ISO/IEC/JTC1/SC22/WG5-N1823

COMBINED TECHNICAL CORRIGENDA 1 TO 5 FOR FORTRAN 2003

This document is derived from the WG5 files which formed the basis of the Technical Corrigenda for ISO/IEC 1539-1:2004 (Fortran 2003), that is N1636, N1664, N1727, N1775 and N1815.

The page and line numbers on the right which introduce each edit refer to J3/04–007. The interpretation upon which the edit is based and the Technical Corrigendum in which the edit was first published are shown in italics. Page and line numbers are not relevant for ISO/IEC 1539–1:2004 or for WG5–N1601. For these documents subclause and paragraph should be used to locate the edit.

[36:14] *f03/0020 TC2*

Subclause 4.4.1

In the third line of the first paragraph of the subclause, before the sentence beginning “The kind type parameter...” insert the following sentence.

> The kind type parameter is of type default integer.

[37:30] *f03/0020 TC2*

Subclause 4.4.2

In the fourth line of the first paragraph of the subclause, before the sentence beginning “The kind type parameter...” insert the following sentence.

> The kind type parameter is of type default integer.

[38:2] *f03/0029 TC2*

In the second line of the second paragraph of the subclause, insert “mathematically” before “equivalent”.

[39:15] *f03/0020 TC2*

Subclause 4.4.3

In the fourth line of the second paragraph of the subclause, before the sentence beginning “The kind type parameter...” insert the following sentence.

> The kind type parameter is of type default integer.

[40:10] *f03/0020 TC2*

Subclause 4.4.4

In the third line of the first paragraph of the subclause, insert “its kind is processor-dependent and” before “its value”.

[40:14] *f03/0020 TC2*

In the third line of the second paragraph of the subclause, before the sentence beginning “The kind type parameter...” insert the following sentence.

> The kind type parameter is of type default integer.

[41:9] *f03/0027 TC2*

Subclause 4.4.4.1

In constraint C416, in the second line of list item (3), delete “or”.

[41:9+] *f03/0027 TC2*

In constraint C416, add the following new list item immediately after item (3).

> (3.5) in the *type-spec* or *derived-type-spec* of a type guard statement (8.1.5), or

[41:11-12] *f95/0098 TC5*

In constraint C417, change "unless ... dummy function" to "unless it is of type CHARACTER and is the name of a dummy function or the name of the result of an external function".

[41:33+] *f03/0027 TC2*

In the final paragraph of the subclause, add the following new list item immediately after item (3).

> (3.5) If used in the *type-spec* of a type guard statement, the associating entity assumes its length from the selector.

[41:34,36] *f95/0098 TC5*

At the end of subclause 4.4.4.1 in list item (4), after "invoking the function" insert "or passing it as an actual argument"; change "host or use" to "argument, host, or use".

[44:2] *f03/0020 TC2*

Subclause 4.4.5

In the second line of the second paragraph of the subclause, before the sentence beginning "The kind type parameter..." insert the following sentence.

> The kind type parameter is of type default integer.

[47:11] *f03/0122 TC5*

Subclause 4.5.1.3

In the last sentence of the second paragraph, delete "declared to be PRIVATE or".

[50:40+] *f03/0072 TC2*

Subclause 4.5.3

Following constraint C447, add the following new constraint.

> C447a (R440) If *component-initialization* appears, every type parameter and array bound of the component shall be an initialization expression.

[53:1] *f03/0009 TC1*

Subclause 4.5.3.3

Add the following sentence at the end of constraint C453.

> It shall not have the VALUE attribute.

[58:14] *f03/0094 TC3*

Subclause 4.5.5

In the last sentence of C473 replace "not be INTENT(OUT)" by "not have the INTENT(OUT) or VALUE attribute".

[59:27] *f03/0062 TC2*

Subclause 4.5.5.2

In the first line of the fourth paragraph of the subclause, after the first occurrence of "structure constructor", insert "or array constructor". On the same line, delete the second occurrence of "structure".

[59:30] *f03/0007 TC1*

In the fifth paragraph, replace "first executable statement" by "executable constructs".

[59:30+] *f03/0007 TC1*

Add the following paragraph after the fifth paragraph.

> If a specification expression in a scoping unit references a structure constructor, the entity created by the structure constructor is finalized before execution of the executable constructs in the scoping unit.

[59:30+] *f03/0062 TC2*

In the new sixth paragraph introduced in Technical Corrigendum 1, after the first occurrence of "structure constructor", insert "or array constructor". In the same sentence, delete the second occurrence of "structure".

[67:19+] *f03/0090 TC5*

Subclause 4.7

Following constraint C493, add a new constraint:

> C493a (R469) An *ac-value* shall not be unlimited polymorphic.

