DRAFT MINUTES FOR 25-28 September 2005
MEETING OF ISO/IEC JTC1 SC22/WG14 AND INCITS J11

WG14/N1145

Meeting Times

Sunday   25 Sep 2005 08:30-12:30 13:30-17:00
Monday   26 Sep 2005 09:00-12:30 14:00-17:00
Tuesday   27 Sep 2005 09:00-12:30 14:00-17:00
Wednesday 28 Sep 2005 09:00-12:30 14:00-17:00

Meeting Location:

Club Tremblant
L'Hotel Du Lac
121, rue Cuttle
Ville de Mont-Tremblant
Quebec, Canada J8E 1B9
Tel: 819 425-2731
Fax: 819 425-5617

Host:
Standards Council of Canada

Host Contact information:
Stephen Michell
E-Mail: Stephen.Michell@maurya.on.ca

Meeting / venue information: N1121

1. Opening activities

1.1 Opening Comments (Michell, Benito)

Steven Michell welcomed us to Mont Tremblant. He should be here most of the
next two weeks, 613 299-9043 is his cell, and is staying in condo 313. Lunch is
booked at 12:30. Supper is at 6:30. Social event at the Sugar Shack, leaves at
5:45 PM, Tuesday night.

1.2 Introduction of Participants/Roll Call

John Benito	WG14 Convener	USA
Barry Hedquist	Perennial	USA
Fred Tydeman	Tydeman Consulting	USA
David Keaton	self	USA

Cecilia Galvan	Freescale	USA	
P. J. Plauger	Dinkumware, Ltd	USA	
Tana L. Plauger	Dinkumware, Ltd	USA	
Randy Meyers	Silverhill Systems	USA	
Douglas Walls	Sun Microsystems	USA	HOD
Mark Terrel	Cisco	USA	
Nick Stoughton	FSG	USA	
John Parks	Intel	USA	
Robert C. Seacord	SEI/CMU	USA	
Jeff Muller	Oracle	USA	
Rich Peterson	HP	USA	
Bill Seymour	self	USA	
Martyn Lovell	Microsoft	USA	
Clark Nelson	Intel	USA	
John Hill	SC22 Chair		
Edison Kwok	IBM	CANADA	HOD
Francis Glassborow	self	UK	HOD
Willem Wakker	ACE	Netherlands	HOD
Keld Simonsen	RAP	Norway	HOD

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. 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, (N1116) (Hedquist)

Approved as amended - N1140

1.6 Review of Previous Action Items and Resolutions (Hedquist)

1. ACTION - Convenor to look at generating a cross reference of DR #s to TC changes in the Standard. DONE

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

3. ACTION - Nick Stoughton to produce a proposal for additional functions to be considered as possible additions to WDTR 24731. DONE

4. ACTION - Randy to produce a new WDTR in time for an editorial meeting that also incorporates the discussions held to date. DONE

5. ACTION - Editorial Committee review N1114 prior to sending up to SC22 (Hedquist, Meyers). DONE

6. ACTION - Convenor to send disposition of comments for WDTR24731 to SC 22 after review by editorial committee. DONE

7. ACTION - Randy to define "known constant size". DONE

8. ACTION - Convenor to establish a liaison with the SC22 POSIX Advisory Group. OPEN

1.7 Approval of Agenda (Benito) (N1133)
lunch will be 1 hr at 12:30. On From Monday on, lunch will be 1 1/2 hours, 12:30 - 2:00.

Add to #5 N1123, in post Lillehammer mailing

Add to #5 N1138 from Andrew Josey, TOG

Two item 7's become 7a, and 7b.

Item #7a, add TR18037, delete TR24731

Item #7b, note name change

Item #9a - 9d, replace with TR24731

MOTION: Agenda approved as modified (no objection)

1.8 Distribution of New Documents

N1135, N1136, N1137, N1138 are all on the Wiki. N1136 is on the Berlin page.

1.9 Information on Next Meeting (Germany - Mori) (N1136)

The meeting in Berlin will take place at DIN Headquarters.

