DRAFT MINUTES FOR 27-30 March 2006
MEETING OF ISO/IEC JTC1 SC22/WG14 AND INCITS J11

WG14/N1168

Meeting Times

Monday:	27 March 2006 09:30-12:00 13:00-16:00
Tuesday:	28 March 2006 09:00-12:00 13:00-17:00
Wednesday:	29 March 2006 09:00-12:00 13:00-17:00
Thursday:	30 March 2006 09:00-12:00 13:00-17:00
Friday:	31 March 2006 09:00-12:00

Meeting Location:

DIN German Institute for Standardization
Burgrafenstraße 6
10787 Berlin
GERMANY

Host:

German Institute for Standardization

Host Contact information:

Cord Wischhöfer
DIN Deutsches Institut für Normung e. V.
Normenausschuss Informationstechnik (NI)
Information Technology Standards Committee
10772 Berlin
Deutschland/Germany

E-Mail: Cord.Wischofer@din.de

Meeting / venue information: N1136

1. Opening activities

1.1 Opening Comments (Wischofer, Benito)

Cord Wischofer welcomed us to DIN and Berlin. Lunch today is scheduled for 12:15 - 13:30 at the Crown Plaza, and is provided by SAP. Remaining lunches will be provided by SAP, and brought to the meeting room. DIN rules require that we exit the building no later than 18:00, 18:30 the latest.

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	
David Keaton	self	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	
Robert C. Seacord	CMU/SEI	USA	
David Schwab	Oracle	USA	
Rich Peterson	HP	USA	
Bill Seymour	self	USA	
Martyn Lovell	Microsoft	USA	
Tom Plum	Plum Hall	USA	
Hal Burch	CMU/SEI	USA	
Christopher Walker	Dinkumware, Ltd	USA	
Noubo Mori	SAP	Germany	HOD
Edison Kwok	IBM	CANADA	HOD
Francis Glassborow	self/Plum Hall	UK/USA	UK HOD
Derek Jones	Knowledge Software	UK	
Willem Wakker	ACE	Netherlands	HOD
Randy Marques	Atos Origin	Netherlands	

1.3 Selection of Meeting Chair (Benito)

John Benito - Meeting Chair
Barry Hedquist - Meeting Secretary

1.4 Procedures for this Meeting (Benito)

The Chair 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.

1.5 Approval of Previous Minutes, (N1145) (Hedquist)

Fred has comments.
1.2 Intro of Participants:
Is Nick Stoughton FSG and/or Usenix Association?
Both. Nick presented an FSG Liaison Report.

1.10 Identification of National Bodies:
"Six" should be "Five".
Corrected

8. Specification for Managed Strings:
3rd para: "... to allocate a very large buffer top the point ..." ???
Corrected

10. TR24731:
6.6.1.4 strict_handler_s:
First question has no answer. The "Yes." after the second question is ambiguous.
Corrected

TR24731 - The Name of the TR (Plum):
2nd paragraph: "... and the only SC22 ..." should be "... and that only SC22 ...".
Corrected

12. Defect Reports:
Open DRs:
DR 321: __STDC_MB_MIGHT_NEG_WC__: Perhaps NEG is NEQ?
Corrected

DR 322: "...we cannot says what...": says -> say.
Corrected

J11 TAG minutes:
Extra spaces before "USA" in Tydeman's entry.
Format / PDF issue

Cecilia Galvan is Freescale (not Metrowerks).
Corrected

Approved as amended. Document is N1166

1.6 Review of Previous Action Items and Resolutions (Hedquist)

1. ACTION - Convenor and PJ to come up with words to add to Rationale addressing issue #3 in N1094. DONE

2. ACTION - Convenor to request establishing a liaison with the SC22 POSIX Advisory Group. DONE

3. ACTION: Request that the Convenor ask SC22 for a one year extension for TR 24732 - DONE

4. ACTION: TR 24731 Rationale. Randy to collaborate with Nick to add a paragraph to 6.7.3.1. strtok_s - DONE

5. ACTION: TR 24731 Rationale. Randy to add words on the Committees decision regarding the use of a handler. DONE

6. ACTION: TR 24731 Rationale. Nick to add words & send to Randy regarding fgets to Rationale 6.5.4.1 - DONE

7. ACTION: TR 24731 Rationale. Randy to add some words on the frequency of attacks related to buffer overflow. DONE

8. ACTION: Rich Peterson and/or John Parks to propose words for DR 311. - OPEN

9. ACTION: Tom Plum to propose a response to DR 315 based in where C++ is going. OPEN

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

11. ACTION: Tom and Randy to propose words for DR 324. DONE

12. ACTION: Fred and Edison. Write a rational for the Decimal FP TR. DONE

1.7 Approval of Agenda (Benito) (N1156)

Add a day, 31 March, 09:00 - 12:00

Correct prior minutes, N1145 vice N1146.

