

DRAFT ISO/IEC 1539-1:2010 - TECHNICAL CORRIGENDUM 4

Notes for WG5:

Edits included in this document from the interpretations in N2042, N2062, N2086 and N2092 as amended by N2047, N2080, N2087, N2093, N2094 and N2102:

f08/0099 to f08/0104, f08/0106, f08/0108, f08/0109, f08/0112, f08/0113, f08/0115 to f08/0124, f08/0126, f08/0127, f08/0129 to f08/0137, f08/0139 to f08/0145, f08/0147, f08/0148.

Interpretations from N2042, N2062, N2086 and N2092 approved without generating edits:
f03/0042, f08/0099, f08/0105, f08/0108, f08/0138.

Two interpretations (f08/0110 and f08/0128) from this group of documents were withdrawn and returned to J3 for further consideration. Some changes were made to the text of the balloted interpretations without affecting the resulting edits.

Page and line numbers on the right refer to 10-007r1. They, the interpretation references and notes in italics are for WG5 use only and will be deleted before the document is submitted to ISO.

[xv] f08/0121 (N2062)

Introduction

In the second paragraph, append to the ‘Data declaration’ bullet point: “A *defined-operator* can be used in a specification expression.”.

[xv] (new in N2102)

In the second paragraph, append two new sentences to the ‘Data usage and computation’ bullet point: “All transformational functions from the intrinsic modules IEEE_ARITHMETIC and IEEE_EXCEPTIONS can be used in constant expressions. All transformational functions from the intrinsic modules IEEE_ARITHMETIC, IEEE_EXCEPTIONS, and ISO_C_BINDING can be used in specification expressions.”

[xvi] f08/0131 (N2062&N2102)

In the second paragraph, in the ‘Intrinsic modules’ bullet point, before “The function C_SIZEOF” insert the new sentence:

A contiguous array variable that is not interoperable but which has interoperable type and kind type parameter (if any), and a scalar character variable with length>1 and kind C_CHAR, can be used as the argument of the function C_LOC in the intrinsic module ISO_C_BINDING, provided the variable has the POINTER or TARGET attribute.

[xvi] f08/0127 (N2062&N2080&N2102) and f08/0139 (N2092)

In the second paragraph, in the ‘Programs and procedures’ bullet point, replace the final sentence, “A line in the program is permitted to begin with a semicolon.” with “A free form continuation line can begin with zero or more blanks followed by a semicolon.”. Also, append the sentence, “The name of an external procedure that has a binding label is a local identifier and not a global identifier.”.

[6:7+] f08/0124 (N2062&N2102)

Subclause 1.3

After the definition of **parent component** (1.3.33.2) insert a new term:

1.3.33.2a

potential subobject component

nonpointer component, or potential subobject component of a nonpointer component

[24:11+] *f08/0147 (N2086) and f08/0141 (N2092&N2093)*

Subclause 1.6.2

After the six paragraphs added to the subclause in Technical Corrigendum 2, add the following new paragraphs:

Fortran 2003 interpreted assignment to an allocatable variable from a nonconformable array as intrinsic assignment, even when an elemental defined assignment was in scope; this part of ISO/IEC 1539 does not permit assignment from a nonconformable array in this context.

Fortran 2003 permitted a statement function to be of parameterized derived type; this part of ISO/IEC 1539 does not permit that.

[52:6+] *f08/0129 (N2062&N2080&N2102)*

Subclause 4.3.1.3

After the first paragraph, insert the following new paragraph:

Where a data entity other than a component is declared explicitly using the CLASS specifier to be of derived type, the specified derived type shall have been defined previously. If the data entity is a function result, the derived type may be specified in the FUNCTION statement provided the derived type is defined within the body of the function or is accessible there by use or host association. If the derived type is specified in the FUNCTION statement and is defined within the body of the function, it is as if the function result variable were declared with that derived type immediately following its *derived-type-def*.

[70:3] *f08/0145 (N2086)*

Subclause 4.5.4.6

In the first sentence of constraint C461, insert “, noncoindexed” after “nonallocatable” so that the sentence reads:

C461 (R443) The designator shall designate a nonallocatable, noncoindexed variable that has the TARGET and SAVE attributes and does not have a vector subscript.