1.10 Identification of National Bodies (Benito)

Six countries represented:

UK,
Canada,
USA,
Netherlands
Norway

1.11 Identification of J11 voting members (Tydeman)

16 voting J11 members out of 20 possible members. (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 April, DIS ballot in process for revision of ISO/IEC 11404, for LIS, Arithmetic. Meets again in New York 5-6 October.

2.3 SC22/WG14 (Benito)

Expect John Benito to be renamed as Convenor for WG14 at SC22 plenary.

2.4 J16/WG21 (Sutter)

WG21 will meet next week. WG21 Library TR is in ballot for WDTR registration & ballot. The ballot has closed. Keld asked to be added to the WG21 liaison list. Done by Convenor.

2.5 FSG - Free Standards Group (Stoughton) (N1139)

Nick presented a written report (good job!!). Final result of the Linux Standard Base Core Specification (Part 1-8) was 100% approval. The document can be seen at www.linuxbase.org.

2.6 Other

WG14 asked Tom Plum, and any other WG21 liaisons who wish to contribute, to provide a list of the TRs in under consideration, in process and published by WG14 for potential inclusion on C++0X.

3. Defect report status (Benito) (N1125)

Willem Wakker and John Benito developed a document that cross references the DR / publication and status. See N1125

4. Implications of C++ Memory Model Discussions on the C Language. (Benito) (N1131)

This paper has been provided for our information, and is being presented to WG21. We should follow this effort as it develops. The authors have asked to be made aware of any C specific issues that are likely to arise. Current proposals and discussions can be found at http://www.hpl.hp.com/personal/Hans_Boehm/c++mm.

There is no real action for us to take at this time.

5. Potential defect reports (N1128, N1138) (Stoughton), (N1123) (Benito)

N1128 - TC2 made perror a byte output function. Add to DR log. DR 322

N1138 - Add to DR log. DR 323

N1123 - Submitted via Convenor. Add to DR log. DR 324

Paper on FILE as an object type, rather than an incomplete type, to preclude nasty things from being done to the file. Clark pointed out that any of the nasty things would be nonconforming anyway. Randy would like to dig into, PJ agrees with Clark - no need to do anything. On second thought, there is an opportunity to restate, and clarify the issue.

6. Proposal to augment the interface of malloc/free/realloc/calloc (N1085) (Galvan)

These items add new features to the C language, and would likely have to be handled as a Technical Report. The paper reflects existing implementation(s).

The potential impact on existing implementations is unknown at this time, but none are known at this time. Nick suggested that Austin Group would probably be interested in discussing this. Tom Plum believes this work rightfully belongs in C. If we decide to make this a TR, we will need an editor. Tom believes that we should move quickly on this item, as a TR, as market demand already exists. PJ believes there is no conflicting art to reconcile, so there is no need to rush into doing this. Tom pointed out, and PJ agreed, that it would be 'dirty pool' to not do anything with this proposal, and embrace an alternative one a year from now. JB's concern is not taking on more work than we can accomplish given the number of TRs we already have in process. Tom suggested that C++ may decide to incorporate these changes, since Boost has already implemented them. He further suggested that WG14 give WG21 its blessing. Nick pointed out that this work could also fit within the scope of his dynamic memory allocation proposal. Francis spoke in favor of encouraging WG21 take this on. There is general consensus that this work is a 'good' thing, and should be undertaking 'somewhere'.

7a. How to proceed with TR18037, TR 24732, and TR 24747. (Benito)

TR 18037 - Embedded Systems. Willam has an updated version, N1120, based on resolved defects reports to the initial version. The document will be forwarded for republication.

TR 24732 - Decimal Floating Point. The committee revising IEEE 754R has run into a significant issue in the revision. TR 24732 depends on that document. There is nothing we can do to proceed with that TR until the key issues are resolved. We need to tell SC22 what our plans are for proceeding. Fred volunteered to work with Edison to develop a rationale.

ACTION: Request that the Convenor ask SC22 for a one year extension for TR24732.

TR24747 - Special Math. This TR is tied to C++ TR1, which has been balloted once. We should have a 'C' version to look at in April. Expect it in the pre-Berlin mailing.