MOTION: Agenda approved as modified (no objection)

1.8 Distribution of New Documents

New documents are posted to the Wiki, two potential DRs, N1165 strerror, a paper from Bill Seymore, N1164.

1.9 Information on Next Meeting (Portland) (N1153)

Next meeting will be in Portland, OR, hosted by ANSI and Intel Corporation.

WG21 / J16 - Sunday, Oct 15 - Friday, Oct 20
WG14 / J11 - Monday Oct 23 - Fri, Oct 27.

1.10 Identification of National Bodies (Benito)

5 countries represented:

Germany
UK
Canada
USA
Netherlands

1.11 Identification of J11 voting members (Walls)

17 voting members out of 19 possible. (See attached J11 membership list for attendees.)

2. Liaison Activities

2.1 INCITS/J11 (Walls, Meyers)

Nothing to report.

2.2 SC22/WG11 (Wakker)

WG11 met in New York, 5-6, October, 2005. Next meeting is not yet scheduled.

2.3 SC22/WG14 (Benito)

John Benito was renamed as Convenor for WG14 at SC22 plenary last October, at Mont Tremblant. Extensions given to two TRs, name change on another. Next SC22 plenary is in London at BSI. A new type of working group, called “Other” was created, called OWG/Vulnerabilities. Jim Moore, Ada convenor, is the Convenor. N1163 describes the group.

2.4 J16/WG21 (Plum)

No report.

2.5 FSG - Free Standards Group (Stoughton)

The LSB, having passed its ISO ballot, is still awaiting final publication from ISO. There has been a minor hold-up as the Project Editor has not had sufficient bandwidth to deal with copyright issues. The roadblock has been cleared, but the editor has not yet implemented the agreed changes.

2.6 POSIX / Austin Group (Stoughton) (N1159)

N1159, a report for POSIX/Austin Group. Met one month ago in Ottawa. POSIX Standard is open for a full revision to incorporate a backlog of defect reports, clarify issues, and add new features. They would like closure from us on three of our open defects: DR321, DR322, and DR323.

2.6 OWG - Vulnerability (Benito)

Already covered in SC22 report, item 2.3 above.

2.7 Other

IEEE FP (754) is finally making progress. They meet every month.

3. Defect report status (Benito) (N1142)

N1142 lists a status of CLOSED DRs

4. Potential Defect Status Reports. (Benito) (N1152)

N1152. This paper is a proposal for a new string literal lexem, pp-string-literal, by Ivan A. Kosarev, Unicals Group, Russia.

This is really a proposal for a feature, rather than a defect. Tom suggested that we postpone this until Randy has wording for DR324 from the same author, since it addresses essentially the same issues. See N1157.

No document – Fred noticed a potential problem with a function (logb) handling of a non-normalized number. PJ believes the wording is fine as it is. Randy believes the wording could be ‘nudged’, but the formula is correct. Tom suggested adding words to the rationale. No consensus on making this a defect.

No document – Tom Plum. Corner case: dividing by zero, conflict between Clause 7, and App F. Email to SC22WG14 reflector. Fred believes this should be this item should be a DR. Moved to DR agenda.

Strerror.htm – Nick Stoughton / Austin Group. Make a DR325

AI-053.txt – Nick Stoughton / Austin Group. asctime. Make a DR326

5. Threads (N1164) (Seymour)

Bill Seymour presented a paper on threads proposals being considered by the Evolution Working Group within the C++ Standards Committee (SC22/WG21 and INCITS/J16), for possible inclusion in the next revision to the C++ Standard, C++0X. Bill believes that compatibility between C and C++ is essential, and offered to take any input we have to the C++ Evolution Working Group meeting next week. Do we have any 'over my dead body' issues with how WG21 proceeds? Not at this time.

Basically, three types of implementations are proposed for C++.

1) An implementation that revises the language, and adds threads to the library.

2) Boost Threads, with a C interface.

3) POSIX threads, also known as pthreads.

Nick pointed out that pthreads is an existing standard, and has been implemented. However, the same can be said of the Boost threads.

Tom suggested that the issue of sequence points is an important consideration, and that once an approach is selected by WG21 (C++), we will want to examine the approach with that in mind. Randy pointed out that all of this is doable, and has been done.