[90:15] *f08/0115 (N2062&N2102)*

Subclause 5.3.4

In the first bullet point of the second paragraph, after “the variable” insert “is a dummy argument or” and insert a comma after “scoping unit”.

[102:9] *f08/0122 (N2062)*

Subclause 5.3.19

In constraint C560, after “for a coarray” insert “, or a variable with a coarray ultimate component,”.

[102:11] *f08/0122 (N2062)*

In constraint C561, after “for a coarray” insert “, or a variable with a coarray ultimate component,”.

[111:13-14] *f08/0101 (N2042)*

Subclause 5.6

Replace the second paragraph, by:

The order in which the values appear on output is the same as the order of the *namelist-group-objects* in the namelist group object list; if a variable appears more than once as a *namelist-group-object* for the same namelist group, its value appears once for each occurrence.

[119:13] *f08/0124 (N2062)*

Subclause 6.4.2

In constraint C617, replace “subcomponent” with “potential subobject component”.

[127:8-9]f08/0109 (N2086)

Subclause 6.7.1.1

In constraint C642, change “C_PTR,” to “C_PTR or” and delete “, LOCK_TYPE ... LOCK_TYPE”.

[127:9+]f08/0109 (N2086&N2087)

Following constraint C642, add a new constraint:

C642a (R627) If SOURCE= appears, the declared type of *source-expr* shall not be LOCK_TYPE or have a potential subobject component of type LOCK_TYPE.

[127:18-19]f08/0109 (N2086)

Instead of the edit in Technical Corrigendum 2, which replaced the entire fourth paragraph of the subclause, make the following change to this paragraph: replace “If *allocate-object* is” by “If an ALLOCATE statement has a SOURCE= specifier and an *allocate-object* that is”. The edited paragraph thus reads:

If an ALLOCATE statement has a SOURCE= specifier and an *allocate-object* that is a coarray, *source-expr* shall not have a dynamic type of C_PTR, C_FUNPTR, or LOCK_TYPE, or have a subcomponent whose dynamic type is LOCK_TYPE.

[128:15-17]f08/0130 (N2062&N2094&N2102)

Subclause 6.7.1.2

In the second sentence of the fourth paragraph, change “On each image” to “If no error condition other than STAT_STOPPED_IMAGE occurs” and change “all other images” to “all non-stopped images”. Append a new sentence so that the entire paragraph reads:

When an ALLOCATE statement is executed for which an *allocate-object* is a coarray, there is an implicit synchronization of all images. If no error condition other than STAT_STOPPED_IMAGE occurs, execution of the segment (8.5.2) following the statement is delayed until all non-stopped images have executed the same statement the same number of times. The coarray shall not become allocated on an image unless it is successfully allocated on all non-stopped images.

[129:8]f08/0133 (N2062)

Subclause 6.7.1.3

In the second bulleted item of the first paragraph, fourth sentence, insert “nonoptional” before “nonallocatable dummy argument”.

[131:17-19]f08/0130 (N2062&N2094&N2102)

Subclause 6.7.3.2

In the second sentence of the eleventh paragraph, change “On each image” to “If no error condition other than STAT_STOPPED_IMAGE occurs” and change “all other images” to “all non-stopped images”. Append a new sentence so that the entire paragraph reads:

When a DEALLOCATE statement is executed for which an *allocate-object* is a coarray, there is an implicit synchronization of all images. If no error condition other than STAT_STOPPED_IMAGE occurs, execution of the segment (8.5.2) following the statement is delayed until all non-stopped images have executed the same statement the same number of times. If the coarray is a dummy argument, its ultimate argument (12.5.2.3) shall be the same coarray on every image. The coarray shall not become deallocated on an image unless it is successfully deallocated on all non-stopped images.

[132:4]f08/0112 (N2042)

Subclause 6.7.4

Append to the first paragraph the sentence: “The *stat-variable* shall not depend on the value of the *errmsg-variable*.”.

[132:22]f08/0112 (N2042)

Subclause 6.7.5

Append to the first paragraph the sentence: “The *errmsg-variable* shall not depend on the value of the *stat-variable*.”.

