TR 24731 Editorial Group Meeting
Draft Minutes
WG14/N1129

30 June 2005 09:00-12:00 13:30-17:00
01 July 2005 09:00-12:00 13:30-17:00

Meeting Location:
Microsoft Corp.
1 Microsoft Way
MS Conference Center
Redmond WA, USA 98052

Building 41, meeting room 3541.

Host Company:
Microsoft Corp.
Host Contact information:
Herb Sutter
Microsoft Corp.
1 Microsoft Way
Redmond WA, USA 98052
E-Mail: Herb Sutter

1.1 Opening Comments (Lovell, Benito)

JB and Martin Lovell welcomed everyone to the meeting.

1.2 Introduction of Participants/Roll Call

Attendees:

John Benito	Blue Pilot - WG14 Convenor
Randy Myers	Silverhill Systems - TR 24731 Editor
Nick Stoughton	USENIX
Barry Hedquist	Perennial - Secretary
David Keaton	Keaton Consulting
Tana Plauger	Dinkumware
Tom Plum	Plum Hall
Robert Secord	US-CERT
Jeffrey Havrilla	US-CERT
Josef Wankerl	Metrowerks
Martin Lovell	Microsoft
PJ Plauger	Dinkumware
Mark Terrel	Cisco Systems

1.3 Selection of Meeting Chair (Benito)

Meeting Chair is John Benito.

1.4 Selection of Meeting Secretary (Benito)

Meeting Secretary is Barry Hedquist.

1.5 Approval of Agenda - w/o objection

2.0 Review

2.1. Document Name - JB pointed out that we cannot use the term 'safer C '. It is trademarked in the UK, and we have been asked by the trademark holder to not use that term. There are others that do not like the term 'secure' in the title. 'Constrained' may be useful. Possibly use simply 'Extensions to the C libraries, Part I'. JB will explore if anyone has a serious objection to the word secure in the title, as well as exploring the above. ACTION on JB to do this. Nick believes that Austin Group will object to the word 'secure', in any form, in the title.

ACTION - JB will explore if anyone has a serious objection to the word secure in the title, as well as exploring the use of 'Extensions to the C Library, Part I."

2.2. Constraints vs. Usage Requirements

Latest version of the TR uses the term 'constraint', Randy would prefer to use something like 'usage requirement'. There is enough opposition to the term constraint, since it is a change to the existing definition, to merit consideration to using some other term. PJ likes 'runtime constraint', sees 'usage' as too broad a term. Using 'constraint' allows the compiler the option to detect the problem at runtime, and issue a diagnostic. RM likes the idea of allowing the compiler to detect these problems anytime it can. However, it's not likely that we can 'require' a compiler to do that. Allow the compiler to 'permit fatals', i.e. issue an error at compile time, however we cannot require it.

Straw poll to permit fatals: Yes - 11, No - 1, Abstain - 0

ACTION: Tom Plum will write up exactly what 'permit fatals' means.

Given the above, using 'constraint' as part of the term is appropriate.

RM sees a problem with confusing 'runtime constraint' with 'constraint'. TP proposes making the definition very clear, even to pointing out that is NOT a subset of the initial term. We do not want to change the baseline definition for constraint.

Straw poll, use of the term 'runtime-constraint', Yes - 8, No - 1, Abstain - 3.

2.3 gets_s. runtime-constraint if string does not fit in the buffer.

2.4 Overlapping operands. See 2.6 below.

2.5 Document Review

iv, p1, member should be members

3.1 constraint definition will change.

4. make macro __STDC_LIB_EXT1__

5.1.1 make macro __STDC_WANT_LIB_EXT1__

5.1.4, constraint violation, p3s2, should should be shall

5.5.1.2, tmpnam_s, something needs to be said about the size of the array pointed to by 's'.

General - need to address assumptions on array sizes throughout many of the functions w/r/t requirements on the application.

5.5.2.1, fopen_s, p3, add - if stream not equal to NULL, what the stream points to will be set to the null pointer.
add 'ptr'

5.5.2.1 - footnote 10, make it a 'shall' if supported. Add: A file opened for writing should be opened with exclusive access...

General question/discussion on access of a file after it has been created. Add a default mode character 'u', & make the default behavior more secure. Same w/prior fn #7 (tmpfile_s), except for the 'u' flag.

Need a new function that returns the name created by tmpnam_s? tmpfile_nam_s? Deferred. JB pointed out there was no written proposal to review, the propose of this meeting is to review the current text of the TR not invent new functionality

5.5.3.1 fprintf_s - add %m specifier? no - outside the scope. %n - really need to outlaw what %n does. (Future) There are lots of things that fprintf_s can do to exploit security, that can be prevented by not allowing dynamic strings, or only allowing strings that come only from a trusted source. Add words to rationale and a footnote regarding the use of format specifiers not specified in the TR.

5.5.3.2 fscanf_s - similar to above, as applicable.

Friday, July 1, 2005

5.5.3.5 snprintf_s - may need an additional argument? Footnote to support truncation. Constraint violation if the string is to too large for the buffer. Also, add words to para 4 that correspond to C99 w/r/t overlapping (undefined behavior). The string must always be null terminated.