We cannot pick a solution, but agree that we need to keep an eye on whatever C++ decides to do. We believe that we have sufficient liaison coverage to track what C++ is doing, and bringing C++ papers forward.

6. Specification for Managed Strings (N1158) (Seacord)

This paper is an extensive revision of a paper presented at Mont Tremblant (N1132). It is proposed Technical Report intended to address vulnerabilities that occur through the manipulation of strings using C functions. These vulnerabilities include errors created by buffer overflow, string termination, string truncation, and improper data sanitization. It proposes a set of secure string libraries, and a sample implementation exists. This approach uses dynamic memory allocation rather that leaving the allocation to the user as done in TR23731, Part1. However, the Part 1 approach means that users could err in allocating memory, and still create buffer overflows. The dynamic memory approach prevents that from happening.

The WDTR designation on the cover sheet (WDTR N1158) is wrong. There is no WDTR designator. It is simply a WG14 document, number N1158. That can be removed in the next revision, if there is one.

Nick has also been tasked to present a paper, which makes use of existing functions. There is some overlap between the approach that Nick is taking and that covered in Robert's paper, with some memory management functions included. All of the functions either already exist in POSIX, or will be included, in the next revision of POSIX.

Both Fred and Douglas have numerous comments on this paper.

Douglas comments:

1) What does C style string mean? Null terminated byte string.

2) Any maximum string length? TBS

3) 6.4 – How does an implementation determine that memory cannot be allocated? TBS

4) 7.1.1 – isnull_m description needs work. Document is missing a definition of null string.

Fred has a mix of general, technical, and editorial comments. The Convenor asked Fred to post his comments on the Wiki, and that we focus first on general comments. We've not really decided whether or not we want to proceed with this paper.

A sample implementation does not yet implement everything in this paper.

Where do we want to go with this paper? Many believe the approach is a good, but are concerned about whether or not implementers will agree. It really depends on whether or not a market even exists.

7. Max significant decimal digits. (N1151) (Benito)

This paper is based on a proposal to WG21/J16 (C++) to add designations for the maximum number of significant decimal digits to the C++ Standard Library, proposing to do the same for the C Standard Library. It is a revision. Randy suggested adding it to the Decimal FP TR. Edison Kwok has no problem adding it. No objection to doing so. Fred has some comments on the paper, including that it needs a formula.

ACTION: Fred to work with Edison to write up corrections to the N1151. DONE - Fred’s comments are below:

Comments on N1151 Max significant decimal digits macros

- Formulas are inconsistent: First has 3010/1000, second has 3010/10000.

- It would help to add floor() to formulas.

- #define's Formula suffers from overflow on 16-int systems.

- Assuming this magic value is log10(2), Either use 301/1000 or 30103uL/100000uL. Better would be to state the mathematical formula for a general radix [see below].

- Places where .3 and .301 produce different values: 103, 113, 123, 133, 143, 153, 163, 173, 183, 193, 203, 206, 213, 216, 223, 226, 233, 236, 243, 246, 253, 256, 263, 266, 273, 276, 283, 286, 293, 296, ...

- Places where .301 and .30103 produce different values: 196, 299, 392, 495, 588, 598, 681, 691, 784, 794, 877, 887, 897, 980, 990, ...

- C99 already has DECIMAL_DIG as ceil (1+precision*log10(radix)).

8. TR 24731 (N1146, N1147, N1160) (Meyers, Stoughton)

N1146 - Latest version of TR 24731 (Oct 25, 2005)

N1147 - Latest rationale for TR 24731 (Oct 2005)

N1160 - Austin Group concerns on PDTR 24731.

Randy pointed out that he has not received any comments on N1146, N1147, other than those submitted in N1160 by Austin Group. If we can get this document out this week, there is a good chance we can get published this year.

N1160 Discussion. Comments are compiled from those submitted by Austin Group members, and center on:

- ABI Changes are expensive, and will be required to adopt this TR. Comments: Randy pointed out that’s true, but it is also true for the Part II approach, or with any approach that tries to solve the problems. That is a cost for becoming more secure. That cost is higher when dealing with buffers of unknown size, but those buffers are exactly the ones that should be changed. What ABIs ? A third function that calls a new ‘_s’ function. But, rewriting the call changes the ABI anyway. There are no ‘existing’ ABIs being broken.

- Use of exception handling thru runtime constraints. Concern is that errors can be ignored, and therefore introduce security holes. Comments: Mark believes the approach taken is practical, providing better options than what already exists. PJ also believes that getting past the existing model of simply making it undefined behavior is a step up. There was one large company that made it clear to Microsoft that they wanted the ability to ignore errors.

