

ISO/IEC 1539-1:2023 - TECHNICAL CORRIGENDUM 1
(DRAFT ANNOTATED VERSION)

Notes for WG5:

Edits are included in this document from interpretations:

f23/003, f23/004, f23/005, f23/006, f23/008, f23/009, f23/010, f23/011, f23/012, f23/013, f23/015, f23/017, f23/018.

Interpretations approved without generating edits:
f23/016.

Page and line numbers on the right refer to 24-007. As some interpretations referred to a different base document, page numbers below may be different from those in N2241. They, the interpretation references and notes in italics, are for WG5 use only and will be deleted before the document is submitted to ISO.

[xiv] f23/006

Introduction

In the second paragraph, at the end of bullet point “Intrinsic modules”, append the sentence:

“The result of the function IEEE_SCALB from the intrinsic module IEEE_ARITHMETIC has been corrected to conform to the IEEE standard.”

[xiv] f23/008

In the second paragraph, at the end of bullet point “Changes to the intrinsic module IEEE_ARITHMETIC for conformance with ISO/IEC 60559:2020”, append the sentence:

“The function IEEE_SCALB from the intrinsic module IEEE_ARITHMETIC now performs the scaleB operation.”

[34:1+] f23/015

Subclause 4.3.3

After the fourth paragraph of the subclause, insert a new paragraph:

“Fortran 2018 permitted a *component-data-source* in a structure constructor to be a coindexed object with a pointer subcomponent. This document does not permit such usage.”

[34:13+] f23/006

In the fifth paragraph of the subclause, append a new bullet point:

- “• The result of a reference to the function IEEE_SCALB from the intrinsic module IEEE_ARITHMETIC has been corrected to return the representable number having a magnitude nearest to $|X \times 2^l|$ with the same sign as X , if $X \times 2^l$ is too small to be represented with full accuracy.”

[35:8+] f23/015

Subclause 4.3.4

After the fourteenth paragraph of the subclause, insert a new paragraph:

“Fortran 2008 permitted a *component-data-source* in a structure constructor to be a coindexed object with a pointer subcomponent. This document does not permit such usage.”

[35:17+] f23/006

In the final (fifteenth) paragraph of the subclause, append a new bullet point with text identical to the new bullet point in subclause 4.3.3 above.

[36:13+] f23/006

Subclause 4.3.5

In the final (thirteenth) paragraph of the subclause, append a new bullet point with text identical to the new bullet point in subclause 4.3.3 above.

[49:26,27] f23/018

Subclause 5.4.7

In the third paragraph of the subclause, change “For each coarray” to “For each established coarray” and after “in which it is established (5.4.8).” insert the new sentence:

“For each unallocated coarray, there exists a corresponding unallocated coarray with the same declared type, rank, corank, and non-deferred type parameters on each active image of the current team.”

[49:27] f23/018

Insert a new paragraph in between the above insertion and the rest of what was paragraph 3:

“For a named coarray that is not a dummy argument, its corresponding coarrays are the ones with the same name in that scoping unit. For a coarray that is a component at any level of component selection, its corresponding coarrays are the same components of the base object that has the same name in that scoping unit. If a coarray component is a potential subobject component of an array element, the array element for its corresponding coarrays has the same position in array element order on each image.”

[49:30] f23/018

In the fourth paragraph of the subclause, after “The set of corresponding” insert “established”, making the whole sentence read:

“The set of corresponding established coarrays on all images in a team is arranged in a rectangular pattern.”

[79:31-32] f23/009

Subclause 7.5.4.1

Delete constraint C753. which begins “C753 A data component whose type has a coarray”.

[80:0+] f23/009

After NOTE 1 insert new NOTE 1.5”

“NOTE 1.5

A data component whose type has a coarray potential subobject component cannot be a coarray or a pointer, see constraint C825.”

[93:21+] f23/015

Subclause 7.5.10

In the first paragraph of the subclause, after the penultimate constraint C7109, insert new constraint:

“C7109a (R758) If *expr* is a coindexed object, it shall not have a pointer component at any level of component selection.”

[93:23-] f23/015

After NOTE 1, insert new NOTE:

“NOTE 1a

Although a coindexed object with a pointer subcomponent is not the only way for the structure constructor to produce a value with an undefined pointer subcomponent, copying a pointer from another image is particularly likely to cause undiagnosed incorrect results, and thus precluded in this context.”

[95:18+] f23/013

Subclause 7.6.1

In the first paragraph of the subclause, in rule 762, after

R762 *enumerator* is *named-constant* [= *scalar-int-constant-expr*]

insert:

“ or *named-constant* [= *boz-literal-constant*]”

[96:5-9] f23/013

In the sixth paragraph of the subclause, in the numbered list following “The enumerator is a scalar named constant, with the value determined as follows.”, make the following changes.

Insert after (1):

“(1a) if *boz-literal-constant* appears, the enumerator has the value specified by `INT(boz-literal-constant, C_INT)`, where `C_INT` is from the intrinsic module `ISO_C_BINDING`.”