7b. Specification for Additional C Library Functions - Part 2: Dynamic Allocation Functions (N1126) (Stoughton)

Nick pointed out that, in most cases, functions in this document are in existing practice either in POSIX, or Linux; and that all wide character functions are not addressed. It still addresses buffer overflow, like Part 1, but in a dynamic way rather than static. Whether static or dynamic allocation is used would depend largely on the environment.

8. Specification for Managed Strings (N1132) (Seacord)

Robert gave a slide presentation of the above paper. This has a similar purpose to N1126 above, however none of the functions exist in practice.

The presentation was followed by some discussion on exactly how much safety can be built into the language while maintaining one of its best characteristics, efficiency. Adding safety mechanisms will always detract from efficiency and performance, thus trade-offs are always a consideration.

PJ pointed out that C++ has a mechanism that does dynamic allocation. Tom pointed out one problem with dynamic allocation is that a program could attempt to allocate a very large buffer top the point of creating a denial of service problem. However, both Nick and Robert believe that problem could be addressed. Many believe that a dynamic allocation approach is desirable, but are concerned over whether or not we are creating a complete solution, or creating band-aids.

If we want to proceed, it's not clear what, if anything, that Nick and Robert can do with their proposals that they have not already done.

For now, we are taking no action.

9. ISO/IEC 9899:1999 + TC1 +TC2 (N1124) (Meyers)

Discussion on whether or not we want to submit N1124 for publication. We have not done a formal review, and there is no pressing need to do so. The Convenor, in his report to SC22, stated that the document would be forwarded. WG14 asked the Convenor to not take that action.

10. TR24731 (N1135) (Meyers) (added)

Randy walked thru N1135, with emphasis on the changes made since the last time the document was reviewed in this committee. The bulk of the changes came from the editorial review meeting in June 2005. In general, no new material will be added to this document.

3.1. One of the larger changes had to do with the use of the term "constraints", which was changed to "runtime constraints". The definition of "runtime constraint" still needs to be better distinguished from the definition of "constraint" in the C Standard, and that it need not be diagnosed at 'translation time'. Randy to modify Note 1 in 3.1 to clarify the above.

6.1.4;p5. Should we keep 6.1.4;p5? Yes, but make it a footnote.

6.1.4;p3. Reword to avoid the use of the term "sometimes".

6.6.1.1;p2. Needs to be reworded.

6.6.1.2;p2. Change "is" to "shall be". Same for the two functions that follow.

6.6.1.3;p3. Change to "...shall simply return to its caller."

6.6.1.4 strict_handler_s - does it reflect what we agreed to at the editorial meeting in June? Do we want to eliminate it altogether? Yes. Also - Add a footnote to the 'abort handler' that will allow the implementation to do something before the abort, referring to calling a 'debugger'. Robert Secord pointed out that introducing the handlers creates an insecure feature in the TR, and that it may be worthwhile to point this out, somewhere, such as the rationale. Any function pointer is a potential area of attack. We may end up adding additional handlers. There may be other ways to handle the issue, such as using read only memory.

Overlapping operands - "Copying shall not take place between objects that overlap." Add to strcpy, strncpy, strcat, strncat, wcscpy, wcsncpy, wcscat, and wcsncat, (_s for all), runtime-constraints list. Delete any paragraph that says any such copying results in objects of unspecified values. Copy these words in 6.9;p3 to 6.5.3;p3

6.5.2.1 Open mode 'u' - for fopen_s, and freopen_s, means use system default protections when creating a file.   Two cases: 'w' creating a file, 'a' appending a file. Add 'u' to w -or- a, i.e. 'uw', or 'ua'. etc, wherever 'w' or 'a' can be used. Example: ua+b and uab+, etc. Nick raised an additional proposal. The text below is a corrected version of Nick's text: apply to fopen_s and freopen_s

