Editorial review comments from CFP
August 15, 2022

Below are comments from CFP reviewers for the C23 review draft N3047 (August 4). These comments have been reviewed (and in some cases, modified) by CFP. Comments are listed by initial submitter. Comments are colored:

red to indicate “mismatch with voted-in paper”

blue to indicate “editorial”

green to indicate “non-editorial”

- Jim Thomas

Rajan Bhakta:

Clause 4#7:
"and the strto * floating-point" -> "and the strto* floating-point"
(There should be no space between the strto and the *)

7.24.1.3#2
Missing end of paragraph line saying "Use of these functions with any other format string results in undefined behavior." from N2314 (TS 18661-1 for C2X integration).

7.24.1.3#3
The last line has “both nonnegative and” which is not in N2314. These words are not in 7.24.1.4#3, consistent with N2341 (TS 18661-2 for C2X integration). If the words are needed in 7.24.1.3#3 they should also be added in 7.24.1.4#3.

7.24.1.5#3 last bullet:
Seems to have the grammar term "n-char-sequence:" on the same line instead of a new line.

Mike Cowlishaw:

5.2.4.2.3, para 3: hyphen in '+ or - operator' should be a minus sign in text (as used in previous paragraph). See http://speleotrove.com/decimal/decifaq2.html#minuses.

5.2.4.2.3, para 10, Preferred quantum exponents table: formatting should not be justified in the 'fmin, fmax ...' operation column.

5.2.4.2.3, para 10, Preferred quantum exponents table: hyphen should be minus in the 'd32sub, d64sub' operation column

5.2.4.2.3, para 10, Preferred quantum exponents table: hyphen should be minus in the '-(x), +(x)' operation column

5.2.4.2.3, para 10, Preferred quantum exponents table: formatting should not be justified in the '*res returned by ...' operation column.

Fred Tydeman:

Page 25, #21: "The signaling NaN macros ... each is defined" ->
"Each of the signaling NaN macros ... is/are defined"

Page 261, 7.12.11.5 The nextup functions
Change
If x is the positive number (finite or infinite) or maximum magnitude in
to
If x is the positive number (finite or infinite) of maximum magnitude in

Page 262, 7.12.11.7 The canonicalize functions
In the Synopsis, three of the 'cx' are missing the '*'.

Page 467: Bottom of page: I see two lines on top of each other.

Page 604: What are '__suffix__' and '__prefix__' that only appear in J.6.1 Rule based identifiers and the index?

Pages 671...: There are several suffix terms in the index that are listed as prefix terms. One example is _DECIMAL_DIG.
Also, these index entries (at least):
_DECIMAL_DIG identifier prefix, 29, 329, 352,
_DIG identifier prefix, 550, 601
_EPSILON identifier prefix, 474, 551, 601
X_DECIMAL_DIG identifier prefix, 550, 615
X_DIG identifier prefix, 550, 615
X_EPSILON identifier prefix, 551, 615
should be changed to suffix.

Page 692: There are 2 index entries for IEC 60559. I assume that they should be merged together.

Pages 163-165: 6.10.3.1 #embed preprocessing directive
#1: limit embed parameter (??),
#10: CHAR_BIT (??env-consider-characteristics-of-integer-

Page 169: 6.10.3.4 prefix parameter
#4: <string.h> and <assert.h> are different colors.

Page 508: F.3 Operations
#13: IEEE 60559 -> IEC 60559

Page 544: H.1 Introduction
#1: ISO/IEC/IEEE 60559 -> IEC 60559

Page 669: M.4 Second Edition
IEEE arithmetic -> IEEE 754 arithmetic

Page 670: Bibliography
Add IEEE 754-1985 IEEE Standard for Binary Floating-Point Arithmetic
Add IEEE 754-2019 IEEE Standard for Floating-Point Arithmetic
[Both are referenced in F.1]

Page 67: 6.4.7 Header names
#4: delimited by a { on the left and } on the right
  Should not have the {} be italic.