[67:21] *f03/0090 TC5*

In constraint C494, replace "same type" by "same declared type".

[68:9] *f03/0090 TC5*

In the second paragraph of the subclause, after "; in this case, the", replace "type and type parameters" by "declared type and type parameters".

[68:11] *f03/0090 TC5*

In the first sentence of the third paragraph of the subclause, replace "type and type parameters" by "declared type and type parameters".

[68:14+] *f03/0090 TC5*

Following the third paragraph of the subclause, insert the following new paragraph:

> The dynamic type of an array constructor is the same as its declared type.

[72:23] *f03/0013 TC1*

Subclause 5.1

Add the following sentence at the end of constraint C509.

> It shall not have the VALUE attribute.

[72:28] *f03/0012 TC2*

In the first line of constraint C512, delete ", EXTERNAL".

[73:7] *f03/0012 TC2*

In the second line of constraint C521, after "dummy procedure" insert ", a procedure pointer".

[73:35-36] *f03/0012 TC2*

Replace constraint C536 by the following.

> C536 (R501) If the PROTECTED attribute is specified, the INTRINSIC or PARAMETER attribute shall not be specified. If the PROTECTED and EXTERNAL attributes are specified, the POINTER attribute shall also be specified.

[78:21-22] *f03/0014 TC1*

Subclause 5.1.2.5.1

In constraint C542, replace "a dummy argument, a function result, or an automatic array of a procedure" by "declared only in a subprogram or interface body".

[78:23] *f03/0014 TC1*

In the paragraph following constraint C542, after "subprogram" add "or interface body".

[80:9] *f03/0045 TC2*

Subclause 5.1.2.5.4

In the first line of constraint C544, insert "polymorphic, of a finalizable type, of a type with an ultimate allocatable component, or" before "of a type".

[81:15] *f03/0127 TC5*

Subclause 5.1.2.7

In the first sentence of the second paragraph, change “during the execution’ to “during the invocation and execution”.

[81:26] *f03/0127 TC5*

Append the following sentence to the third paragraph: “Any undefinition or definition implied by association of an actual argument with an INTENT(OUT) dummy argument shall not affect any other entity within the statement that invokes the procedure.”

[85:10] *f03/00099 TC4*

Subclause 5.1.2.16

In the third paragraph of the subclause replace “association status and array bounds” by “association status, dynamic type and type parameters, and array bounds”.

[86:1] *f03/0012 TC2*

Subclause 5.2

Replace the last sentence of the first paragraph of subclause 5.2 (“This also applies ... statements.”) by the following.

> This also applies to procedure declaration statements, and to EXTERNAL and INTRINSIC statements.

[91:5] *f03/0012 TC2*

Subclause 5.2.10

Replace constraint C568 by the following.

> C568 (R541) The EXTERNAL attribute (5.1.2.6) shall be explicitly specified for a *proc-entity-name*.

[93:9] *f03/0134 TC5*

Subclause 5.3

In the first sentence of the fourth paragraph, after “intrinsic function,” insert “is not a component,”.

[97:7+] *f03/0131 TC5*

Subclause 5.5.1.1

Insert the following new paragraph at the end of the subclause:

> If any data object in an *equivalence-set* has the SAVE attribute, all other objects in the *equivalence-set* have the SAVE attribute; this may be confirmed by explicit specification.

[98:17] *f03/0063 TC5*

Subclause 5.5.2

In rule R558, delete the second line: “**or** *proc-pointer-name”*

[98:18] *f03/0063 TC5*

In constraint C587, delete “or *proc-pointer-name*”

[98:21] *f03/0063 TC5*
[98:22] *f03/0133 TC5*

In constraint C588, after “that is allocatable,” insert “a procedure pointer,” and after “BIND attribute,” insert “an unlimited polymorphic pointer,”.

[98:25] *f03/0063 TC5*

In constraint C590, delete “or *proc-pointer-name*”.

[100:12-15] *f03/0063 TC5*

Subclause 5.5.2.3
In the sixth paragraph of the subclause, delete the third, fourth and fifth sentences: "A procedure pointer shall be storage ... type parameters.".

[103:17-18] *f03/0114 TC4*

Clause 6
Delete rule R605, which defines *default-logical-variable,* and constraint C604.

[107:2-3] *f03/0077 TC4*

Subclause 6.2.1
Replace the first paragraph of the subclause ("A **whole array** ... to the name.") by the following:

> A **whole array** is a named array or a structure component whose final *part-ref* is an array component name; no subscript list is appended.

[107:7,8] *f03/0077 TC4*

In the third paragraph, replace both occurrences of "whole array name" by "whole array designator".