Description
===========

4 The fopen_s function opens the file whose name is the string pointed
 to by filename, and associates a stream with it.

5 The mode string shall be as described for fopen, with the addition that both "w"
  and "a" modes may be preceded by "u", see below:
        uw   truncate to zero length or create text file for writing, default
permissions
        ua   append; open or create text file for writing at end-of-file, default
permissions
        uwb truncate to zero length or create binary file for writing, default
permissions
        uab append; open or create binary file for writing at end-of-file, default
permissions

uw+ truncate to zero length or create text file for update, default permissions

ua+ append; open or create text file for update, writing at end-of-file, default permissions

uw+b or uwb+ truncate to zero length or create binary file for update, default permissions

ua+b or uab+ append; open or create binary file for update, writing at end-of-file, default permissions

6 To the extent that the underlying system supports the concepts,   files opened for writing shall be opened with exclusive (also known   as nonshared) access. If the file is being created, and the first character of the mode string is not "u", to the extent that the underlying system supports it, the file shall have a file protection that prevents other users on the system from accessing the file. If the file is being created and first character of the mode string is   "u", then after the file has been closed it shall have the system default file access permissions.[footnote - These are the same permissions that the file would have been created with by fopen]

7 If the file was opened successfully, then the pointer to FILE pointed to by streamptr will be set to the pointer to the object controlling the opened file. Otherwise, the pointer to FILE pointed to by streamptr will be set to a null pointer.

==== end of Nick's text ===

6.5.2.2;p4 2nd sentence - needs to be reworded, tied to fopen.

6.5.1.1;p4. next to last sentence, add: "...as if fopen_s.", then delete the last sentence.

6.5.2.1 - Add a runtime constraint: "The string pointed to by the mode parameter shall be an implementation-defined mode string." Much discussion over adding this runtime constraint at this time.
Straw Poll: Do we want to add the above or do nothing?
Do nothing - 16. Do something like this with 'real' words - 1.
RESULT: DO NOTHING

6.6.1.1;p2;s2, Change to read: "The set_constraint_handler_s function sets the runtime constraint handler to be 'handler'. The runtime constraint handler is the function to be called when a library function detects a runtime constraint violation."

All printf_s functions, added: forbid %n.
Straw Poll: Do we want to keep the restriction of eliminating %n? No support to delete the restriction. RESULT: forbid %n.

sprintf_s - editorial committee changed the return value.
count += sprintf_s (dest, sizeof dest, fmt1, arg1, arg2);
count += sprintf_s (dest + count, sizeof dest - count, fmt2, arg3, arg4);
Discussion on the return value, to have a coding error return a "-1".
ACTION: David Keaton to write a proposal regarding the return value for
sprintf_s.

6.6.5 mbstowcs_s, et al: mbstowcs_s, wcstombs_s, mbsrtowcs_s, wcsrtombs_s
Should the above functions always null terminate results? YES
Should retval count the null terminator? The Standard functions do not. Straw
poll: Do not count the null terminator - 15; Count the null terminator - 2; Count it
conditionally - 2. RESULT: DO NOT COUNT THE NULL TERMINATOR.

6.6.2.1 getenv_s - rename parameter 'needed' to 'len'; p5s2: 'one plus the length'
becomes 'the length'.

strtok_s, wcstok_s - new parameter, s1max, to make sure function does not store
outside of string tokenized. wcstok_s was added by the editorial committee in
June.

6.6.3.1 - bsearch_s. When can 'key' be null? Replace the runtime constraint in
6.6.3.1;p2s2 with: "If 'nmemb' is not equal to zero, then none of 'key', 'base', or
'compar' shall be a null pointer."

6.9.3.2.2p2 wcsrtombs_s - Add to runtime constraint: "If dst is a null pointer,
dstmax shall be zero."

Known defects that remain in TR
- make sure strnlen_s and wcsnlen_s are not called strnlen or wcsnlen
- page 4, refs to clause 5 should be to clause 6
- title page
- delete fn 70

=== end of TR24731 review ===

TR24731 - Rationale
Randy walked us through the rationale. The emphasis of this review was to
make sure that the items we said we would add to a rationale in our Disposition
of Comments (N1114) is included in the rationale. Unless otherwise noted, all of
the items are unchanged. The notations below refer to the comment number
from the Disposition of Comments, i.e. CA02, followed by the reference in the
Rationale that addresses that comment, i.e. 6.7.3.1

There was some discussion regarding whether or not the Rationale was
'ballotable'. The answer to that is no. The bulk of the comments are addressed
in 1.1 Goals

CA02 re: 1.1.15, 6.7.3.1, 6.7.2.1, 6.7.1.3

CA02 re:1.1.15 <changes?>

CA02 re: 6.7.3.1 (strtok_s) Nick wants an additional paragraph to address the fact that we have not really solved the problem - there is an issue that we did not address. Is there a security issue here? Not likely. ACTION: Randy to collaborate with Nick to add a paragraph to TR24731 Rationale 6.7.3.1.