[150:28+]f08/0104 (N2042&N2047&N2102)

Subclause 7.1.11

In the second paragraph after item (10), insert a new item:

- (10a) a reference to a transformational function from the intrinsic module IEEE_ARITHMETIC (14), IEEE_EXCEPTIONS (14), or ISO_C_BINDING (15.2), where each argument is a restricted expression,

[151:7-8]f08/0126 (N2062&N2102)

In the eighth paragraph, replace “a type parameter or an array bound” with “a type parameter, array bound, or cobound” and replace “the type parameter, or array bound” with “the type parameter, array bound, or cobound”.

[152:7-8]f08/0104 (N2042&N2102)

Subclause 7.1.12

In the first paragraph, replace item (8) by:

- (8) a reference to a transformational function from the intrinsic module IEEE_ARITHMETIC or IEEE_EXCEPTIONS (14), where each argument is a constant expression,

[153:25-28]f08/0140 (N2092&N2102)

Subclause 7.2.1.2

In the second paragraph, after “coindexed object,” delete “the variable”. Before each “shall not” insert “the variable” and before “shall have” insert “of the variable” so that the second paragraph reads:

If *variable* is a coindexed object,

- the variable shall not be polymorphic,
- the variable shall not have an allocatable ultimate component, and
- each deferred length type parameter of the variable shall have the same value as the corresponding type parameter of *expr*.

[157:14]f08/0147 (N2086)

Subclause 7.2.1.4

In item (5) (b) of the second paragraph, change “ x_1 and x_2 are conformable” to “ x_2 is scalar or has the same rank as x_1 ”.

[157:16]f08/0147 (N2086)

In the third paragraph of the subclause, append a new sentence:

If the subroutine is elemental, x_2 shall have the same shape as x_1 .

[170:19]f08/0118 (N2062&N2080)

Subclause 8.1.3.1

In constraint C801, change “*associate-name* shall not appear” to “neither the *associate-name* nor any subobject thereof shall appear”.

[171:12]f08/0118 (N2062&N2080)

Subclause 8.1.3.3

In the second paragraph, change “the associate name shall not appear” to “neither the associate name nor any subobject thereof shall appear”.

[172:13+]f08/0119 (N2062)

Subclause 8.1.4

Following the third paragraph, and before NOTE 8.5, insert a new paragraph:

It is permissible to branch to an *end-block-stmt* only from within its BLOCK construct.

[173:21+]f08/0119 (N2062)

Subclause 8.1.5

Following the third paragraph, and before NOTE 8.6, insert a new paragraph:

It is permissible to branch to an *end-critical-stmt* only from within its CRITICAL construct.

[178:16+]f08/0144 (N2092&N2102)

Subclause 8.1.6.7

In the first paragraph, after the fourth bullet point, add new bullet point:

- A DO CONCURRENT construct shall not contain an input/output statement that has an ADVANCE= specifier.

[184:13]f08/0118 (N2062&N2080)

Subclause 8.1.9.1

In constraint C836, change “*associate-name* shall not appear” to “neither the *associate-name* nor any subobject thereof shall appear”.

[190:5+]f08/0134 (N2062)

Subclause 8.5.3

Following constraint C851, add new constraint:

C851a (R859) A *stat-variable* or *errmsg-variable* in a *sync-stat* shall not be a coindexed object.

[190:16-]f08/0113 (N2042)

Subclause 8.5.4

Following constraint C852 and before the first paragraph, insert a new paragraph:

The value of *image-set* shall not depend on the value of *stat-variable* or *errmsg-variable*.

[194:6-]f08/0113 (N2042&N2047)

Subclause 8.5.6

Following constraint C853 and before the first paragraph, insert a new paragraph:

The *lock-variable* shall not depend on the value of *stat-variable*, *errmsg-variable*, or the *scalar-logical-variable* in the ACQUIRED_LOCK= specifier. The *scalar-logical-variable* shall not depend on the value of the *lock-variable*, *stat-variable*, or *errmsg-variable*.

[195:2-]f08/0113 (N2042)

Subclause 8.5.7

Before the first paragraph, insert a new paragraph:

The *stat-variable* shall not depend on the value of the *errmsg-variable*, *lock-variable*, or the *scalar-logical-variable* in the ACQUIRED_LOCK= specifier. The *errmsg-variable* shall not depend on the value of the *stat-variable*, *lock-variable*, or the *scalar-logical-variable* in the ACQUIRED_LOCK= specifier.