[111:11] *f03/0011 TC1*

Subclause 6.3.1
In constraint C625, after "unlimited polymorphic" add "or is of abstract type".

[113:21] *f03/0098 TC4*

Subclause 6.3.1.1
At the end of the last sentence in the last paragraph of the subclause insert "unless the SOURCE= specifier appears and the corresponding component of the *source-expr* is allocated".

[116:8] *f03/0007 TC1*

Subclause 6.3.3.1
In the second paragraph following Note 6.24, replace "first executable statement" by "executable constructs".

[116:25] *f03/0024 TC5*

Subclause 6.3.3.2
After the first sentence of the second paragraph, insert the new sentence: "The pointer shall have the same dynamic type and type parameters as the allocated object, and if the allocated object is an array the pointer shall be an array whose elements are the same as those of the allocated object in array element order."

[126:19+] *f95/0030 TC1*

Subclause 7.1.6
Add the following paragraph immediately before Note 7.10.

> If a specification expression in a module includes a reference to a generic, that generic shall have no specific procedures defined in the module subsequent to the specification expression.

[127:33+] *f95/0030 TC1*

Subclause 7.1.7
Add the following paragraph immediately before Note 7.11.

> If an initialization expression in a module includes a reference to a generic, that generic shall have no specific procedures defined in the module subsequent to the initialization expression.

[139:17] *f03/0006 TC1*

Subclause 7.4.1.3

In the first paragraph, replace "the evaluation of all operations in *expr* and *variable*" with "the evaluation of *expr* and the evaluation of all expressions in *variable*".

[139:22-] *f03/0093 TC3*

Insert the following sentence at the beginning of the third paragraph of the subclause:

> If *variable* is an unallocated allocatable array, *expr* shall have the same rank as *variable*.

[139:25] *f03/0093 TC3*

In the second sentence of the third paragraph of the subclause, change "corresponding type parameters of *expr*," to "corresponding type parameter of *expr*. If *variable* is an array and *expr* is scalar it is allocated with the same bounds as before, otherwise it is allocated".

[141:20,21,22] *f03/0006 TC1*

In list item (2) of the paragraph immediately following Note 7.39, insert "the value of" before each occurrence of "*expr*".

[143:12,24,35,37] *f03/0008 TC2*

Subclause 7.4.2

In the second line of rule R736, replace "*variable %*" by "*scalar-variable %*".

In the first line of constraint C722, replace "*variable*" by "*scalar-variable*".

In rule R741, replace "*variable*" by "*scalar-variable*".

In the second line of constraint C725, replace "*variable*" by "*scalar-variable*".

[144:5-6] *f03/0138 TC5*

In constraint C727, change "an external, module," to "a module" and change "or a procedure pointer" to "a procedure pointer, or an external procedure that is accessed by use or host association and is referenced in the scoping unit as a procedure, or that has the EXTERNAL attribute".

[145:5] *f03/0092 TC3*

Subclause 7.4.2.2

In the fifth paragraph of the subclause, after "the same type", insert "or both be unlimited polymorphic".

[161:18-19] *f03/0015 TC1*

Subclause 8.1.4.3

In the first paragraph delete ", TARGET,".

[161:19] *f03/0015 TC1*

In the same paragraph, after "the attribute.", insert the following sentence.

> The associating entity has the TARGET attribute if and only if the selector is a variable and has either the TARGET or POINTER attribute.

[162:17,19,20,21] *f03/0025 and f03/0026 TC2*

Subclause 8.1.5.1
In rule R823, replace the line

or CLASS IS (*type-spec*) [*select-construct-name*]

by the line

or CLASS IS (*derived-type-spec*) [*select-construct-name*]

In constraint C814, after "*type-spec*", insert "or *derived-type-spec*".

In constraint C815, after "*type-spec*", insert "or *derived-type-spec*".

In the first line of constraint C816, after "*type-spec*", insert "or *derived-type-spec*".

[187:2] *f03/0113 TC4*

Subclause 9.5.1
In rule R913, replace "ID = *scalar-int-variable*" by "ID = *id-variable*".

[187:10+] *f03/0113 TC4*

Following rule R913, insert new rule and new constraint:

R913a *id-variable* **is** *scalar-int-variable*

C908a (R913a) The *scalar-int-variable* shall have a decimal range no smaller than that of default integer.

[189:7,9] *f03/0070 TC2*

Subclause 9.5.1.3
In the second line of the subclause, replace "this input/output statement" by "a nonchild input/output statement".

In the fourth line of the subclause, replace "from an input/output statement" by "from a nonchild input/output statement".

At the end of the paragraph insert the following sentence.