CA02 re: 6.7.2.1(strcat_s) ACTION: Randy to add words on the Committees decision regarding the use of a handler to Rationale 6.7.2.1.

DE01 re: 1.1.3, 1.1.7

JP-General re: 1.1

JP01 re: 6.5.4.1 ACTION: Nick to add words regarding fgets to Rationale 6.5.4.1

JP02 re: 1.1.1 ACTION: Randy to add some words on the frequency of attacks related to buffer overflow.

Terminated review. ACTION: Randy to generate a cross reference of comment to Rationale paragraph numbers.

TR24731 - N1141 - Keaton

David presented this paper regarding sprintf_s return values, and proposes to improve the ability to automatically remediate code by returning a negative value if, and only if, sprintf would have returned a negative value, and return zero in all other cases. Nick asked if the return logic could be reversed. David explained that the return logic would have to match sprintf in order to be a drop in replacement.

2. Edits - move "if" in first sentence to beginning of sentence.

Martin is skeptical about doing this for swprintf_s. Consensus to keep it parallel. Martin agreed.

Paper accepted to add to TR24731.

TR24731 - Permit Fatals (Plum) (N1134)
Tom walked us through N1134. The general purpose is to permit the compiler to treat compile-time violations of a runtime-constraint in the same way that syntax violations, or other caught errors, are treated. The compiler can generate an error, or a warning that can be suppressed.   Rich asked what happens if he is

using the 'ignore' handler. Good question, it's not readily clear. PJ believes that the situation already exists, and is not a good reason to 'not' permit fatals. The key to this concept is allowing a compiler to terminate compilation, i.e. not generate an object file. The TR does currently not allow for termination of compilation for a runtime-constraint violation, instead, it requires that the constraint handler be called at runtime.

Straw vote: Permit fatals with an ID mechanism to disable . Yes 14, No 4

ACTION: Tom Plum to write up wording to add to TR24731 to permit fatals with an ID mechanism to disable it.

Tom then decided to NOT champion this effort, but asked if anyone else wanted to, they could talk to him.

TR24731 - THE NAME OF THE TR (PLUM)

Tom recapped the evolution of the name changes. He suggested that we call it the 'safer library'. PJ objected simply due to the myriad of problems that can arise over complaints. Tom's fallback position is that we agree on an informal name to allow us to communicate this to the world.

Nick expressed a concern that changing the name to "Library Extensions" might be seen as an expansion of scope. Several disagreed, pointing out that the 'scope' is a matter of 'fact', and the only SC22 is the arbiter of whether or not our scope has changed.

JB previously polled a number of folks on a variety of names making use of 'safer', 'secure' and 'reliable', and got no response.

Straw polls:

1. Keep "safer" in the name - yes 10, no 9.
2. Use "C Library Extensions" - yes 13, no 7.

JB and Randy will go off and come up with another list of proposed names.

TR24731 - THE NAME OF THE TR II (Benito)
JB polled some folks, got more suggestions, et al
- bounds-checking interfaces - yes 10; round 2, yes-12
- runtime constraint handling - yes 11; round 2, yes - 8
- improved reliability - 3
- seatbelted - 2
- extensions only - 7
Keep Part 1 in title.

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. Defect Reports (Stoughton)

DR 219 - has a proposed response posted. Moved to REVIEW.

DR 236 - has a proposed response. Edison reported that Raymond Mak was OK with the proposed response. Moved to REVIEW.

DR 298 - has proposed TC. Look at SC22WG14_10903 There are two possible ways to incorporate the suggestion, or possibly do both.
Straw poll: Change list -13, by footnote-0. RESULT, change the list. Status: OPEN, modify Proposed TC.

