Proposal for C2X
WG14 N2559

Title:	Update to IEC 60559 2020
Author, affiliation:	C FP group
Date:	2020-09-01
Proposal category:	Editorial
Reference:	N2478, N2531, N2532

The suggested changes below update draft C2X (N2478) to the latest version of the floating-point standard: IEC 60559:2020 (IEEE 754-2019). For background on the update see N 2531.

The suggested changes include:

- Updates to the references to the floating-point standard. (Change #1 - 3)
- An update to the introduction to Annex F. (Change #3)
- Corrected/sharpened qualifications. (Change #4, 5, 14)
- Changes to move bindings from the remarks following the binding table in F.3 #1 into the table itself. (Change #6 -13)
- Clarification about how C functions are represented in the binding tables in F.3. (Change #5, 14)
- New IEC 60559 operations (corresponding to existing C functions) added in the binding table in F.3 #23. (Change #15)

N2532 includes changes to accommodate the new min-max functions in IEC 60559:2020.

Suggested changes:

1. Change 2 #6:

> ~~[6] IEC 60559:1989, Binary floating-point arithmetic for microprocessor systems (previously designated IEC 559:1989).~~

to:

> [6] ISO/IEC 60559:2020, *Floating-point arithmetic*.

2. Change footnote 23:

> 23) IEC 60559~~:1989~~ specifies quiet and signaling NaNs. For implementations that do not support IEC 60559~~:1989~~, the terms quiet NaN and signaling NaN are intended to apply to values with similar behavior

3. Change F.1 #1-2:

> ~~[1] This annex specifies C language support for the IEC 60559 floating-point standard. The IEC 60559 floating-point standard is specifically Floating-point arithmetic (ISO/IEC/IEEE 60559:2011), also designated as IEEE Standard for Floating-Point Arithmetic (IEEE 754–2008). The IEC 60559 floating-point standard supersedes the IEC 60559:1989 binary arithmetic standard, also designated as IEEE Standard for Binary Floating-Point Arithmetic (IEEE 754–1985). IEC 60559 generally refers to the floating-point standard, as in IEC 60559 operation, IEC 60559 format, etc.~~
>
> ~~[2] The IEC 60559 floating-point standard specifies decimal, as well as binary, floating-point arithmetic. It supersedes IEEE Standard for Radix-Independent Floating-Point Arithmetic (ANSI/IEEE 854–1987) which generalized the binary arithmetic standard (IEEE 754-1985) to remove dependencies on radix and word length.~~

to:

> [1] This annex specifies C language support for the IEC 60559 floating-point standard. The *IEC 60559 floating-point standard* is specifically *Floating-point arithmetic* (ISO/IEC 60559:2020), also designated as *IEEE Standard for Floating-Point Arithmetic* (IEEE 754–2019). *IEC 60559* generally refers to the floating-point standard, as in IEC 60559 operation, IEC 60559 format, etc.
>
> [2] The IEC 60559 floating-point standard is a minor upgrade to IEC 60559:2011 (IEEE 754-2008). IEC 60559:2011 was a major upgrade to IEC 60559:1989 (IEEE 754–1985), specifying decimal as well as binary floating-point arithmetic.

4. Change footnote 380:

> 380)Since NaNs created by IEC 60559 arithmetic operations are always quiet, quiet NaNs (along with infinities) are sufficient for closure of the arithmetic.

5. In F.3 #1, change the first sentence:

> [1] C operators, functions, and function-like macros provide ~~the~~ operations ~~required~~specified by IEC 60559 as shown in the following table. In the table, C functions are represented by the function name without a type suffix.

6. In the operation binding table in F.3 #1, add:

getPayload	**getpayload**	F.10.13.1
setPayload	**setpayload**	F.10.13.2
setPayloadSignaling	**setpayloadsig**	F.10.13.3

7. In the operation binding table in F.3 #1, add:

quantize	**quantize**	7.12.15.1
sameQuantum	**samequantum**	7.12.15.2
quantum	**quantum**	7.12.15.3

8. In the operation binding table in F.3 #1, add:

encodeDecimal	**encodedec**	7.12.16.1
decodeDecimal	**decodedec**	7.12.16.2
encodeBinary	**encodebin**	7.12.16.3
decodeBinary	**decodebin**	7.12.16.4

9. Delete F.3 #9:

> ~~[9] The C **getpayload**, **setpayload**, and **setpayloadsig** (F.10.13) functions provide program access to NaN payloads, defined in IEC 60559.~~

10. Delete F.3 #14:

> ~~[14] The **quantized***N* functions (7.12.15.1) provide the quantize operation defined in IEC 60559 for decimal floating-point arithmetic.~~

11. Delete F.3 #18:

> ~~[18] The **samequantumd***N* functions (7.12.15.2) provide the sameQuantum operation defined in IEC 60559 for decimal floating-point arithmetic. 1~~

12. Change F.3 #20:

> [20] The ~~**quantumd***N* (7.12.15.3) and~~ **llquantexpd***N* ~~(7.12.15.4)~~ functions (7.12.15.4) compute ~~the quantum and~~ the (quantum) exponent *q* defined in IEC 60559 for decimal numbers viewed as having integer significands.

13. Delete F.3 #21-22:

> ~~[21] The **encodedecd***N* (7.12.16.1) and **decodedecd***N* (7.12.16.2) functions provide the encodeDecimal and decodeDecimal operations defined in IEC 60559 for decimal floating-point arithmetic.~~
>
> ~~[22] The **encodebind***N* (7.12.16.3) and **decodebind***N* (7.12.16.4) functions provide the encodeBinary and decodeBinary operations defined in IEC 60559 for decimal floating-point arithmetic.~~

14. In F.3 #23, change the first sentence:

> [23] The C functions in the following table provide mathematical operations recommended by IEC 60559 ~~and similar operations~~. The C functions are represented by the function name without a type suffix.

15. In the binding table in F.3 #23, change the three rows:

tanPi	**tanpi**	7.12.4.14, F.10.1.14
asinPi	**asinpi**	7.12.4.9, F.10.1.9
acosPi	**acospi**	7.12.4.8, F.10.1.8