> A formatted child input/output statement is a nonadvancing input/output statement, and any ADVANCE= specifier is ignored.

[190:17] *f03/0113 TC4*

Subclause 9.5.1.8
In the second sentence of the first paragraph of the subclause, replace "This value" by "If this value is zero, the data transfer operation has been completed. A nonzero value".

[190:18] *f03/0113 TC4*

After the second sentence of the first paragraph, insert the following sentence: "This identifier is different from the identifier of any other pending data transfer operation for this unit."

[193:13-15] *f03/0132 TC5*

Subclause 9.5.2
Replace the twelfth paragraph of the subclause ("If a derived-type list item … input/output procedure.") by the following new paragraph:

> If a derived-type list item is not processed by a user-defined derived-type input/output procedure and is not treated as a list of its individual components, all of the subcomponents of that list item shall be accessible in the scoping unit containing the input/output statement and shall not be pointers or allocatable.

[195:23+] *f03/0117 TC4*

Subclause 9.5.3

After the last paragraph of the subclause, insert the following new paragraph:

> If execution of the program is terminated during execution of a WRITE or PRINT statement, the contents of the file become undefined.

[196:29+] *f03/0050 TC3*

Subclause 9.5.3.4

After the seventh paragraph of the subclause, "If an internal file has been specified, an input/output list item shall not be in the file or associated with the file.", add the following two paragraphs:

> During the execution of an output statement that specifies an internal file, no part of that internal file shall be referenced, defined, or become undefined as the result of evaluating any output list item.
>
> During the execution of an input statement that specifies an internal file, no part of that internal file shall be defined or become undefined as the result of transferring a value to any input list item.

[198:12] *f95/0096 TC1*

Subclause 9.5.3.4.2

In the eighth paragraph of the subclause, replace "input item and its corresponding data edit descriptor" by "effective input item and its corresponding data edit descriptors".

[199:8+] *f03/0070 TC2*

Subclause 9.5.3.7.1

In the final paragraph of the subclause, add the following after the first item in the bulleted list.

- Any ADVANCE= specifier in a child input/output statement is ignored.

[206:18] *f03/0113 TC4*

Subclause 9.6.1

In the first sentence of the third paragraph of the subclause, replace "the identifier" by "zero or the identifier".

[206:20] *f03/0113 TC4*

In the second sentence of the same paragraph, after "transfer operation" insert ", if any,".

[210:18, 25, 28, 30] *f03/0114 TC4*

Subclause 9.9.1

In rule R930, replace:
"or EXIST = *scalar-default-logical variable*" by "or EXIST = *scalar-logical variable*",
"or NAMED = *scalar-default-logical variable*" by "or NAMED = *scalar-logical variable*",
"or OPENED = *scalar-default-logical variable*" by "or OPENED = *scalar-logical variable*", and
"or PENDING = *scalar-default-logical variable*" by "or PENDING = *scalar-logical variable*".

[212:15] *f03/0106 TC3*

Subclause 9.9.1.8

Add to the end of the last sentence of the subclause: "or if the unit specified by UNIT= is not connected to a file".

[212:21] *f03/0106 TC3*

Subclause 9.9.1.9

In the last sentence of the subclause, after "file" insert: "or if the unit specified by UNIT= is not connected to a file".

[212:24] *f03/0114 TC4*

Subclause 9.9.1.10
In the first sentence of the subclause, replace "*scalar-default-logical variable*" by "*scalar-logical-variable*".

[212:36] *f03/0106 TC3*

Subclause 9.9.1.12
Add to the end of the last sentence of the subclause: "or if the unit specified by UNIT= is not connected to a file".

[213:9] *f03/0114 TC4*

Subclause 9.9.1.15
Replace "*scalar-default-logical variable*" by "*scalar-logical-variable*".

[213:15-16] *f03/0106 TC3*

Subclause 9.9.1.16
In the third sentence of the subclause, change "or if" to ", if" and after "condition" insert ", or if the unit specified by UNIT= is not connected to a file".

[213:20-21] *f03/0106 TC3*

Subclause 9.9.1.17
Replace the subclause by the following:

> Execution of an INQUIRE by file statement causes the *scalar-int-variable* in the NUMBER= specifier to be assigned the value of the external unit number of the unit that is connected to the file. If there is no unit connected to the file, the value -1 is assigned. Execution of an INQUIRE by unit statement causes the *scalar-int-variable* to be assigned the value specified by UNIT=.

[213:23, 25] *f03/0114 TC4*

Subclause 9.9.1.18
In both the first and second sentences of the subclause, replace "*scalar-default-logical variable*" by "*scalar-logical-variable*".