DR 300 - moved to CLOSED.

DR 301 - moved to CLOSED

DR 302 - In REVIEW with a proposed TC. Moved to CLOSED

DR 303 - In REVIEW with a proposed TC. Moved to CLOSED

DR 304 - Open, with proposed TC. Moved to REVIEW

DR 305 - In REVIEW with proposed TC. Moved to CLOSED

DR 306 - In REVIEW with proposed TC. Moved to CLOSED

DR 307 - In REVIEW with proposed TC. Moved to CLOSED

DR 308 - In REVIEW with proposed TC. Moved to CLOSED

DR 309 - In REVIEW with proposed TC. Moved to CLOSED

DR 310 - In REVIEW with proposed TC. Moved to CLOSED

DR 313 - In REVIEW with proposed response. Moved to CLOSED

DR 315 - In REVIEW with proposed response. Change answer #2 to "..sizeof(int) - editorial. Additional discussion. Answer to #2 may actually be undefined, same with #3. There does not seem to be a positive statement of semantics for Question #2. Tom suggested that we wait until we know what C++

adopts in this area. Clark pointed out that C++ promotes to int. The answers to question 2 & 3 should be withdrawn. See 6.3.1.1;p2. ACTION: TOM PLUM to propose a response to DR 315 based in where C++ is going. Moved back to OPEN.

DR 316 - In REVIEW with proposed response.   We believe the answers to Q1 & 2 are yes, however, we also feel that the Standard is unclear with respect to 3 & 4. As discussed previously, behavior of unprototyped functions is not an area we want to refine or clarify. Moved to CLOSED

DR 317 - In REVIEW with proposed response. Add to response: "The answer to question #1 is NO, and to question #2 is YES. There are no constraint violations, however, if the function call were executed it would have undefined behavior. See 6.5.2.2;p6. Keep at REVIEW.

DR 318 - In REVIEW with proposed TC. Moved to CLOSED.

OPEN DRs

DR 311 - Definition of variable modified types. Tom: It takes a sequence of specifiers followed by a declarator to specify a type. We may have oversimplified what we said. The sentence in 6.7.5;p3 that defines variably modified may be wrong, and that may not even be the right place for it to be defined. The definition ties it too closely to the declarator. The definition in the standard for variable length array does not seem to be in italics (side issue?? - See 6.7.5.2;p4 In the example provided in the DR, the type of y is variably modified. The declarator for y does not include a variable length array type. Para 3 needs to have it's wording adjusted in some fashion, the text there is insufficient to provide us the answer we should get. Someone needs to dig into this and propose words - Rich volunteered.
ACTION: Rich Peterson and or John Parks to propose words for DR 311.
Status: remains OPEN

DR 312 - Does 'known constant size' mean something different from 'not a VLA'? OPEN, but has a proposed TC. Clark pointed out 6.7.5.2, seems to negate the proposed response. Randy thinks the definition there is recursive, and supports his proposed TC response.   Moved to REVIEW.

DR 314 - Cross-translation-unit tagged type compatibility. OPEN, in REVIEW, with a proposed response. Randy wants to write a paper on this because he believes the proposed response is wrong. ACTION: Randy to write a paper with a proposed response for DR 314. He believes that the Standard is unambiguous. However, it's a complex issue and NEEDS a paper to try to explain. Leave OPEN.

DR 319 - OPEN Status. Trailing zeros with format specifier %a. This report asks whether trailing zeros are removed or kept in the case of printf ("%a", x); when x is a double (double x = 1.0), and FLT_RADIX is 2. Fred believes our intent was that trailing zeros be removed.   The Standard does not specify whether or not trailing zeros are removed. Nick pointed out that 'changing' the Standard could easily break implementations, and is not what we really want to do. Plum agreed. Not a defect, however we may want to consider establishing a rule for removing or not removing at some point in the future. Moved to REVIEW.