- A handler that returns, rather than takes action, increases the probability that programmers will mishandle the return values, and introduce new problems. Comments: No better solution is proposed. While it is recognized that such may be the case, it is a recognized cost. Generating returns, even if they are ignored, is certainly better than doing nothing. Willem suggested possible text that can be added to the rationale

- The drop-in-copy technique can introduce changes to the behavior of the program with a latent undetected bug that can create errors, obviating the savings that are expected to be achieved. Comments: The problem could be even worse with using dynamic allocation. Any change whatever may cause a surprising change in a program with a latent bug. Should we better address the raised issues in the rationale or in a response to this paper, or both? Both.

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

ACTION: Randy to add words to the rationale to address concerns raised in N1160.

9. TR 24732 (N1154, N1161) (Kwok)

N1154 - latest version of TR 24732, decimal floating point arithmetic, Feb 27, 2006

N1161 - latest version of the rationale, undated.

Noubo raised the issue of the physical format - that the decimal representation is stored in one system of one encoding, and how it may be changed if moved to another system using a different character encoding. Will IEEE required two different representations, and if so, will C do likewise? The problem of converting the representation of a floating point value from one system to another is easier in decimal than in binary. Issue is unresolved.

Edison walked us through N1154.

- Page 7, Issue of how the types are named. There is a general consensus to follow the model we used for complex, and fixed point, that allows the use of underscore capital letters as keywords while also providing syntactic sugar that yields ‘nice’ type names. That’s a two step process. PJ suggested that we consider establishing the pattern now to anticipate the predicted success of decimal floating point, i.e. use a one-step process, but is satisfied with it as is.

- Page 7, Do we need all the values provided for DEC_EVAL_METHOD? Yes. ACTION: Edison to add words to the rationale to better explain why we need all the values provided for DEC_EVAL_METHOD.
- Page 7, Fred believes the values for min / max exponent are off by one. Edison will check.
- Page 8, Fred, 754R requires an exception raised when converting FP to integer, and it does not fit.
- Page 9, Fred, converting from integer to FP that overflows, the result is either infinity or the maximum finite number, depending on the rounding mode.
- 6.2, page 9, Douglas, new type should be promoted type
- 6.2, page 10, ‘ulp’ should be defined. Agree.
- 6.4, page 10. Should violation be a constraint violation, i.e. required to generate a diagnostic. Yes. Add to Sec 9 in document for Constraints for each operator.
- 7.1.1, page 14, unsuffixed types. The present Standard treats unsuffixed FP data types as double, this changes that. Make it user selectable? No. It is not our intent to change existing programs. Anyone wishing to change this should submit words.
- 7.1.1, Page 13, ‘fixed point’ types mentioned. There needs to be a reference to the other TR (Embedded – Fixed Point).
- 8, Add default rounding mode. FE_DEC_TONEAREST
- 8.1, page 16. Remove this section.
- 9, Constraint violations mentioned earlier need to be added here.
- 10.3, Use D, Dl, and DL as conversion specifiers.
- 10.4, SNAN is not needed.
- 10.5, page 29, SNAN not needed.

Editing group formed: Edison, Fred, David, PJ to review changes, then forward document to SC22 for Registration Ballot.

ACTION: Convenor to ask Sally (ANSI) to distribute the Rationale during Registration Ballot.

10. Defect Reports (N1157, et al) (Meyers, Galvan)

Cecilia Galvan Chaired the DR portion of the meeting.

N1157 - DR 324. Tokenization obscurities. Randy walked through the response proposed by him, and Tom Plum. Willem suggested modifying the example as well. Doing so, might create even more confusion. Add a footnote in the proposed TC that points to the existing example, as a means to clarify.
Randy will produce a new version during this session.

N1152 – New lexem for string-literal. This paper addresses issues similar to those in N1157, and essentially proposes that the preprocessor token ‘string-literal’ be replaced by ‘pp-string-literal’, and that new grammar be added that distinguishes the two. The discussion focused on whether or not such a change is needed, or warranted. The proposal essentially asks us to better define what happens with an implementation that accepts undefined behavior (a header name containing a backslash in a pp directive).

ACTION: Tom Plum to write up a proposed response to N1152. DONE

Tom provided the following draft response:

The committee appreciated the very interesting discussion.

The C standard is not being revised at this time, so we cannot consider major substantive revisions.