[214:19-20] *f03/0106 TC3*

Subclause 9.9.1.21
In the third sentence of the subclause, change "or if" to ", if" and after "conditions" insert ", or if the unit specified by UNIT= is not connected to a file".

[215:2] *f03/0106 TC3*

Subclause 9.9.1.23
Add to the end of the last sentence of the subclause: "or if the unit specified by UNIT= is not connected to a file".

[215:7] *f03/0106 TC3*

Subclause 9.9.1.24
Add to the end of the last sentence of the subclause: "or if the unit specified by UNIT= is not connected to a file".

[215:26] *f03/0106 TC3*

Subclause 9.9.1.27
Add to the end of the last sentence of the subclause: "or if the unit specified by UNIT= is not connected to a file".

[215:34] *f03/0106 TC3*

Subclause 9.9.1.29
In the second sentence of the subclause, after "determined", insert: "or if the unit specified by UNIT= is not connected to a file".

[216:5] *f03/0106 TC3*

Subclause 9.9.1.30
Add to the end of the last sentence of the subclause: "or if the unit specified by UNIT= is not connected to a file".

[216:10] *f03/0106 TC3*

Subclause 9.9.1.31
Add to the end of the last sentence of the subclause: "or if the unit specified by UNIT= is not connected to a file".

[216:15] *f03/0106 TC3*

Subclause 9.9.1.32
Add to the end of the last sentence of the subclause: "or if the unit specified by UNIT= is not connected to a file".

[218:6-7] *f95/0096 TC1*

Subclause 9.10.3
In list item (1) replace "input list item (9.5.3.4.2) and corresponding data edit descriptor that requires" by "effective input item (9.5.2) and its corresponding data edit descriptors that require".

[219:30] *f03/0117 TC4*

Subclause 9.11
Delete the last paragraph of the subclause, viz. "A STOP statement shall not be executed during execution of an input/output statement".

[227:15+] *f03/0079 TC3*

Subclause 10.6.1
Following the numbered list in subclause 10.6.1, add a seventh item:

(7) On output of a real zero value, the digits in the exponent field shall all be zero.

[240:13] *f03/0028 TC2*

Subclause 10.9.1
In the eighth line of the penultimate paragraph of the subclause, replace "blank, comma, slash" by "blank, comma (if the decimal edit mode is POINT), semicolon (if the decimal edit mode is COMMA), slash".

[241:5] *f03/0049 TC4*

Subclause 10.9.2
Add the following sentence at the end of the first paragraph of the subclause and before the new text that was added in Technical Corrigendum 3 (below):

> Two undelimited character sequences are considered adjacent when both were written using list-directed input/output, no intervening data transfer or input/output file positioning operations on that unit occurred, and both were written either by a single data transfer statement, or during the execution of a parent data transfer statement along with its child data transfer statements.

[241:5] *f03/0101 TC3*

At the end of the first paragraph of the subclause, add the following sentence:

> The form of the values produced by a user-defined derived-type output routine invoked during list-directed output is specified by the invoked routine. This form need not be compatible with list-directed input.

[243:5] *f03/0097 TC3*

Subclause 10.10

Replace the final paragraph of the subclause by the following:

> A value separator for namelist formatting is a value separator for list-directed formatting (10.9), or one or more contiguous blanks between a nonblank value and the following object designator or "!" comment initiator.

[244:8,10] *f03/0057 TC2*

Subclause 10.10.1.2

In the second line of the third paragraph of the subclause, delete "of intrinsic data types".

In the fourth line of the same paragraph, replace "intrinsic type" by "type".

[244:29,30,32,33] *f03/0028 TC2*

Subclause 10.10.1.3

In the second line of the second paragraph of the subclause, replace "a comma" by "a comma (if the decimal edit mode is POINT) or a semicolon (if the decimal edit mode is COMMA),".

In the third line of the same paragraph, before the sentence beginning "The first numeric input field..." insert the following sentence.

> The separator is a comma if the decimal edit mode is POINT; it is a semicolon if the decimal edit mode is COMMA.

In the fifth and sixth lines of the same paragraph, replace "comma or between the comma" by "separator or between the separator".

[245:4] *f03/0028 TC2*

In the last line of the fifth paragraph of the subclause, after "comma," insert " semicolon,".

[246:4] *f03/0101 TC3*

Subclause 10.10.2

After "logical values" at the end of the first sentence of the first paragraph of the subclause, add ", and output produced by user-defined derived-type output".

[246:7] *f03/0101 TC3*

At the end of the same paragraph add the following two sentences:

> The form of the output produced by a user-defined derived-type output routine invoked during namelist output is specified by the invoked routine. This form need not be compatible with namelist input.