In the current (2) and (3), replace “If *scalar-int-constant-expr* does not appear” with “If neither *scalar-int-constant-expr* nor *boz-literal-constant* appears” such that the new list items read:

“(2a) If neither *scalar-int-constant-expr* nor *boz-literal-constant* appears and the enumerator is the first enumerator in *enum-def*, the enumerator has the value zero.

(3a) If neither *scalar-int-constant-expr* nor *boz-literal-constant* appears and the enumerator is not the first enumerator in *enum-def*, it has the value obtained by adding one to the value of the enumerator that immediately precedes it in the *enum-def*.”

[100:30] f23/013

Subclause 7.7

In the second paragraph of the subclause, in constraint C7119, after “variable of type integer or real” insert: “, as the value in an *enumerator*”.

[148:32-40] f23/018

Subclause 9.7.1.2

In the fourth paragraph of the subclause, delete the third sentence “If the coarray is a...” and delete to the end of the paragraph and insert “The coarray shall be corresponding (5.4.7) on those images.”.

Following this change, insert a new paragraph:

“If an allocation specifies a coarray, the same `ALLOCATE` statement shall be executed on every active image of the current team. If the coarray is an unsaved local variable of a recursive procedure, the execution of the `ALLOCATE` statement shall be at the same depth of recursion of that procedure on those images.”

[152:12] f23/012

Subclause 9.7.3.2

In the tenth paragraph of the subclause, in the final sentence change “it is” to “the corresponding coarrays are”, and insert “other” between “all” and “active”, making that whole sentence read:

“A coarray shall not become deallocated on an image unless the corresponding coarrays are successfully deallocated on all other active images in this team.”

In the following paragraph, which begins “If an *allocate-object* is a coarray dummy argument”, append the new sentence:

“If an *allocate-object* is a coarray subcomponent of a dummy argument, those components of the ultimate arguments on those images shall be corresponding coarrays.”

[167:27] f23/005

Subclause 10.1.6

In the second paragraph of the subclause, item (3), change “the type of d_2 is compatible with the dynamic type of x_2 ,” to “ d_2 is type-compatible with x_2 ,”

[168:13] f23/005

In the fifth paragraph of the subclause, item (3), change “the types of d_1 and d_2 are compatible with the dynamic types of x_1 and x_2 , respectively,” to “ d_1 and d_2 are type-compatible with x_1 and x_2 , respectively,”

[179:10] f23/005

Subclause 10.2.1.4

In the second paragraph of the subclause, item (3), change “the types of d_1 and d_2 are compatible with the dynamic types of x_1 and x_2 , respectively,” to “ d_1 and d_2 are type-compatible with x_1 and x_2 , respectively,”

[306:21+] f23/003

Subclause 14.2.2

In the second sentence of NOTE 4 delete:

“a *common-block-name* or”

Add the following sentence at the end of the note:

“Restrictions on *local-name* being the same as a *common-block-name* are detailed in 19.3.2.”

[423:22-27] f23/010

Subclause 16.9.147

In the third paragraph of the subclause, in the description for Argument FROM, append:

“If it is a coarray, it shall correspond to the FROM arguments in all corresponding invocations of MOVE_ALLOC.”

In the description for Argument TO, append:

“If it is a coarray, it shall correspond to the TO arguments in all corresponding invocations of MOVE_ALLOC.”

[428:0+] f23/011

Subclause 16.9.155

In the sixth paragraph of the subclause, in the entry “initialization for an object...” in Table 16.5, change “object” to “entity”, twice, making the whole entry read:

“initialization for an entity in a declaration | the entity”.

[429:12] f23/004

Subclause 16.9.157

In the third paragraph of the subclause, at the end of the description for Argument ROUND, change “shall be present only” to “shall appear only”.

[471:7] f23/008

Subclause 17.9

In the first paragraph of the subclause, in the last bullet point, after “logB,” insert “scaleB,” and after “IEEE_LOGB,” insert “IEEE_SCALB”.

[488:3] f23/008

Subclause 17.11.37

In the third paragraph of the subclause, in the description for Argument I, change “integer” to “integer or of type real with the same kind type parameter as X” so that the line reads:

“I shall be of type integer or of type real with the same kind type parameter as X.”

[488:9+] f23/008

In the sixth paragraph of the subclause, before the description for Result Value Case (i), insert:

“The value of the result shall conform to the scaleB operation of the IEEE standard.”.

[488:14] f23/006

In the same paragraph, in the description for Result Value Case (iii), change “ $|2^i|$ ” to “ $|X \times 2^i|$ ”.

[524:15] f23/017

Subclause 18.5.5.5

In the second paragraph of the subclause, in the description for the attribute parameter, append the new sentence:

“If it is CFI_attribute_other, base_addr shall not be a null pointer.”

[524:29-31] f23/017

In the third paragraph, after “for an unallocated allocatable” change the comma to “or”, and after “disassociated pointer” delete “, or is... data object”, making that sentence read:

“If `base_addr` is a null pointer, the established C descriptor is for an unallocated allocatable or a disassociated pointer.”

[535:15] *f23/003*

Subclause 19.3.1

In the second paragraph of the subclause after “is a common block name (19.3.2)” insert “and the local identifier is not that of a named constant or intrinsic procedure,”