Moreover, there are some technical reasons why the more permissive rules governing "header-name" should not be permitted outside of #include and #pragma. The contexts of #include and #pragma are syntactically very restrictive; the parser can determine the end of a q-char-sequence because a newline is reached. But in any other "middle-of-a-line" context the lexer should have only one syntax to recognize.

We believe there is a small technical error in the proposal; the syntax for pp-s-char-sequence would not permit an embedded backslash-doublequote within an ordinary string literal.

Syntax
[#1] pp-string-literal:
" pp-s-char-sequence opt "
L" pp-s-char-sequence opt "

pp-s-char-sequence:
pp-s-char
pp-s-char-sequence pp-s-char

pp-s-char:
any member of the source character set except the double-quote ", new-line character or backslash \ followed by a new-line character.

===== end of draft response =====

David pointed out that Annex A may also have an error.
Tom suggested that we shorten his draft proposal to, essentially, we are not considering revisions to the Standard.

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

DRs in REVIEW Status

DR 219 - Moved to CLOSED

DR 236 - Moved to CLOSED

DR 304 – Moved to CLOSED

DR 312 - Moved to CLOSED

DR 317 - Moved to CLOSED

DR 319 - Moved to CLOSED

DR 320 - Moved to CLOSED

DR 321 - There is a typo: NEG should be NEQ. Move to CLOSED.

DRs in OPEN status

DR 298

In Discussion, change “The macros were…” to

“The macros were not intended to be very smart. It is permissible for them to use compiler magic.”

Keep as OPEN

DR 311
ACTION: Rich Peterson to propose words for DR 311. - DONE

Reviewed Rich’s write up of Committee Discussion and a proposed TC. The proposed TC is:

Change 3rd sentence in 6.7.5p3 from:

If the nested sequence of declarators in a full declarator contains a variable length array type, the type specified by the full declarator is said to be *variably modified*.

to

If in the nested sequence of declarators in a full declarator there is a declarator specifying a variable length array type, the type specified by the full declarator is said to be *variably modified*. Furthermore, any type derived by declarator syntax from a variably modified type is itself variably modified.

The write up also contains a discussion of additional issues that would results in an additional DR.

Leave OPEN.

ACTION – Rich to write up Proposed Response to DR 311.
ACTION - Rich to write up a new DR that addresses the Discussion issues.

DR 314

Remains OPEN (Randy ACTION)

DR 315

Discussion:
Question 1 – Response is unchanged.
Question 2 – Different implementations return a different value. EDG always return 8. Sun won't accept the input. GCC 4.0 returns either 4 or 8. It seems that we did not specify this quite as well as we should have. Tom may have been mistaken about this being an issue with C++.

We reopened the discussion Thursday morning. Tom presented a revised response to this DR:

Consider this relevant citation:

6.3.1.1 **Boolean, characters, and integers**

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, the value is converted to an **int**; otherwise, it is converted to an **unsigned int**. These are called the *integer promotions*.[48] All other types are unchanged by the integer promotions.

[48] The integer promotions are applied only: as part of the usual arithmetic conversions, to certain argument expressions, to the operands of the unary **+**, **-**, and **~** operators, and to both operands of the shift operators, as specified by their respective subclauses.

This citation supports the committee’s view that the type of a bit-field does not include the width; if int:7 were a type that is different from int , then these rules would not apply to any bit-field type, an obviously incorrect conclusion.

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 are all 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, the type is long long, and it does occupy an integral number of bytes. The expression evaluates to sizeof (long long).

Typo: The ‘long long’ should be ‘unsigned long long’.

Moved to REVIEW.

DR 322

Agree that the approach proposed in Mont Tremblant is viable.
ACTION: Nick to write up for review, so we can then move it to REVIEW – DONE
Reviewed Nick’s write up, moved the Suggested TC to Proposed TC.
Moved to REVIEW

DR 323 – Value of macro “I” for complex.

Douglas pointed out that the proposed change is does not change the ABI, and is not one that is expensive to implement. Possible suggestion is for POSIX to add the definition as an extension. However, an extension is not supposed to break the Standard, and this does. Randy rethought this issue, again, and we reached a new consensus. Undo DR207.

ACTION: Fred to write up the approach. DONE.

Fred’s write up proposes three possible approaches:

1. The smallest one is back out parts of DR 207 so that C99 allows Annex G implementators to define the **imaginary** macro and have **I** to **_Imaginary_I** and still conform.
2. The middle one is restore C99 back to the state before DR 207 was applied.
3. The hardest one is to re-process DR 207 and look at the suggested alternate changes it has (and even other imaginary issues identified).