[247:33] *f03/0068 TC2*

Subclause 10.10.2.2

In the first line of the final paragraph of the subclause, before "continuation" insert "new records created by explicit formatting within a user-defined derived-type output procedure or by".

[260:2] *f03/0141 TC5*

Subclause 12.3.2.1

In the last sentence of the paragraph beginning "If an explicit specific interface ...", i.e. the sixth paragraph of the subclause, after "scoping unit" insert ", except that if the interface is accessed by use association, there may be more than one local name for the procedure". Append a sentence at the end of the paragraph: "If a procedure is accessed by use association, each access shall be to the same procedure declaration or definition.".

[261:3] *f03/0071 TC5*

At the end of the paragraph beginning "A generic interface block specifies ...". i.e. the ninth paragraph of the subclause, add the following sentence: "If a specific procedure in a generic interface has a function dummy argument, that argument shall have its type and type parameters explicitly declared in the specific interface.".

[262:16] *f03/0088 TC3*

Subclause 12.3.2.1.1

At the end of the second paragraph of the subclause, add the sentence:

> All restrictions and constraints that apply to actual arguments in a reference to the function also apply to the corresponding operands in the expression as if they were used as actual arguments.

[263:6] *f03/0069 TC2*

Subclause 12.3.2.1.2

In the first line of the second paragraph of the subclause, replace "Each argument shall be nonoptional" by "The dummy arguments shall be nonoptional dummy data objects".

[263:12] *f03/0088 TC3*

In the second paragraph of the subclause, after the penultimate sentence ("A defined assignment ... the second argument."), add the sentence:

> All restrictions and constraints that apply to actual arguments in a reference to the subroutine also apply to the left-hand side and to the right-hand side enclosed in parentheses as if they were used as actual arguments.

[263:13+16+] *f03/0112 TC5*

At the end of the subclause, after Note 12.10, add the following new note:

> NOTE 12.10a
> If the second argument of a procedure specified in a defined assignment interface block has the POINTER or ALLOCATABLE attribute, it cannot be accessed by defined assignment, since the right-hand side of the assignment is enclosed in parentheses before being associated as an actual argument with the second argument. This makes it an expression, which does not have the POINTER or ALLOCATABLE attribute.

[266:8] *f03/0044 TC1*

Subclause 12.3.2.5

Change "referenced" to "invoked".

[266:24] *f03/0003 and f03/0004 TC4*

Subclause 12.4

In constraint C1224, add as a second sentence: "The *data-ref* shall not be an unallocated allocatable variable or a pointer whose association status is disassociated or undefined.".

[266:24+] *f03/0016 TC1*

Insert a new constraint following C1224.

C1224a (R1219) If *data-ref* is an array, the referenced type-bound procedure shall have the PASS attribute.

[268:17] *f03/0043 TC1*

Subclause 12.4.1.1

After "procedure", insert ", or a procedure pointer component,".

[268:23] *f03/0010 TC1*

Subclause 12.4.1.2

In the first paragraph, before "the declared type of the actual argument", insert "either both the actual and dummy argument shall be unlimited polymorphic, or".

[269:3,5,8] *f03/0074 TC4*

After "of type default character" in the second paragraph of the subclause, add "or of type character with the C character kind (15.1.1)".

After "of type default character" in the first sentence of the third paragraph, add "or of type character with the C character kind".

After "of type default character" in the third sentence of the third paragraph, add "or of type character with the C character kind".

[270:1] *f03/0005 TC1*

In the paragraph following Note 12.22 replace "associated with an actual argument that is" by "used as an actual argument that is associated with".

[270:27] *f03/0061 TC2*

In the third line of the sixteenth paragraph of the subclause (which begins "If the actual argument is scalar,...") replace "assumed-shape or pointer" by "assumed-shape, pointer, or polymorphic".

[271:28] *f03/0137 TC5*

Subclause 12.4.1.3

Append the following two sentences at the end of the fifth paragraph of the subclause, that is after "function procedure pointer, or dummy procedure." : "If both the actual argument and dummy argument are known to be functions, they shall have the same type and type parameters. If only the dummy argument is known to be a function, the function that would be invoked by a reference to the dummy argument shall have the same type and type parameters, except that an external function with assumed character length may be associated with a dummy argument with explicit character length.".

[273:12+] *f03/0109 TC4*

Subclause 12.4.1.6

Following list item 10 of the subclause, insert:

(11) It shall not be supplied as the *data-ref* in a *procedure-designator*.

(12) It shall not be supplied as the *variable* in a *proc-component-ref.*

[275:2,5] *f03/0127 TC5*

Subclause 12.4.1.7

In list item (2) of the first paragraph, in each of the first and second sentences change "during the execution" to "during the invocation and execution".

