

June 25, 2010

## 1. Status

N1494 is a revised draft incorporating the changes from the Boulder meeting plus a few more corrections and editorial changes. It contains diff marks for all of the changes from N1425 except for minor formatting changes and subclause number changes.

## 2. Newly Applied Content

1. N1371 *Thread Unsafe Standard Functions* (21.5 changes).  
Change applied to 7.27.1, 7.28.6.3, and 7.28.6.4.
2. N1439 *Completeness of types*
3. N1441 *Generic macro facility*
4. N1444 *Dependency Ordering for C Memory Model*
5. N1447 *C and C++ Alignment Compatibility*
6. N1459 *Comparison Macros* (option 2)
7. N1460 *Subsetting the Standard*  
Significant editing
8. N1462 *errno and threads*  
Significant editing
9. N1464 *Creation of complex value* (move static init to Recommended Practice and make "as if" a Note)
10. N1468 *Assumed types in F.9.2* (along the lines - change font)
11. N1472 *Comparison Macro Argument Conversion* (option 3)  
Also J.1.
12. N1478 *Supporting the 'noreturn' property in C1x*  
Also J.2.
13. N1480 *Updates to C++ Memory Model Based on Formalization*
14. N1482 *Explicit Initializers for Atomics* (use data race words from N.1 for N.2, remove "where it applies" from N.1)
15. N1485 *Atomic proposal* (minus ternary op)

I omitted the note from modification 6 since it's covered by the reference to Compound Assignment. Modification 9 is actually to 6.7 *Declarations* rather than to 6.7.2 and the *init-declarator-list* should not be optional in this case (the constraint requires at least one declarator per declaration).

In modification 10, the added "constraint" is not a constraint (it merely specifies undefined behavior) and properly belongs in 6.5.2.3 *Structure and union members* instead.

Also J.2.

16. N1486 **fabs** (F.3 change only)  
Also clarified following bullet (unary minus).
17. N1487 *Comparison Macro Argument Conversion*
18. N1488 *UTF-8 string literals*
19. *E-mail 12207* (fix to 6.7p2 for `static_assert` declarations)
20. *dr269fix*

### 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 *Contractions 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 **c16rtomb***
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`* ()

#### 4. Other changes and comments

1. Increased the depth of the Table of Contents for some of the library subclauses and annexes as appropriate.
2. Defined the term *observable behavior* in 5.1.2.3 as suggested by Tom Plum in e-mail based on WG21 having adopted the same term. It seems like a handy thing to have available.
3. Augmented the definition of automatic storage duration in 6.2.4 to include compound literals that have automatic storage duration but not declarations.
4. Revised the definition of lvalue in 6.3.2.1 once again since the old definition included values like 42 that are clearly not supposed to be lvalues.
5. Move the notation description from 7.16.6 to 7.16.1 since we’ve added earlier material that uses the same notation.
6. Revised the wording in J.1 to better match the wording in 6.2.6.1.

#### 5. Open Issues

1. N1485’s addition to 6.7 Declarations is almost certainly incorrect as it makes no provision for storage class specifiers or function specifiers. This needs to be rethought.  

The prohibition against accessing members of an `_Atomic`-qualified structure or union would seem to make them rather difficult to use. Should this say that undefined behavior *may* occur rather than that it *does* occur?
2. The index probably needs work, particularly for the newly added material.
3. The list of changes from the previous edition needs to be populated.
4. The library synopses need to be examined and extraneous headers removed.