Proposal for C2Y
WG14 N3779

Title:	Annex F floating-point environment updates
Author, affiliation:	C FP group
Date:	2026-01-08
Proposal category:	Technical
Reference:	N3685

Background

C99 added subclauses F.8.3 "Translation" and F.8.4 "Execution" as part of support for IEEE 754-1985. These subclauses have remained unchanged though they are out of date with respect to revisions of the floating-point standard.

Issue 1: obsolete modes

The original floating-point standard IEEE 754-1985 allowed implementations that deliver results only in formats wider than the single (float32) or double (float64) basic formats. To avoid double rounding (e.g. in `float_z = float_x / float_y`), the floating-point standard required those implementations to provide a rounding precision mode to round to the precision of the narrower supported formats.

IEEE 754-2008 and subsequent revisions have required operations with operands and results in the supported basic formats. Rounding precision mode was deemed unnecessary and removed from the standard entirely.

C2Y continues to say that rounding precision mode is a feature of ISO/IEC 60559 (IEEE 754), though rounding precision mode is no longer mentioned in the floating-point standard.

A similar situation pertains to trapping mode, which was an optional feature in IEEE 754-1985 and removed in IEEE 754-2008. Current C also mentions a stopping mode as an ISO/IEC 60559 feature, though it never was. (Perhaps there was an assumption that a trap handler to stop execution would be available along with a trapping mode.)

The suggested changes below update the C2Y specification regarding these modes to align with current ISO/IEC 60559.

The suggested changes do not remove the footnote in F.4.1:

> 441)Dynamic rounding precision and trap enablement modes are examples of such extensions.

because the modes are examples of existing extensions.

The suggested changes below regarding obsolete modes do not invalidate existing implementations, because the changes accommodate the modes as extensions.

Issue 2: decimal floating point not accounted for

The current specification in F.8.3 and F.8.4 has not been updated to account for decimal floating point which was added to C in C23.

ISO/IEC 60559:2008 introduced a second "to nearest" rounding direction: "to nearest with ties away from zero". For decimal formats support for the new rounding direction is required and it is an implementation option to be the default rounding direction. For binary formats support for the new rounding direction is optional.

The second "to nearest" rounding direction is already acknowledged in 5.3.5.3.3 and 5.3.5.3.4. However, F.8.3 and F.8.4 simply refer to "rounding to nearest" which originally meant "to nearest with ties to even" but now appears to refer to two "to nearest" rounding directions. The suggested changes below avoid the collective "to nearest".

We think the suggested changes will not invalidate any existing implementations. The current specification, read with the collective meaning of "to nearest", would allow "to nearest with ties away from zero" as the default for standard floating types, which the suggestions below (and ISO/IEC 60559) do not allow; however, we think such an implementation is extremely unlikely.

Suggested Changes

In F.8.3 #1 change:

From:

> During translation, constant rounding direction modes (7.6.3) are in effect where specified. Elsewhere, ~~during translation the ISO/IEC 60559 default modes are in effect:~~
>
> ~~— The rounding direction mode is rounding to nearest.~~
>
> ~~— The rounding precision mode (if supported) is set so that results are not shortened.~~
>
> ~~— Trapping or stopping (if supported) is disabled on all floating-point exceptions.~~

To:

> During translation, constant rounding direction modes (7.6.3 and 7.6.4) are in effect where specified. Elsewhere, the dynamic rounding direction modes are in effect with the default values prescribed in ISO/IEC 60559:
>
> — "to nearest with ties to even" (represented by the macro `FE_TONEAREST`) for operations for standard floating types.
>
> — Either "to nearest with ties to even" or "to nearest with ties away from zero" (represented by the macros `FE_TONEAREST` and `FE_TONEARESTFROMZERO`, respectively) for operations for decimal floating types. Which of these two rounding directions is the default is implementation-defined.

Recommended practice

The default for the rounding direction mode for decimal floating types should be “to nearest with ties to even”, as recommended by ISO/IEC 60559.

In F.8.4 #1 change:

From:

At program startup the dynamic floating-point environment is initialized as prescribed by ISO/IEC 60559:

— All floating-point exception status flags are cleared.

— The dynamic rounding direction mode ~~is rounding to nearest.~~

~~— The dynamic rounding precision mode (if supported) is set so that results are not shortened.~~

~~— Trapping or stopping (if supported) is disabled on all floating-point exceptions.~~

To:

At program startup the dynamic floating-point environment is initialized as prescribed by ISO/IEC 60559:

— All floating-point exception status flags are cleared.

— The dynamic rounding direction modes have the defaults described in F.8.3.