[276:36+] *f03/0135 TC5*

Subclause 12.4.4
Add the following new list item after item (2) (a):

(a2) if that scoping unit is of a subprogram that defines a procedure with that name;

[278:5+] *f95/0078 TC1*

Subclause 12.4.4.1
Add the following list item at the end of the subclause.

(5) If (1), (2), (3), and (4) do not apply, the name is that of an intrinsic procedure, and the reference is consistent with the interface of that intrinsic procedure, then the reference is to that intrinsic procedure.

[278:15+] *f03/0135 TC5*

Subclause 12.4.4.2
Add the following new list item after item (3):

(3a) If the scoping unit is of a subprogram that defines a procedure with that name, the reference is to that procedure.

[280:6-7] *f03/0086 TC3*

Subclause 12.5.2.1
Replace subclause C1242 by the following:

C1242 An elemental procedure shall not have the BIND attribute.

[286:22+] *f03/0126 TC5*

Subclause 12.6
Following constraint C1271, add the following new constraint:

C1271a The *designator* of a variable with the VOLATILE attribute shall not appear in a pure subprogram.

[287:17, 288:1] *f03/0119 TC5*

Subclause 12.7.1
In constraint C1278, after “scalar” change “and” to a comma and at the end of the sentence insert, “, and shall not have a type parameter that is defined by an expression that is not an initialization expression”.

[293:5-6] *f90/0207 TC1*

Subclause 13.3
Delete the last sentence of the subclause, viz. “In particular ... processor dependent.”.

[316:5-6] *f03/0054 TC1*

Subclause 13.7.37
In the **Result Value** paragraph of the subclause, replace “model representation (13.4) for the value of X” by “representation for the value of X in the model (13.4) that has the radix of X but no limits on exponent values”.

[316:16-17] *f03/0125 TC5*

Subclause 13.7.38
In the **Arguments** paragraph, in each of the two argument descriptions, replace “of extensible type” by “of extensible declared type or unlimited polymorphic”.

[316:21-22] *f03/0125 TC5*

In the **Result Value** paragraph, after the second "otherwise" insert "if the dynamic type of A or MOLD is extensible," and at the end of the sentence insert "; otherwise the result is processor dependent".

[317:8] *f03/0054 TC1*

Subclause 13.7.40

In the **Result Value** paragraph of the subclause, replace "model representation of X" by "representation for the value of X in the model that has the radix of X but no limits on exponent values".

[326:8] *f03/0077 TC4*

Subclause 13.7.60

In *Case (i)* of the **Result Value** paragraph of the subclause, after "a whole array" delete "or array structure component".

[347:22] *f03/0055 TC1*

Subclause 13.7.100

In the **Result Value** paragraph of the subclause, replace "model representation of X" by "value nearest to X in the model for real values whose kind type parameter is that of X; if there are two such values, the value of greater absolute value is taken".

[347:30, 348:1] *f03/0125 TC5*

Subclause 13.7.101

In the **Arguments** paragraph, in each of the two argument descriptions, replace "of extensible type" by "of extensible declared type or unlimited polymorphic".

[348:3,4] *f03/0125 TC5*

In the **Result Value** paragraph, change "The result" to "If the dynamic type of A or B is extensible, the result" and append the following new sentence to the paragraph: "If neither A nor B has extensible dynamic type, the result is processor dependent."

[351:5] *f03/0054 TC1*

Subclause 13.7.107

In the **Result Value** paragraph of the subclause, replace "model representation of X" by "representation for the value of X in the model that has the radix of X but no limits on exponent values".

[353:9] *f03/0055 TC1*

Subclause 13.7.113

In the **Result Value** paragraph of the subclause, replace "model representation of X" by "value nearest to X in the model for real values whose kind type parameter is that of X; if there are two such values, the value of greater absolute value is taken".

[358:6-7] *f03/0077 TC4*

Subclause 13.7.124

In *Case (i)* of the **Result Value** paragraph of the subclause, after "a whole array" delete "or array structure component".

[363:9-10] *f03/0022 TC5*

Clause 14

In the first sentence of the second paragraph, replace "for all kinds of real and complex data" with "for all kinds of real and complex IEEE floating-point data".

[370:8-9] *f03/0107 TC3*

Subclause 14.9.2

In the first paragraph of the subclause replace “for reals X and Y for which IEEE_SUPPORT_DATATYPE(X) and IEEE_SUPPORT_DATATYPE(Y) are true” with “for all reals X and Y”.

[374:21] *f03/0023 TC2*

Subclause 14.10.7

In the first line of the **Argument** paragraph, after “shall be” add “scalar and”.