Page 665: M.1 Fifth Edition
This:
  removed obsolete sign representations and integer width constraints
  (so-called "2's complement");
seems like it should be:
  removed integer width constraints and obsolete sign representations
  (so-called "1's complement" and "sign-magnitude");

"integration of decimal floating-point technical specification TS 18661-4a" ->
"integration of mathematical functions technical specification TS 18661-4a"

Page 666: "added $ and $ into the source and execution character set;"
"$ and $" should be "@, $, and ` (backtick)"

Jim Thomas:

JT-028.1 H.3#3 In the text for N and N + 1, a comma is needed between “type” and “evaluate”. as in H.3#2.

JT-029 H.3#6: Remove the line “The macro”.

JT-030 H.3#6, 2nd sentence: The spacing around the “-“ symbol looks wrong.

JT-032 H.9 and throughout: I don’t understand the color scheme, but some things seem inconsistent. For example, in 7.3.5 in cacosfNx the “x” is part of the function name so should have the same color as “cacosf”. Similarly, in _FloatNx the “x” is part of the type name so should have the same color as “_Float”. Is the coloring still in progress? There are further comments about color, where it seems obviously wrong, but my review of color is spotty.

JT-032.1 throughout (This comment elaborates on JT-032) In the “Nx” and “NX” suffixes for types, functions, and macros the “x” and “X” should have the same typeface and as the unsuffixed name. It is only the “N” that is template. Note that the typefaces in the similar “Mx” and “MX” prefixes are correct, e.g. in the macros in H.11.1#5 and in the 7.12.14 functions

JT-050 H.11.3.1.1#2 and #3: The “N” in encodefN is not italicized (both paragraphs).

JT-051 H.11.3.1.2#2 and #3: The “N” in decodefN is not italicized (both paragraphs).

JT-052 H.11.3.2.1#2: The “M” and “N” in fMencfN are not italicized.

JT-053.1 H.11.3.2.1#2: The “N” in “encNptr" should not be bold.

JT-054.1 H.11.3.2.2#2: In the first sentence, the “M” and “N” in the two function names are not italicized. In three instances in the paragraph the type size indicator “N” is bold.

JT-056 H.12: In many instances in the subclauses of H.12 the type size indicator “N” appears to be bold.

JT-059 H.12.3.1: In the heading, there is a space before the “N” in the function name.

JT-063 H.12.4: In the first sentence “N” is not italicized.

JT-073 5.2.4.2.2#18 “Annex F” doesn’t have a link.

JT-075 6.2.5#17, footnote 50. The footnote is broken across two pages.

JT-076 various. There is inconsistent use of a space in front of the parentheses in “alignof (“ and “sizeof (“. I’m guessing the space comes from following the syntax in 6.5.3. The space is appropriate for the syntax, but it seems undesirable for uses of the operators.

JT-077 6.4.4.1#2 There is an undesirable line break in “(‘)”.

JT-078 6.5.1#5 Change “See 7.27 how such a macro …”. Maybe to “7.27 shows how such a macro …”.

JT-079 6.5.1.1#5 Since the example does not provide the type-generic macro required in 7.27 (for implementations with more than one rounding mode), it would be better to change “The cbrt type-generic macro …” to “A cbrt type-generic macro …”.

JT-080 5.2.4.2.3#10 The preferred quantum exponent for prefix and postfix increment and decrement operators is not specified. In the “Preferred quantum exponents” table, after the row for logb, insert the row:

postfix ++ operator, postfix -- operator, prefix ++ operator, prefix -- operator min(Q(x), 0)

JT-081 6.7.1#11 The last sentence has “see ??”.

JT-082 6.7.1#15 432000000 (base 10) = 11001101111111100110000000000 (base 2) = 1.100110111111110011*2^28 which is representable exactly in IEC 60559 float, contrary to what the example implies. The number 536900000 could be used instead of 432000000.

JT-083 6.7.2.5#5 Change “operation” to “operator” and change “operations” to “operators”.

JT-084 6.7.2.5#5 Change the last sentence to “The typeof operator preserves all qualifiers.”

JT-085 6.7.6.3#4 Needs space after the form.

JT-086 6.7.9#4 In the next to last sentence, “double” has incorrect typeface.

JT-087 6.7.9#5 “shadows” is used twice in the example, but its meaning is ambiguous and not defined.

JT-089 6.7.10#11 In the second bullet change “(positive or unsigned) zero” to “positive zero”. There is no unsigned zero in decimal floating types.

JT-090 7.27#14 Change “where where” to “where”.

JT-091 7.27#4 There is a spurious line break after “modf”.