5.5.3.6 sprintf_s. Constraint violation is supposed to return zero (0). Suggestion to define an encoding error as a constraint violation. OK Same changes for vsprintf_s.

Straw Poll - Do we want to change the spec so that sprintf_s returns zero for a constraint violation? Yes - 6, No - 4, Abstain - 2.

5.5.3.8, vfprintf_s, is va_list guaranteed to be a pointer? No. Drop any constraints on va_list. Add a footnote to check whether or not va_list has been initialized. Applies to all 'v' series. Footnote #17 needs to account for the printf_s family.

5.5.4.1, gets_s - Para 3 needs to be modified. Recommended Practice, s/be para 6, an on ..., change 'process safely' to 'safely process'.

5.6 prototype signature for constraint_handler: Too many arguments. Lack of support for wchar_t. Change the arguments & types to something like:

```
typedef void (*constraint_handler_t)(
      const char * restrict msg,
      void * restrict ptr,
      errno_t error);
```

5.6.1.2, set_constraint_handler. TP: Should 'ignore me' be a valid default setting? PJ: Yes. Checking function error returns is the first order of checking things. TP Concern over programmer practices where the default is 'do nothing'. DK: Different users are going to require different defaults. Add a recommended practice section? Instruction to programmer to define the handler function? Name three handlers: strict, abort and ignore. Strict somehow interrupts the program. It does not 'ignore'. No argument over abort and ignore. Strict is what Microsoft is doing right now.

Straw poll: add strict Yes - 4, No - 3, Abstain - 5. PJ shifted to support strict. Revote: Yes - 7, No - 2, Abstain- 4. Exact names, such as 'abort_handler_s', are TBD.

5.6.2, p1s2, add 'if" in front of memo.

5.6.2.1 bsearch_s. Remove the constraint that the key cannot be null. Add a recommended practice for what to do if key is zero, possibly with sample code. Footnote #25, insert 'this means that' after 'in practice'.

5.7.3.1, strtok_s: ACTION: TP to check if all Undefined Behavior has been removed from strtok_s DONE. Discussion of a number of different ways to implement this function. Proposal to make the value of s1max unspecified in subsequent calls of strtok_s. Let n be the value of *s1max when s1 != null. Each token found shall end before &s1[n-1] on the first and subsequent calls. Concept needs to be folded into the function. This should turn into a proposal. Randy looking for someone to write it. Nick S volunteered to write a proposal. If he does not do it, we'll keep what we've got. Martin may also write a proposal.

5.7.4.1, strerror_s. NS thinks para 6 is over specified. Will come up with words for a proposed change.

5.8.1.2 asctime_s, p2, timer s/be timeptr.

5.8.1.3 swprintf_s should support truncation as swprintf does. snwprintf_s will be added by Randy. add a 'v' version as well.

ACTION: Randy to add snwprintf_s, and a corresponding 'v' function.

General - do we need a wcstok_s? There is a reentrant wcstok (C99), which is why it was not initially included. No objection to adding it. MSFT added it.

ACTION: Randy to add wcstok_s.

5.6.3.1 - wctomb_s - should a null be written to the end of the string after conversion? NO! (wctomb does not write a null). Randy may need to adjust some constraints.

2.6 Overlapping Pointers (See 2.4) Particularly an issue with concatenation ('cat') functions. Randy believes he has a 50/50 chance in getting it right. Sprintf_s and scanf_s family - undefined behavior. In general, many of these areas cannot easily be caught, and will result in undefined behavior. No restriction in allowing implementations to check for overlapping pointers.

3. Administration

3.1 Future Meetings.

We'll need a close reading of the doc in the pre-meeting mailing (Aug 29).

3.2 Review of Decisions Reached (Straw Polls).

Straw poll: To permit fatals (allow, rather than require, the compiler to issue an error message). Yes - 11, No - 1, Abstain - 0

Straw poll: Use of the term 'runtime-constraint'. Yes - 8, No - 1, Abstain - 3.

Straw Poll - Change the spec so that sprintf_s returns zero for a constraint violation. Yes - 6, No - 4, Abstain - 2.

Straw poll: Add 'strict' to 'abort' and 'ignore' constraint handlers. Yes - 7, No - 2, Abstain- 4. Exact names, such as 'abort_handler_s', are TBD.

3.3 Review of Action Items.

ACTION - JB will explore if anyone has a serious objection to the word secure in the title, as well as exploring the use of 'Extensions to the C Library, Part I."

ACTION: Tom Plum will write up exactly what 'permit fatals' means.

ACTION: TP to check if all Undefined Behavior has been removed from strtok_s DONE.

ACTION: Randy to add snwprintf_s, and a corresponding 'v' function.

ACTION: Randy to add wcstok_s.

3.4 Thanks to Host - Thanks to Microsoft for hosting the meeting, providing lunches, refreshments, etc.
Thanks also to Tom Plum for hosting a great dinner Thursday evening!!

4. Adjournment - adjourned at 3:00 PM, Friday afternoon.