[376:17+] *f03/0034 TC5*

Subclause 14.10.12

At the end of the **Result Value** paragraph, add the following two new cases:

Case (iii): If IEEE_SUPPORT_INF(X) is true and X is infinite, the result is +infinity.

Case (iv): If IEEE_SUPPORT_NAN(X) is true and X is a NaN, the result is a NaN.

[380:13] *f03/0023 TC2*

Subclause 14.10.22

In the first line of the **Argument** paragraph, after “shall be” add “scalar and”.

[395:8] *f03/0129 TC5*

Subclause 15.1.2.5

In list item (1) of the **Argument** paragraph, before “type parameters” insert “kind”.

[395:16+] *f03/0129 TC5*

At the end of the **Argument** paragraph, insert the following new sentence after the list: “X shall not be a zero-length string.”.

[396:5-7] *f03/0129 TC5*

Subclause 15.2.1

In the second sentence of the first paragraph of the subclause replace “; if the type is character ... one.” by “. If the type is character, the length type parameter is interoperable if and only if its value is one.”.

[398:9] *f03/0089 TC3*

Subclause 15.2.3

In the first line of the second paragraph of the subclause, after “A Fortran derived type is interoperable with a C struct type if” insert “and only if”.

[398:11-12] *f03/0075 TC4*

In the first sentence of the second paragraph, replace “have types and type parameters that are interoperable with the types of the corresponding components of the struct type” with “would interoperate with corresponding components of the C struct type as described in 15.2.4 and 15.2.5 if the components were variables”.

[399:2-3] *f03/0129 TC5*

Subclause 15.2.4
Replace the first paragraph by the following paragraph:

> A named scalar Fortran variable is **interoperable** if and only if its type and type parameters are interoperable, it has neither the pointer nor the allocatable attribute, and if it is of type character its length is not assumed or declared by an expression that is not an initialization expression.

[399:7-8] *f03/0129 TC5*

Subclause 15.2.5
Replace the first paragraph by the following paragraph:

> A Fortran variable that is a named array is **interoperable** if and only if its type and type parameters are interoperable, it is of explicit shape or assumed size, and if it is of type character its length is not assumed or declared by an expression that is not an initialization expression.

[407:28] *f03/0136 TC5*

Subclause 16.2.3
In the second paragraph, after "**distinguishable** if" insert "one is a subroutine and the other is an array, or if".

[409:19] *f03/0140 TC5*

Subclause 16.3
In the second sentence of the second paragraph, before "scoping unit that includes" insert "innermost executable construct or".

[409:26] *f03/0140 TC5*

In the second sentence of the third paragraph, before "scoping unit that includes" insert "innermost executable construct or".

[411:21] *f03/0063 TC5*

Subclause 16.4.1.3
Delete list item (7): "A *proc-pointer-name* in a *common-block-object* in a *common-stmt*;"

[415:27+] *f03/00099 TC4*

Subclause 16.4.2.1.4
After the second paragraph of the subclause, add the following new paragraph:

> The association status of a pointer object with the VOLATILE attribute might change by means not specified by the program.

[416:23] *f03/0063 TC5*

Subclause 16.4.3.1
In list item (8) of the second paragraph, replace "A pointer" by "A data pointer".

[421:27] *f03/0098 TC4*

Subclause 16.5.5
In list item 19 of the subclause, after "default-initialized subcomponent", insert ", except by an ALLOCATE statement with a SOURCE= specifier,"

[421:28+] *f03/0098 TC4*

Following list item 19, insert:

(19a) Successful execution of an ALLOCATE statement with a SOURCE= specifier causes a subobject of the allocated object to become defined if the corresponding subobject of the SOURCE= expression is defined.

[421:43] *f03/00099 TC4*

In list item 26, replace “becomes” by “might become”.

[422:41-43] *f03/0098 TC4*

Subclause 16.5.6

Replace list item 11 of the subclause by the following:

(11) Successful allocation of an ALLOCATE statement with no SOURCE= specifier causes a subcomponent of an allocated object to become undefined if default initialization has not been specified for that subcomponent.

[422:43+] *f03/0098 TC4*

Following list item 11, insert:

(11a) Successful execution of an ALLOCATE statement with a SOURCE= specifier causes a subobject of the allocated object to become undefined if the corresponding subobject of the SOURCE= expression is undefined.

[423:28+] *f03/00099 TC4*

Following list item 18, insert:

(19) An object with the VOLATILE attribute (5.1.2.16) might become undefined by means not specified by the program.

[436:36] *f03/0077 TC4*

Annex A

In the **whole array** paragraph of the subclause, after “named array” insert “or an array component of a structure, with no subscript list”.