JB pointed out that another option is ‘do nothing’.

Discussion on whether or not we want to say anything about the affect on the ABI, and whether or not the effect is significant. Tom support removing the discussion, since he originally brought it up.

Randy believes that we should listen to what POSIX is saying. Nick believes that #1 of Fred’s list would be best for POSIX.

Straw poll on which approach to use.

1. Do Nothing – 0
2. Option 1 – 15
3. Option 2 – 0
4. Option 3 – 1 (Fred)

ACTION: Fred to write up a proposed TC for this DR – DONE

The proposed TC is:

In 7.3.1 of C99+TC1+TC2, replace paragraphs 3 and 4 with:

> [#3] The macros
>
> **`imaginary`**
>
> and
>
> **`_Imaginary_I`**
>
> are defined if and only if the implementation supports imaginary types;[165] if defined, they expand to **`_Imaginary`** and a constant expression of type **`const float _Imaginary`** with the value of the imaginary unit.
>
> [#4] The macro
>
> **`I`**
>
> expands to **`_Imaginary_I`** or **`_Complex_I`**. If **`_Imaginary_I`** is not defined, **`I`** shall expand to **`_Complex_I`**.
>
> [#5] Notwithstanding the provisions of subclause 7.1.3, a program may undefine and perhaps then redefine the macros **`complex`**, **`imaginary`** and **`I`**.
>
> [165] A specification for imaginary types is in informative annex G.

Moved to REVIEW.

DR 324 - Tokenization obscurities.

Proposed new words are on the Wiki, adding a reference to a new footnote.
Moved to REVIEW

NEW DRs

DR 325 – Is an implementation permitted to return an empty string for strerror? The C Standard strongly implies that they cannot be empty.

The issue occurs when strerror is given an unknown errno, it doesn't know what to do. Can it return an empty string? POSIX would like us specify that something be returned, but we see no reason to force this on all environments. In one sense, doing so would add a new requirement to the Standard.

Conclusion: The Committee declines to make any change in this area.
Status: OPEN

DR 326 – asctime called with a tm structure, tm_year, greater than 9999.
The DR appears to be a duplicate of DR 217, on which no action was taken. It's undefined behavior, because the values are out of range. The suggested change requested in the DR would allow implementations to normalize erroneous data, but it could also result in changing the behavior of conforming implementations. None of the data values that are presently allowed are bounded, so technically they are legal, even though they exceed normal values. We don't really have a problem with the suggested TC, however it is worded awkwardly, and needs to be reworded. Need to clarify what 'normal' ranges mean – point to the footnote in the Standard that does this. Clarify a proposed TC to indicate criteria for undefined behavior.

ACTION – Nick to write up a proposed TC. - DONE

Status: OPEN.

Email: SC22WG14.11073: Email from Tom Plum, with responses from Fred Tydeman.

Subject: (SC22WG14.11073) domain errors, revision #1

On Fri, 24 Mar 2006 12:23:31 -0500, Thomas Plum wrote:

>[I omitted one of the crucial sentences from 7.12.7.4; please discard
>the previous post and keep this one.]

What you have here is still wrong. See below.
pow() has been modified by TC2; DR 241.
(John, thanks for the xref in N1142 so I could find the DR).

>I've come to the conclusion that reasonably well-informed people could
>infer two different requirements regarding certain domain errors. Here
>are some citations:
>
>7.12.7.4 The pow functions
>
>2 The pow functions compute x raised to the power y. A domain error
>occurs if x is finite and negative and y is finite and not an integer
>value. A domain error may occur if x is zero and y is less than or
>equal to zero. A range error may occur.

Here is the current text (C99+TC1+TC2):

2 The pow functions compute x raised to the power y. A domain error occurs if x is finite and negative and y is finite and not an integer value. A range error may occur. A domain error may occur if x is zero and y is zero. A domain error or range error may occur if x is zero and y is less than zero.

>F.9.4.4 The pow functions[...]
>1 -pow(±0, y) returns ±[infinity] and raises the "divide-by-zero"
>floating-point exception for y an odd integer < 0.
>- pow(±0, y) returns +[infinity] and raises the "divide-by-zero"
>floating-point exception for y < 0 and not an odd integer.
>
>--------[end of citations]-------------------------------------
>
>I see two reasonable ways to read this.
>
>1. Any time a domain error is reported, if math_errhandling &
>MATH_ERREXCEPT is nonzero, the "invalid" floating-point exception is
>raised ... end of story, no exceptions. When clause F specifies a
>different exception such as zero-divide, then both zero-divide and
>invalid must be raised.

