ISO/IEC/JTC 1/SC 22/WG 23 N1216
19 October 2022

Comments on N1214 by J. Reid

4.6 Numeric model

Separate the two sentences into two paragraphs and insert this new para between them:
"Most Fortran processors support ISO/IEC/IEEE 60559:2011, the IEEE standard for floating-point arithmetic, but perhaps not entirely. It defines binary patterns that represent floating-point values, signed zeros, and signed infinities; other values are NaNs (Not a Number). This allows IEEE arithmetic to be closed, that is, every operation has a result. If an exception occurs, execution continues with the corresponding flag signaling, and the flag remains signaling until explicitly set quiet by the program. The flags are therefore called **sticky** . The flags are **overflow**, **divide_by_zero**, **invalid** (for example 0.0/0.0 or when an operand is a NaN), **underflow**, and **inexact** (when the result cannot be represented exactly). There are five corresponding Fortran exception flags. Each has a value that is either **quiet** or **signaling** and its initial value is quiet. There are procedures for finding and for resetting the value of a flag. If a flag is signaling on entry to a procedure, the processor will set it to quiet on entry and restore it to signaling on return. This allows exception handling within the procedure to be independent of the state of the flags on entry, while retaining their 'sticky' properties. " [Needed in 6.36 and perhaps elsewhere.]

6.13.2 Guidance to language users

Replace final bullet by
"Use initialization in the declaration of any pointer so that its association status can always be determined." [Rationale added and I see no reason for it to apply only to pointers with the save attaribute.} and delete the comment.

6.36.1 Applicability to language

In para 2, sentence 2, replace "status error values that are not requested" by "status error values not being requested" [Grammar!]

Replace para 3 and its Note by
"The intrinsic module `ieee_exceptions` is defined by the standard for the support of floating-point exceptions (Clause 4.6) and is provided by most processors. Accessing this module allows the program to test the Fortran flags.

Fortran does not support exception handling of the kind described in ISO IEC 24772-1 subclause 6.36.3 para 4. For each of the Fortran flags, some processors allow control during program execution of whether to halt image execution or continue after the flag is raised. Halting is not precise and may occur any time after the exception has occurred.

Fortran does not support detection of integer overflow, but some compilers have an option for detecting it."

6.59 Concurrency – Activation [CGA]

In line 1, change comma to semicolon.
In lines 203, change "during the general execution of a construct" to "during general execution".

In para 2, change “Failure of an image causes silent termination of that image.” To “Failure of an image can be detected by an executing image by executing one of the intrinsic functions `failed_images` and `image_status`, or by examining the status variable after executing a statement that involves access to data on another image.”

6.60 Concurrency – Directed termination [CGT]

Delete the final sentence. [It is wrong – `error stop` causes all images to terminate.]

6.61.1 Applicability to language

In para 1, change “Only coarray data” to “Data” so that the sentence reads
“Data are accessible across image boundaries and only by using image selectors in square brackets.” [Consider `a[image]%pointer`]
In para 2, delete “coarray”. [Consider `a[image]%pointer`]

6.61.2 Guidance to language users

Bullet 1. Delete comment [On reflection, I accept this guidance.]

Bullet 3, sub-bullet 3. Change “collective subroutines” to “a collective subroutine”.