DR 320 - OPEN Status. Per Willam, the first sentence of 6.7.5.2p2 seems to suggest that any ordinary identifier both block scope or function prototype scope and no linkage has a variably modified type. This is clearly wrong. The shall emphasis seems to be misplaced. The original text is 'right', but can easily be misunderstood. Clark suggested we also change "..and no linkage." to "..and shall have not linkage." Revised the proposed word to: "An ordinary identifier (as defined in 6.2.3) that has a variably modified type shall have either block scope and no linkage or function prototype scope. " Willam liked the revised words. Put the revised words as a Proposed TC, moved to REVIEW.

DR 321 - OPEN Status. Wide character code values for members of the basic character set. POSIX has a problem with the change made in TC2. In essence, it breaks many implementations. A feature test macro, __STDC_BTOWC_NEQ_WCTOB__, is proposed to resolve the problem. The proposed solution will be adopted by POSIX regardless of what we decide to do. General agreement that adopting the proposed change has no real impact. The behavior is not changed. TC2 defined the default behavior as 'they may not be equal', i.e. equality is optional. Should that be considered the default? We want to allow the latitude that the two may not be equal and advertise its existence. Keld wants the macro to reflect equality, i.e. use "EQ" rather than "NEQ". Nick pointed out that doing so will continue to break implementations. PJ wants to make it clear that equality is still an option if the feature test macro is defined. Another proposed macro: __STDC_BTOWC_MIGHT_NEQ_WCTOB__ . Dave suggested that the proposed macro name implies that converting a byte to a wide character, then back again, may not be equal. That interpretation would be wrong, i.e. such conversion is always equal. New macro proposed: __STDC_MB_MIGHT_NEG_WC__ . The use of "MB" was questions, however it was ultimately judged to be correct. Straw poll: use the new proposed macro yes 18, opposed 0, abstain 1. RESULT, change the macro, moved to REVIEW.

DR 322 (N1128) - OPEN Status. Problem with TC2 change adding perror to the list defining byte I/O functions in 7.19.6.1. This TC created a problem with POSIX implementations, where perror is NOT classified as a byte I/O function. Per POSIX: "The perror function shall not change the orientation of the standard error stream." A high percentage of perror's are followed by exits or aborts, so it's important that perror get it's message out any way it can. PJ believes that perror is only a convenience function, used mostly in small programs to write to standard error. Whatever we do, we cannot says what POSIX says. We do not

yet know what we can do. We do not see a change that will resolve the POSIX problem. Can the POSIX perror rule be considered an extension that does not break C? Not really, due to it being added the list of byte I/O functions in the C Standard. If the stream is known to be wide, strerr and fwrite can be used to write the error message. We may have reached a resolution for this problem, that will permit the POSIX behavior, by removing perror from the list, and permitting perror to set the orientation of the stream, if it was unset. Make such behavior either unspecified, or implementation defined. Making the behavior I-D would require all implementations to document it, so unspecified might be a better choice, or even undefined, which would allow perror to fail if the stream orientation has already been set to wide.   PJ and Nick agreed that using undefined would be OK. Leave OPEN.

DR323 (N1138) - Issue with TC2 Item #34 - No mention of imaginary in the normative text, yet it is still recommended in Annex G. Because it is no longer mentioned in the normative portion, it cannot be defined in an implementation, since the term is no longer reserved.   Support for imaginary was always optional, however removing altogether removed a way to implement it in a 'conforming' way. Further, many believe that implementing Annex G causes the implementation to be nonconforming to C99, thus implementations that implemented 'imaginary' prior to DR 207 (TC2) may not be nonconforming to C99, because the macro 'I' will be incorrectly defined. Prior to TC2, 'I' expanded to '_Complex_I' if '_Imaginary_I' was not defined, otherwise it could expand to either '_Imaginary_I', or '_Complex_I'. PJ wants to know what are we fixing that needs fixing. Nick pointed out that implementing Annex G causes nonconforming changes to the normative text in C99. Exact instances are unknown. Possibly redefine the macro 'I' in 7.3.1? The implications of NOT allowing 'I' to expand to '_Imaginary_I' are not readily clear.

DR 324 (N1123) - Tokenization obscurities.