No.

>2. F.9 para 5 and 12 mean that the behavior defined in clause F shall
>override (supercede) the behavior defined in clause 7. When clause F
>specifies zero-divide, that supercedes the requirement in clause 7, and

>only the zero-divide is raised.

Yes.

>Or actually a third reading is possible: interpretations 1 and 2 are
>both possible readings, therefore applications must be prepared for
>implementations that follow either 1 or 2, and implementations are free
>to implement either 1 or 2.

No and Yes. Cannot have both interpretations 1 and 2. But, the main body (7.12.7.4) allows either (or both) domain and range errors for this condition, so if Annex F is not being claimed, then there are many ways to indicate failure here.

>Can WG14 support, or eliminate, any of these interpretations?

A lot of time was spent getting the special cases correct in Annex F.9. Unfortunately, not much time was spent in altering the main body of C99, 7.12.*, to allow the behavior specified in F.9. As conflicts have been noticed between F.9 and 7.12, I have opened DRs to get 7.12 altered to allow the behavior of F.9. Due to existing implementations, the 7.12 sections cannot be changed to only allow the F.9 behavior; they need to allow the existing practice and the F.9 behaviors. So, for example, DR 241 has TC text that is different than the author's suggested TC text.

In the case of conflicts between F.9 and 7.12, I would use the behavior of F.9 and open a DR to get 7.12 fixed.

As Tom McDonald pointed out, the issue of errno versus FP exception flags (how the error is reported) is independent of which error is reported; but if __STDC_IEC_559__ is defined, at least the FP exception flags must be raised as indicated in F.9.

=================== end of Tydeman email ========================

Are the words contained in 7.12.1 consistent with those in Annex F? The text is vague and needs clarification. Is log(0.0) a domain error or a range error? There are arguments for either. Are users served by us changing whether or not something is a range of domain error? Very debatable.

This whole topic needs to be flushed out further.

Reopened discussion Thursday morning. Key text on this is in 7.12.1;p2. The material in the Standard predates the requirements contained in IEEE 754, which were intended to be reflected in Annex F. Tom believes that the only valid conclusion is that either EDOM, or ERANGE is acceptable. Bill believes the text needs to be rewritten, and agreed to take a cut at it.

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

11. Separate WG14 administration (Benito) and J11/U.S. TAG meetings (Meyers, Walls)

See J11 / WG14 US TAG Minutes at the end of these minutes.

12. OWG: Vulnerability (Moore - teleconference) (N1163)

Jim Moore walked through a PDF slide presentation on new project named “Avoiding Programming Language Vulnerabilities", a presentation made at the SC22 Plenary in Mont Tremblant. The project will produce a Technical Report that will provide comparative guidance on the vulnerabilities of multiple programming languages, and identify costs / programming options of avoiding those vulnerabilities.

The Dept of Homeland Security is funding Jim’s effort in this arena. In SC22, the focus of this effort is likely to be on poor use, and poor design of programming languages. Identify the vulnerabilities that exist in each language, and provide guidance on dealing with those vulnerabilities. The working group must be explicitly continued every year by SC22, otherwise it goes away. There is a web site with additional information, and a mailer. The web site is: http://aitc.aitcnet.org/isai. Membership to the Working Group is open members from all SC22 Groups, as well as other languages not included in the SC22 program of work. Rex Jaeschke is looking into ECMA language groups that might have an interest in this work. Meeting venues, dates, etc are TBD. JB is trying to put together a meeting sometime this summer, possibly around a NIST meeting on the same subject on 29 June, 2006.

Does a list of vulnerabilities exist? A number of organizations are working on lists that will hopefully serve as starting points for this work. About 12 folks indicated an interest in attending a meeting, once scheduled.

13. Special Math Functions (Plauger, Walker) (N1051) (TR 24733)

Chris Walker and Bill Plauger presented the draft TR for a set of special math functions being adopted by C++ Library TR 19768. These are essentially the same functions being adopted by C++ in their library TR, tailored to the C language. This document is probably ready for registration with SC22. Bill pointed out that the values computed with special math functions are very susceptible to ‘sensitivities’ based on the input values provided. Outside of specific ranges, the values computed are essentially useless. Thus, implementations of these functions are not likely to be nearly as robust as implementations of the C90, and C99 math libraries. A few functions will likely be dropped from the TR in sync with those already dropped from the C++ TR. We will extract from the existing C++ TR, account for DRs that exist for the C++ TR, have a small editorial review committee review the document, and forward it to SC22 for registration. Expect this to take about a month. The editorial committee consists of Plauger, Tydeman, Benito, and Willem Wakker.

