new macro __STDC_MB_MIGHT_NEQ_WC__ needed to be listed with conditionally defined macros in 6.10.8;p2. In the Proposed TC, the sentence that reads: "…a member of the basic character set need not have a code value equal to its value …", it is not clear what "its" refers to. This DR is tied to DR 331.

Revised Suggested TC to read:

Add, in proper alphabetic order in the list:

__STDC_MB_MIGHT_NEQ_WC__ The integer constant 1, intended to indicate that, in the encoding for wchar_t, a member of the basic character set need not have a code value equal to its value when used as the lone character in an integer character constant.

Moved to REVIEW.

10. Future of C - Ulrich Drepper

Ulrich believes that the state of C compiler development is such that the C Standard is well behind the technology being used by the community. Virtually all major C compiler developers have developed extensions to the language that go well beyond the Standard. We can either subsume ourselves to C++, or plan on revising the C Standard to adopt existing technologies. PJ pointed out that our pace has been deliberate, and that adoption of C99 has been slow. We probably should consider reopening the Standard soon, and look at adopting things like multi-treading, security features, and others. Ulrich believes we should focus on existing practice, features that are in wide use, minimize the risk of standardizing features that no one will use. Doug Gwyn believes that C will have longevity in embedded programming, but that if we work on a 5 year schedule we should probably consider starting now. There are a number of things that C can probably do better than other languages. Round table discussion on whether or not we should consider revising the C Standard. David believes that are real commercial needs that C can address, such as security. General feeling that making a decision to revise the C Standard with a focus on existing practice would be a good thing. Many developers make use of extensions to the language, some do not. Invention of new feature sets is not a good idea. JB: No one is saying no. If we are going to do this we will need to work on our charter, and proceed from there.

The next step.
JB: We need to start putting a charter together that will define the scope a revision to the C Standard. Tom is in favor of using the Wiki as a vehicle to propose ideas. PJ suggests using a full day in London to process such a list. Start by generating a list, filter it based on criteria, then let the world know what we are doing. We are not ruling out proposals from the outside.

11. Administration

Berlin: TR 18037, Embedded TR. We tried to get this document republished, but for some reason, nothing happened.
ACTION: Convenor to forward TR 18037 for publication, again.

11.1 Future Meetings
11.1.1 Future Meeting Schedule – see above.

2007 – Spring, London following C++. April 23-27, 2007. Specific location is TBD.

2007 – Fall, Kona. Oct 8-11, 2007.

2008 – Spring. Invitation extended by the Netherlands, NEN/ACE. We will accept. April 14-18.

2008 – Fall, Denver CO, hosted by Cisco.

11.1.2 Future Agenda Items

No new items.

11.1.3 Future Mailings

Post Portland mailing items to JB by 27 November, 2006

Pre London mailing items to JB by 26 March, 2007

11.2 Resolutions / Votes

11.2.1 Review of Decisions Reached

Sweep up DRs that are CLOSED, ready to publish in a TC, and create TC-3.

Sweep up DRs that have a response and publish as a Record of Response document.

11.2.2 Formal Vote on Resolutions

None.

11.2.3 Review of Action Items

HELD OVER ACTION ITEMS

ACTION: Randy to write a paper with a proposed response for DR 314.

ACTION: Randy to add words to the rationale for TR 24731 to better clarify the issues raised in N1160.

ACTION: Convenor to forward TR 18037, Support for Embedded Systems, to SC22 for publication, again.

NEW ACTION ITEMS

ACTION: Ulrich Drepper and a small team to write a liaison report on threads to WG21, in response to Lawrence Crowl's presentation, N1196. JB, Nick, Tom Plum, and Bill Seymour to work on this with Ulrich.

ACTION: Stoughton Editorial Group (Stoughton, Benito, PJ Plauger, Randy Meyers, Martyn Lovell, Ulrich Drepper) to edit N1193, TR 24731-2, as needed, and forward the revised document to the Convenor.

ACTION: Convenor to forward TR24731-2, as revised by the Stoughton Editorial Group, to SC22 for CD Registration.

ACTION: Convenor to change the C99 Rationale as proposed in N1189.

ACTION: Tom Plum to notify WG21 as a liaison of a new macro, __STDC_MB_MIGHT_NEQ_WC_ , being added to C99: 6.10.8;p2.

ACTION: Edison Kwok and Editorial Group (PJ, Fred, Jim Thomas, Janis Johnson) to revise DTR 24732, and forward to Convenor.

ACTION: Convenor to forward DTR 24732, as revised by the Edison Kwok editorial group, to SC22 for DTR ballot.

ACTION: Rich Peterson to submit paper for additional comments on TR 24732.

12. Adjournment

Thank to the Host, Intel, for getting a great meeting facility.

Thanks to Dave Keaton for Chairing the DR review.

Thanks to Dinkumware for the Wiki.

Meeting adjourned at 11:10 AM, 26 October 2006.

Agenda for the J11/U.S. TAG Meeting, Tuesday October 24th at 13:30

Attendees

John Benito	Blue Pilot	
Randy Meyers	Silverhill Systems	J11 Chair
Douglas Walls	Sun Microsystems	J11 IR
Fred Tydeman	Tydeman Consulting	J11 Vice Chair
Barry Hedquist	Perennial	Secretary
Cecilia Galvan	Freescale	
P. J. Plauger	Dinkumware, Ltd	
Tana L. Plauger	Dinkumware, Ltd	
Mark Terrel	Cisco	
Nick Stoughton	Usenix	
John Parks	Intel	
Clark Nelson	Intel	
Bill Seymour	self	
Martyn Lovell	Microsoft	

Tom Plum	Plum Hall
Christopher Walker	Dinkumware, Ltd
Doug Gwyn	US Army
Lawrence Crowl	Google
Rich Peterson	Hewlett Packard
Don Goham	Cray
Janis Johnson	IBM
David Keaton	self
Edison Kwok	IBM

1. Select US delegation for the next two meetings.
Motion to approve (Benito, Tydeman) 14-0-0.

Douglas Walls – HOD
John Parks
Nick Stoughton

2. 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.

3. Anti Trust

INCITS J11 members are reminded of the requirement to follow the INCITS Anti-Trust Guidelines which can be viewed at http://www.incits.org/inatrust.htm.

4. Special Math functions open issues(N1185) (Benito)

N1185 is a WG14 document containing comments on DTR 24747, Special Math. The purpose of this discussion is to review this document w/r/t establishing a US Position. This document is currently in registration ballot. The comments are very technical in nature, and not likely to be appropriate for a registration ballot.

Douglas asked when is the appropriate time to object to the "type" a TR is being processed as. The best time is when the New Work Item is processed. The NWI establishes the scope of the project.

N1185 is missing a rationale. This needs to be taken care of. See WG21 N1422, N1514 and N1542 for a starting point on a Rationale. These are the initial, and follow-on proposals from Fermi Labs.

Comments were submitted by two parties, Tydeman, and Sun Microsystems. Both agree to allow the ballot to be reconsidered, and forwarded without their comments. Several of the comments may be handled as submitting a Defect

Report on C++ TR1. PJ agreed to bring any submissions to the Library Working Group at WG21.

Motion: Reconsider the registration ballot for DTR 24747. (Plauger, Tydeman) No Objections.

Motion: J11 recommends the U.S. vote to support the proposal that document SC 22 N4109 be registered as ISO/IEC PDTR 24747, Information technology - Programming languages, environments and system software interfaces - Extensions to the C Library, to Support Mathematical Special Functions without the comments contained in N1185. (Plauger, Tydeman) (12-0-3) Motion Passes.

5. Adjourn at 15:00