[275:18]f08/0142 (N2092)

Subclause 11.2.3

In constraint C1113, after “shall be the name of a nonintrinsic module” insert “that declares a separate module procedure”.

[281:25-28]f08/0132 (N2062&N2102)

Subclause 12.4.3.2

At the end of the first sentence of the fifth paragraph, change “or a dummy procedure” to “, dummy procedure, or procedure pointer”.

In the second sentence of the same paragraph, after “interface body, the procedure is a dummy procedure” change “; otherwise” to “. If the procedure has the POINTER attribute, it is a procedure pointer. If it is not a dummy procedure or procedure pointer”.

This makes that whole paragraph read:

An interface body in a generic or specific interface block specifies the EXTERNAL attribute and an explicit specific interface for an external procedure, dummy procedure, or procedure pointer. If the name of the declared procedure is that of a dummy argument in the subprogram containing the interface body, the procedure is a dummy procedure. If the procedure has the POINTER attribute, it is a procedure pointer. If it is not a dummy procedure or procedure pointer, it is an external procedure.

[282:7]f08/0100 (N2042)

Subclause 12.4.3.3

In the first paragraph, after “imported in this manner and is” change “defined” to “declared”.

[282:14]f08/0100 (N2042)

In the second paragraph, after “is accessed by host association and is” change “defined” to “declared”.

[295:3]f08/0135 (N2062&N2080)

Subclause 12.5.2.4

In the eighteenth paragraph, after applying the changes in Technical Corrigendum 1, between “is nonelemental” and “and the actual argument”, insert “, the dummy argument does not have the VALUE attribute,”. This makes the whole sentence read:

If the procedure is nonelemental, the dummy argument does not have the VALUE attribute, and the actual argument is an array section having a vector subscript, the dummy argument is not definable and shall not have the ASYNCHRONOUS, INTENT (OUT), INTENT (INOUT), or VOLATILE attributes.

[295:4+]f08/0122 (N2062&N2080)

After the eighteenth paragraph, add the following new paragraph before NOTE 12.24:

If the dummy argument has a coarray ultimate component, the corresponding actual argument shall have the VOLATILE attribute if and only if the dummy argument has the VOLATILE attribute.

[295:6]f08/0136 (N2062)

In constraint C1238, append at the end of the sentence: “, unless the dummy argument has the VALUE attribute”.

[295:9]f08/0136 (N2062)

In constraint C1239, after “ASYNCHRONOUS attribute” insert: “, but does not have the VALUE attribute”.

[295:13]f08/0136 (N2062)

In constraint C1240, after “ASYNCHRONOUS attribute” insert: “, but does not have the VALUE attribute”.

[300:14]f08/0117 (N2062&N2102)

Subclause 12.5.2.13

In item (3)(b) of the first paragraph, after “target other than” insert “a coindexed object or”.

[300:22]f08/0117 (N2062)

In item (4)(b) of the first paragraph, after “target other than” insert “a coindexed object or”.

[311:34+]f08/0141 (N2092)

Subclause 12.6.4

In the first paragraph, following constraint C1275 add new constraint:

C1275a A statement function shall not be of a parameterized derived type.

Subclause 12.7

In constraint C1278a which was added by Technical Corrigendum 1, viz

C1278a An INTENT (OUT) dummy argument of a pure procedure shall not be polymorphic.

after the word “polymorphic” insert “or have a polymorphic allocatable ultimate component”.

[312:35]f08/0148 (N2086)

In the second paragraph, replace list item (3) in constraint C1283 by:

- (3) as the *expr* corresponding to a component in a *structure-constructor* if the component has the POINTER attribute or has a pointer component at any level of component selection,

[330:20]f08/0103 (N2042)

Subclause 13.7.16

In the fifth paragraph (Result Value) Case (ii), after “with TARGET” insert: “and, if TARGET is an internal procedure, they have the same host instance”.

[330:22]f08/0103 (N2042)

In the same paragraph Case (iii), after “same procedure” insert: “and, if the procedure is an internal procedure, they have the same host instance”.