14. Administration

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

14.1 Future Meetings

2006 Fall - Portland, Oct 23-27 (C), C++ is the week prior. Lloyd Center, good meeting facilities, good internet, etc., etc.

We have no meetings scheduled past 2006, but tentatively:

2007 – Spring, Talking to BSI, Neil Martin, about hosting. The ACCU conference will likely start around 15 April. Possibly April 23-27, following C++.

2007 – Fall, Kona. Possibly with SC22 plenary. Tom is looking at two places. The Royal Kona Resort, downtown, least expensive location. It is getting a little long in the tooth? PJ says the place is being overhauled. Four miles south, another location that is a bit more upscale, slightly more expensive. Tom will check on the first location. Ironman is October 13, 2007. We may want to start on Tuesday, Oct 16, and go thru Saturday.

2008 – Spring. No host designated, somewhere in Europe. May consider Japan.

2008 – Fall, Boulder CO, hosted by Cisco. C++ will probably meet in Texas, at College Station.

14.1.1 Future Meeting Schedule – see above

14.1.2 Future Agenda Items

No new items.

14.1.3 Future Mailings

Post Berlin mailing items to JB by 28 April 2006.

Pre Portland mailing items to JB by 25 Sep 2006.

14.2 Resolutions / Votes

None

14.2.1 Review of Decisions Reached

Forwarding the TRs listed below in Action Items as indicated.

14.2.2 Formal Vote on Resolutions

None.

14.2.3 Review of Action Items

Prior
1. ACTION: Tom Plum to propose a response to DR 315 based on where C++ is going - DONE
2. ACTION: Randy to write a paper with a proposed response for DR 314.

New
3. ACTION: Randy and Mark to work with Nick to write up a draft response to N1160.
4. ACTION: Randy to add words to the rationale for TR 24731 to better clarify the issues raised in N1160.
5. 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
6. ACTION: Fred to work with Edison to write up FLT_DEN with N1151.
7. ACTION: Convenor to ask Sally (ANSI) to distribute the Rationale during Registration Ballot for TR 24732, Decimal FP.
8. ACTION: Convenor to provide a response to the submitter of N1152 explaining that we are not revising the Standard at this time.
9. ACTION: Rich Peterson to write up Proposed Response to DR 311 – DONE, on Wiki
10. ACTION: Rich Peterson to write up a new DR that addresses the Discussion issues contained in DR 311. – DONE, on Wiki
11. ACTION: PJ to write up a revised version of the text in 7.12.1;p2, and for a proposed defect report.
12. ACTION: Convenor to forward the draft of PDTR 24733, Special Math Functions, as revised by the editorial committee, to SC22 for Registration.
13. ACTION: Convenor to forward the draft of PDTR 24732, Decimal Floating Point, as revised by the editorial committee, to SC22 for Registration.
14. ACTION Convenor to forward the draft of TR 24731, Bounds Checking, to SC22 for DTR ballot.

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

14.2.4 Thanks to Host

Thank you DIN, Standards Germany for the meeting facilities, support and hosting the meeting.
Thank you SAP for providing great lunches.
Thank you Dinkumware for the Wiki.

14.3 Other Business

None.

15. Adjournment

Adjourned at 1150 hrs, 30 March 2006.

===
=========

Minutes for the INCITS/J11 U.S. TAG Meeting, Tuesday, March 28 at 1630 hrs

Attendees:

Randy Meyers	Silverhill Systems	USA	J11 Chair
Douglas Walls	Sun Microsystems	USA	J11 IR
Fred Tydeman	Tydeman Consulting	USA	J11 Vice Chair
John Benito	Blue Pilot	USA	
Barry Hedquist	Perennial	USA	
David Keaton	self	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	
Robert C. Seacord	CMU/SEI	USA	
Rich Peterson	HP	USA	
Bill Seymour	self	USA	
Martyn Lovell	Microsoft	USA	
Tom Plum	Plum Hall	USA	
Francis Glassborow	Plum Hall	USA	
David Schwab	Oracle	USA	
Hal Burch	CMU/SEI	USA	
Edison Kwok	IBM	USA	

Meeting Started at 1645, 28 March, 2006

Meeting Chair: Randy Meyers, J11 Chair, Not Voting.

Meeting Secretary: Barry Hedquist, Perennial.

1. INCITS official designated member/alternate information.

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

2. Adjournment at 1648, 28 March 2006. Motion to adjourn PASSES, Unanimous Consent.