Summary #1. Examples are: 'a, ". A partial pp token is a sequence of characters that matches a prefix of pp tokens as defined in 6.4, but is not a complete pp token. Unterminated character constants, unterminated strings. ACTION: Tom and Randy to propose words for DR 324.

13. Syntax Standardization (Wakker) Willam presented a brief discussion on standardizing syntax structure to prevent typical problems of error and omission in language definitions. A paper on this subject is being presented to SC22, SC22/N3977, which can be downloaded from the SC22 web site.

14. Administration

14.1 Future Meetings

2006 Mar - Berlin, Germany,   See N1136: Mar 27-31, C++ the following week.

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, Kona

Nick asked about the possibility of NOT co-locating with C++ to make things easier on a host. About seven of us attend both meetings. Back to back with weekend travel for a reasonable distance might be OK.

14.1.1 Future Meeting Schedule

14.1.2 Future Agenda Items

>	None

14.1.3 Future Mailings

>	Post Tremblant meeting mailing items to JB by 26 Oct, 2005

>	Pre Berlin mailing items to JB by 27 Feb 2006

14.2 Resolutions / Votes

14.2.1 Review of Decisions Reached

>	No formal decisions reached.

14.2.2 Formal Vote on Resolutions

>	None.

14.2.3 Review of Action Items

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

ACTION - Convenor to establish a liaison with the SC22 POSIX Advisory Group. OPEN

ACTION: Request that the Convenor ask SC22 for a one year extension for TR24732.

ACTION: David Keaton to write a proposal regarding the return value for sprintf_s. DONE N1141

ACTION: TR24731 Rationale. Randy to collaborate with Nick to add a paragraph to 6.7.3.1. strtok_s

ACTION: TR24731 Rationale. Randy to add words on the Committees decision regarding the use of a handler.

ACTION: TR24731 Rationale. Nick to add words & send to Randy regarding fgets to Rationale 6.5.4.1

ACTION: TR24731 Rationale. Randy to add some words on the frequency of attacks related to buffer overflow.

ACTION: TR24731 Rationale. Randy to generate a cross reference of Disposition of Comments to Rationale paragraph numbers. DONE N1143.

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

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

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

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

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

14.2.4 Thanks to Host

Thank you Standards Council of Canada.

Thank you Keld for the web site.

Thank you Sugar Shack.

Thanks to Dinkumware for the network support.

14.3 Other Business

None.

15. Adjournment

Adjourned at 1445 EDT, 28 Sep, 2005

===

Minutes for the INCITS/J11 U.S. TAG Meeting, Tuesday, September 27 at 1710 hrs

Attendees:

Randy Meyers	Silverhill Systems	USA	J11 Chair
Douglas Walls	Sun Microsystems	USA	J11 IR
John Benito	Blue Pilot	USA	
Barry Hedquist	Perennial	USA	Recording Secretary
Fred Tydeman	Tydeman Consulting		USA
David Keaton	self	USA	
Cecilia Galvan	Metrowerks	USA	
Tana L. Plauger	Dinkumware, Ltd	USA	
Tom Plum	Plum Hall	USA	
Francis Glassborow	Plum Hall	USA	
Mark Terrel	Cisco	USA	
John Parks	Intel	USA	
Robert C. Seacord	SEI/CMU	USA	
Nick Stoughton	USENIX	USA	
Edison Kwok	IBM	USA	
Martyn Lovell	Microsoft	USA	
Bill Seymour	self	USA	
Jeff Muller	Oracle	USA	
Rich Peterson	HP	USA	
Clark Nelson	Intel	USA	

Meeting Started at 1710, 27 September, 2005

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

Meeting Secretary: Barry Hedquist, Perennial.

1. Select US Delegation for the next two WG14 meetings in 2006: Walls, Hedquist, Stoughton, Keaton. Motion to accept the preceding names for the US Delegation for the next two WG14 meetings:(Benito, Tydeman). Passes (15, 0, 0, 19).

2. 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.

3. Motion to restore voting privileges to HP (Tydeman, Stoughton), Passes (15, 0, 0,19)

4. Adjournment at 1726. Motion (Hedquist, Peterson) PASSES, Unanimous Consent.