[368:26]f08/0102 (N2042)

Subclause 13.7.110

In the fourth paragraph (Result Characteristics), replace “Same as TSOURCE.” by:

Same type and type parameters as TSOURCE. Because TSOURCE and FSOURCE are required to have the same type and type parameters (for both the declared and dynamic types), the result is polymorphic if and only if both TSOURCE and FSOURCE are polymorphic.

[372:18]f08/0106 (N2042)

Subclause 13.7.118

In the third paragraph (Arguments) in the description of FROM, after “It shall be allocatable” add “and shall not be a coindexed object”.

[372:19]f08/0106 (N2042)

In the same paragraph in the description of TO, after “It shall be allocatable” add “and shall not be a coindexed object”.

[389:4-5]f08/0123 (N2062)

Subclause 13.7.157

In the fifth paragraph (Result Value), change “ $\max(e-p, e_{\min}-1)$ ” to “ $e-p$ ”. After “that of X” replace “; if there are two such values” by “, provided this result is representable; otherwise the result is the same as that of TINY (X). If there are two extended model values equally near to X,”

This makes the whole paragraph read:

If X does not have the value zero and is not an IEEE infinity or NaN, the result has the value b^{e-p} , where b , e , and p are as defined in 13.4 for the value nearest to X in the model for real values whose kind type parameter is that of X, provided this result is representable; otherwise the result is the same as that of TINY (X). If there are two extended model values equally near to X, the value of greater absolute value is taken. If X has the value zero, the result is the same as that of TINY (X). If X is an IEEE infinity, the result is an IEEE NaN. If X is an IEEE NaN, the result is that NaN.

[393:18]f08/0137 (N2062)

Subclause 13.7.168

In the third paragraph (Arguments), to the definition of MOLD, append: “If the storage size of SOURCE is greater than zero and MOLD is an array, a scalar with the type and type parameters of MOLD shall not have a storage size equal to zero.”

[399:17]f08/0109 (N2086&N2102)

Subclause 13.8.2.16

In the second paragraph, in constraint C1302, replace “variable of type LOCK_TYPE” by “variable with declared type LOCK_TYPE”.

[408:1-]f08/0104 (N2042&N2047&N2094)

Subclause 14.10

In the third paragraph, Table 14.1, for procedure IEEE_SUPPORT_ROUNDING change the “Class” column entry from “I” to “T”.

In the same paragraph, Table 14.2, for procedures IEEE_SUPPORT_FLAG and IEEE_SUPPORT_HALTING change the “Class” column entries from “I” to “T”.

[418:16]f08/0104 (N2042)

Subclause 14.11.27

In the second paragraph (Class), change “Inquiry function” to “Transformational function”.

[418:32]f08/0104 (N2042)

Subclause 14.11.28

In the second paragraph (Class), change “Inquiry function” to “Transformational function”.

[420:4]f08/0104 (N2042)

Subclause 14.11.32

In the second paragraph (Class), change “Inquiry function” to “Transformational function”.

[426:19]f08/0104 (N2042)

Subclause 15.2.3.2

In the second paragraph (Class), change “Inquiry function” to “Transformational function”.

[428:9]f08/0104 (N2042)

Subclause 15.2.3.5

In the second paragraph (Class), change “Inquiry function” to “Transformational function”.

[428:21]f08/0104 (N2042)

Subclause 15.2.3.6

In the second paragraph (Class), change “Inquiry function” to “Transformational function”.

[436:15]f08/0116 (N2062)

Subclause 15.5.1

Append the following sentence to the first paragraph: “A C function that has an inline definition and no external definition is not considered to be defined in this sense.”

[436:16-19]f08/0116 (N2062)

Replace the second paragraph by:

If the procedure is defined by means other than Fortran,

- it shall be describable by a C prototype that is interoperable with the interface, and
- if it is accessed using its binding label, it shall
 - have a name that has external linkage as defined by 6.2.2 of ISO/IEC 9899:1999, and
 - have the same binding label as the interface.

[440:4]f08/0120 (N2062&N2102)

Subclause 16.3.1

In item (1) of the first paragraph, after “named constants,” insert “named procedure pointers,”.