1. Status

N1539 is the revised draft incorporating the changes from the Batavia meetings for SC22 ballot. Accordingly, it does not contain diff marks. N1547 is the same, but with diff marks from the previous draft (N1516).

2. Newly Applied Content

1. N1491 Proposal from Defect Report 271
2. N1496 Minimal fix to complex divide
3. N1497 Clarification to fmod
4. N1502 Fix for DR 334 (comparison macros) (1st footnote only)
5. N1503 Fix for DR 300 (translation-time expression evaluation) (1st change only)
6. N1509 Optimizing away infinite loops
7. N1512 Exceptional conditions vs floating-point exceptions
8. N1513 N1181 follow up
9. N1514 Conditional normative status for Annex G
10. N1515 Fix for Annex F pow special case
11. N1517 Editor’s Report (Non-editorial issues 3, 4, 5, and 6)
12. N1518 Recommendations for extended identifier characters for C and C++
13. N1521 Threads API Improvements and Issues (2nd and 3rd sections only)
14. N1526 Atomic C1x/C++0x Compatibility Refinements (1st part only)
15. N1530 Atomic Bitfields Implementation Defined
16. N1532 Small fix for the effect of alignment on struct/union type compatibility
17. N1533 Updates to memchr from POSIX
18. N1537 Synthesis re _Atomic
19. N1531 Clarifications for wide evaluation

After further review of the history of the current wording, I convinced myself that I was mistaken and that the intent of the committee was for all casts to remove excess range and precision, not just non-widening casts as I had originally thought. Accordingly, I applied the original changes from N1531 rather than the approved revised changes from N1538 that were based on my incorrect assertion of the committee’s intent.

3. Previously Applied Content

1. N1252 A finer-grained specification of sequencing
2. N1282 Clarification of Expressions
3. N1285 Extending the lifetime of temporary objects (factored approach)
4. N1300 Draft minutes for April 2008 9.19 N1271
5. N1310 Requiring **signed char** to have no padding bits
6. N1311 Initializing static or external variables
7. N1316 Conversion between pointers and floating types
8. N1319 Adding **EPOLE** to math library functions (modulo change in minutes N1346)
9. N1320 Integrating C89 Defect Report 25 into C1x (modulo N1346)
10. N1321 Split **FLT_EVAL_METHOD** into operations and constants (modulo N1346) [subsequently removed by N1361]
11. N1326 Adding TR 19769 to the C Standard Library
12. N1327 Abandoning a Process (adding **quick_exit** and **at_quick_exit**) (modulo N1346)
13. N1330 Static Assertions (modulo N1346)
14. N1338 More Thoughts on Implementing **errno** as a Macro
15. N1346 Draft Minutes for September 2008 (3.2 Report of the Project Editor) [Update Annex C (Sequence Points) to match revised text]
16. N1349 Parallel memory sequencing model proposal
17. N1350 Analyzability (#1, #4 - conditionally normative)
18. N1353 **FLT_EVAL_METHOD** issues (first change only)
19. N1356 **Bool** bit-fields
20. N1357 **tgamma** range error (first way)
21. N1358 Extensions to the C1X Library (#1, #2, #3 along the lines)
22. N1359 Technical corrigendum for C1X
23. N1360 Benign typedef redefinition
24. N1361 **FLT_EVAL_METHOD** and constants (back out N1321)
25. N1363 Editor’s Report (move unicode feature test macros, make **wchar_t** encoding imp-def if not 10646, and remove **quick_exit** footnote)
26. N1364 Thread-local storage (plus make auto access imp-def just like thread local)
27. N1365 Constant expressions (second bullet except “are covered” rather than “are not covered”)
28. N1367 Constructions and expression evaluation methods
29. N1371 Thread Unsafe Standard Functions (except for 21.5)
30. N1372 Threads for the C Standard Library
31. N1373 Wording improvements for **mblen**, **mbtowc**, and **cl6rtomb**
32. N1376 Adding pole error to math library functions
33. N1377 xxx **DECIMAL_DIG** macros for <float.h>
34. N1380 Minutes for Markham, March/April 2009 (4.29 WG14 Mail 11572)
35. N1381 **memset_s()** to clear memory, without fear of removal
36. N1382 **FLT_EVAL_METHOD** and return
37. N1383 LIA annex correction
38. N1384 xxx **TRUE_MIN** macros for <float.h>
39. N1387 Requested clarifications for thread-local storage (initialization only)
40. N1391 Floating-point to **int/Bool** conversions
41. N1394 Analyzability (along the lines)
42. N1396 Wide function return values (alternate proposal)
43. N1397 Adding Alignment Support to C (use **_Align** keyword rather than [[ ]] syntax)
44. N1398 Treatment of math error conditions
45. N1400 Headers not idempotent
46. N1406 Anonymous Member-Structures and -Unions (modulo “name lookup”)
47. N1420 On The Removal of **gets()**
48. N1371 Thread Unsafe Standard Functions (21.5 changes).
49. N1439 Completeness of types
50. N1441 Generic macro facility
51. N1444 Dependency Ordering for C Memory Model
52. N1447 C and C++ Alignment Compatibility
53. N1459 Comparison Macros (option 2)
54. N1460 Subsetting the Standard
55. N1462 **errno** and threads
4. Open Issues

1. The description of an atomic type in 6.2.5p20 leaves a lot to be desired. It should indicate that it is an object type and describe its unique properties rather than the syntax used to declare it.

2. There is confusion in 7.25 as to whether cond_t, tss_t, and mtx_t are identifiers for their objects or the objects themselves. On most implementations, I would expect that they would be the objects themselves, but I don’t think we want to require that.

   Assuming they are identifiers, the descriptions of the various create functions should be modified along the lines of the thrd_create function to note that an identifier can be reused once the object has been destroyed.

3. In 7.25.4.4p3 (mtx_timedlock), the value thrd_busy should be removed from the possible return values since the function blocks until it either succeeds or times out.

4. ISO 4217, Codes for the representation of currencies and funds, ISO 8601, Data elements and interchange formats — Information interchange — Representation of dates and times, ISO/IEC 9945, Information technology — Portable Operating System Interface (POSIX), and ISO/IEC 10646, Information technology — Universal Multiple-Octet Coded Character Set (UCS) have been updated. Benito is investigating our referencing the newer editions.

5. The index probably needs work, particularly for the newly added material.

---

Larry Jones