JT-092 7.27#15 After the table there is an inadvertent list of macro names “acospi asinpi … dsqrt”.

JT-094 5.2.4.2.2#14 Change “… and of the library functions …” to “and of most of the library functions …”. Some of the library functions have fully specified result accuracy, e.g. fabs and creal.

JT-096 5.2.4.2.2#18 Problem 1: The wording is awkward or worse. The first part indicates two alternatives where the second alternative “perhaps" has an extra feature. But there are three values. Problem 2: A parenthetical remark says “(this does not imply conformance to Annex F)”. But what does matches IEC 60559 operations mean? Can the binding be something different from what’s in Annex F? It seems to offer a way to claim IEC 60559 support without saying what it is.
To address both problems, delete “(and perhaps, operations)” and delete "2 type matches an IEC 60559 format and operations”.

JT-097 5.2.4.2.2#26 Footnote 28 says "If the presence or absence of subnormal numbers is indeterminable, then the value is intended to be a positive number no greater than the minimum normalized positive number for the type.” The presence or absence of subnormal numbers being indeterminable is one of the problematic conditions that the now obsolescent type_HAS_SUBNORM macros attempted to characterize. Also, the direction given by the footnote is not helpful since it is clearly implied by the definition of the type_TRUE_MIN macros. This footnote should be removed. Note that the footnotes that attempted to clarify the specification of the type_HAS_SUBNORM macros have already been removed.

JT-098 H.12.4.2#2: Looks like a space is missing in the first sentence in “strtoencbindNfunctions”.

JT-099 H.13#6: In the last table “undefined” in the second column should be moved down to be on the same line as fmul(dc, d) in the first column.

JT-~~09~~101 7.23.6.1#8 and 7.31.2.1#8 Looks like there is extra space after “b,” before “o, u, x, X”.

JT-~~09~~102 7.23.6.1#8 and 7.31.2.1#8, in the a,A bullet, in the paragraph beginning “If the precision P is present …”, in the first sentence, the “p” should be italicized.

JT-~~09~~103 7.23.6.2#15 and 7.31.2.2#15. Looks like a space is missing in “characters(including …”.

JT-~~09~~104 7.24.1.5#3 In the fourth bullet, “n-char-sequence:” is misplaced.

JT-~~09~~105 7.24.1.5#4 There are spurious line breaks before and after the footnote anchor “356)”.

JT-~~09~~106 6.4.4.2#12 The NOTE contradicts 7.24.1.5#4 footnote 356. Change "In contrast, the numeric conversion functions in the strto family (7.24.1.5, 7.24.1.6) include the sign as part of the input value and convert and round the negated input.” to "In contrast, the numeric conversion functions in the strto family (7.24.1.5, 7.24.1.6, F.5) may include the sign as part of the input value and convert and round the negated input; Annex F requires this behavior.”

JT-~~09~~107 7.24.1.6#3 and 7.31.4.1.3#3 In the third bullet, “d-char-sequence:” is misplaced.

JT-~~09~~108 7.24.1.6#4 and 7.31.4.1.3#4 In the first sentence after the bullets remove the parentheses from “(before rounding)”. The containing parentheses suggest that the contained text is already implied. However, the strtod specification which does not include the parenthetical remark does not imply negation before rounding, as footnote 356 clarifies.

JT-109#1 In 5.2.4.2.3, in the second sentence change: “An implementation that does not support decimal floating types shall not provide these macros.” to “An implementation shall provide these macros if and only if defines __STDC_IEC_60559_DFP__.” This responds to recent email discussion about a need for something about __STDC_IEC_60559_DFP__ in 5.2.4.2.3.

JT-110 throughout Check that a hyphen is not used where a minus sign is needed. For example, H.3#6, 2nd sentence.

JT-111 7.12.17 In several places in the subclauses it appears space is needed in front on a relational operator, e.g. in 7.12.17.1 #2.

Damian McGuckin:

7.6 Floating-point environment <fenv.h>
#12 Remove the word 'simply'.

7.3.2
#1 Replace “to” with "to do so" at the end of the first sentence. English sentences do not end with the word "to".

7.3.9.1
#2 Replace

(also called phase angle)

with

(also called phase)

The NIST uses the word “phase” (i.e. without the word angle).