

March 1, 2009

1. Status

N1362 is a revised draft incorporating the changes detailed in the Santa Clara minutes (N1346) plus a few more corrections and editorial changes. It contains diff marks for all of the changes from N1336 except for minor formatting changes and subclause number changes.

2. Newly Applied Content

1. N1310 *Requiring **signed char** to have no padding bits*
2. N1311 *Initializing static or external variables*
3. N1316 *Conversion between pointers and floating types*

Actually added constraint to 6.5.4 *Cast operators* instead of 6.3.2.3 *Pointers* since conversions between floats and pointers require an explicit cast and there were already constraints in that subclause (and no convenient way to add them in 6.3.2.3).

4. N1319 *Adding **EPOLE** to math library functions* (modulo change in minutes N1346)

I did not make any of the changes to 7.12 since the successful vote recorded in the minutes was only to add the description.

5. N1320 *Integrating C89 Defect Report 25 into C1x* (modulo N1346)

Actually inserted after paragraph 4 rather than paragraph 3 since it reads better that way. Also significantly reworded.

6. N1321 *Split **FLT_EVAL_METHOD** into operations and constants* (modulo N1346)

7. N1326 *Adding TR 19769 to the C Standard Library*

Required extensive editing. N1326 left out vital semantics that were only implied in TR 19769. In addition, TR 19769 omitted some important semantics:

- What if a character constant maps to multiple UTF-16/32 characters? I opted to call it implementation-defined.
- Should **mbstate_t** and **size_t** be declared in `<uchar.h>`? I opted to do so.
- Should the **ps** parameter work the way it does for the restartable multibyte/wide character conversion functions in 7.25.6.3? I opted to make it so.
- Do we need to add a normative reference to The Unicode Standard or is the existing reference to ISO/IEC 10646 sufficient?
- For **mbrtoc16/mbrtoc32**: If there are pending characters from a previous call and a new call is made with **s** pointing to a different string, what happens? What about if **s** is NULL? What if **pc16/pc32** is NULL?

In my opinion, the feature test macros should be moved to 6.10.8.

Should the **wchar_t** encoding be implementation-defined if it's not ISO/IEC 10646, too?

8. N1327 *Abandoning a Process (adding **quick_exit** and **at_quick_exit**)* (modulo N1346)

I omitted the statement that **at_quick_exit** is thread-safe since we haven't adopted threads yet and I presume we will need to add that note to lots of other functions, too, when we do.

For consistency, I generalized the agreed-to prohibition against `quick_exit` handlers calling `exit` and the existing prohibition against `exit` handlers calling `exit` by making it undefined for a program to call either `exit` or `quick_exit` more than once, or to call both of them. I also extended the current prohibition against `longjmp` out of an `exit` handler to apply to `quick_exit` handlers, too.

Conversely, I added `quick_exit` to the list of functions that can be safely called from an asynchronous signal handler.

I replicated the footnote from `at_quick_exit` noting that `exit` and `quick_exit` handlers are distinct under `atexit` as well.

There is a conflict between the description of `quick_exit`, which says that file buffers are not flushed, and the description of `_Exit`, which says that whether file buffers are flushed is implementation-defined.

9. N1330 *Static Assertions* (modulo N1346)

Is the exception of characters not in the basic source character set appropriate for C?

For the footnote in 6.6 about uses of integer constant expressions (102), I made the list non-exhaustive and indexed all the uses.

The committee did not discuss where to define the `static_assert` macro; I opted to do so in `<assert.h>`. We may want to discuss whether it should go elsewhere instead.

10. N1338 *More Thoughts on Implementing `errno` as a Macro*

11. N1346 *Draft Minutes for September 2008 (3.2 Report of the Project Editor)* [Update Annex C (Sequence Points) to match revised text]

3. Still Pending

1. N1333 *Unicode and Raw String Literals*

This will require extensive editing to integrate with the changes from TR 19769 and to remove C++-isms. I also note that additional changes are needed to separate *universal-character-name* from *escape-sequence* in addition to those in the paper.

4. Previously Applied Content

1. N1252 *A finer-grained specification of sequencing*
2. N1282 *Clarification of Expressions*
3. N1285 *C++ parallel memory model* (factored approach)
4. N1300 *Draft minutes for April 2008 9.19 N1271*

5. N1336 Errata

1. In 6.10.8, the `__STDC_MB_MIGHT_NEQ_WC__` macro was erroneously placed in paragraph 1 (required macros) rather than in paragraph 2 (optional macros).

6. My comments

1. I have added a description of the diff marks to the Abstract for those who are not familiar with them.
2. In the description of the lists of type specifiers in 6.7.2p2, I have changed “more than one multiset on a line” to “more than one multiset per item” since some of the lists now occupy multiple lines.
3. In accordance with discussions on the WG14 reflector, I have clarified the return behavior of the various `scanf` functions.

7. Open Issues

1. We currently have a definition of a pole error and **EPOLE** but no references to either one. I presume we'll want to make *some* kind of changes to the math library functions to indicate where pole errors might occur.
2. The decisions I made with respect to TR 19769 should be reviewed (and, hopefully, endorsed) by the committee and the open questions addressed.
3. The decisions I made with respect to **quick_exit** should be reviewed by the committee and the conflict between the descriptions of **quick_exit** and **_Exit** should be resolved.
4. The decisions I made with respect to static assertions should be reviewed by the committee.
5. We don't seem to have been consistent in which "auxiliary" headers are included in the library function synopses, especially in **<wchar.h>**. For example, the synopsis for **wctob** (7.25.6.1.2) includes **<stdio.h>** even though it is not required for the function declaration but only to define **EOF** which the function can return. Interestingly enough, the synopsis for **btowc** (7.25.6.1.1) also includes **<stdio.h>** even though EOF is only mentioned, it is not part of the interface. On the other hand, the synopsis for **swscanf** (7.25.2.4) does *not* include **<stdio.h>**, even though it's a possible return value, just like **wctob**.

I think we need to decide on a general policy and then make a pass through the library section to determine where to add or remove headers.

--

Larry Jones