

Information technology — Programming languages — Fortran — Part 1: Base language

TECHNICAL CORRIGENDUM 1

Technologies de l'information — Langages de programmation — Fortran —

Partie 1: Langage de base

RECTIFICATIF TECHNIQUE 1

Technical Corrigendum 1 to International Standard ISO/IEC 1539-1:1997 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 22, *Programming languages, their environments and system software interfaces*.

Page 32

Subclause 4.3.1.1

Replace text of constraint after R407 with “A *boz-literal-constant* may appear only as a *data-stmt-constant* in a *DATA* statement.”

Pages 39 and 40

Subclause 4.4.1

In the text of the constraint preceding R428, replace: “a constant specification expression (7.1.6.2)” with “an initialization expression (7.1.6.1)”.

In the second constraint after R429, replace “a constant specification expression (7.1.6.2)” with “an initialization expression (7.1.6.1)”.

In NOTE 4.19, replace “a constant” with “an initialization”.

Pages 48 and 49

Subclause 5.1

In the final paragraph of Page 48, replace: “may be a nonconstant expression provided the specification expression” with “shall be an initialization expression unless it”.

In the first paragraph of page 49:

- (i) delete the sentence: “If a ... nonconstant expression.”.
- (ii) replace “such a nonconstant expression” with “a *specification-expr* that is not an initialization expression”.

In the first paragraph following NOTE 5.3, replace: “a nonconstant expression” with “an expression that is not an initialization expression”.

Page 51

Subclause 5.1.1.5

In the paragraph following NOTE 5.6, replace “a constant specification” with “an initialization”.

Page 53

Subclause 5.1.2.3

Replace item (2) of the second constraint with

- (2) A *pointer-object* in a *pointer-assignment-stmt* or *nullify-stmt*,

Replace item (3) of the second constraint with

- (3) A *do-variable* in a *do-stmt* or *io-implied-do*,

Page 54

Subclause 5.1.2.4.1

In the constraint, replace: “nonconstant expressions” with “expressions that are not initialization expressions”.

In line 2 after the constraint, replace “nonconstant specification” with “not initialization”.

In line 3 after the constraint, replace “nonconstant specification” with “not initialization”.

In lines 5 and 6 after the constraint, replace the sentence by “The bounds of such an array are unaffected by the redefinition or undefined of any variable during execution of the procedure.”.

Page 56

Subclause 5.1.2.4.4

In line 1 of the final paragraph of the subclause, replace “nonconstant specification” with “not initialization”.

In the final paragraph of the subclause, replace the final sentence “The bounds of such...procedure.” with “The bounds of such an array are unaffected by the redefinition or undefined of any variable during execution of the procedure.”.

Page 69

Subclause 5.5.2

In lines 3-4 of page 69, replace “a constant specification expression (7.1.6.2)” with “an initialization expression (7.1.6.1)”.

Pages 89 and 90

Subclause 7.1.2

In Table 7.1, penultimate line, change “requiring” to “with”.

To the second paragraph of Page 90 (line 9), append: “For the relational intrinsic operators with character operands, the kind type parameters shall be the same”.

In the third paragraph of page 90 (line 12), delete: “and have the same kind type parameter value”.

Page 111

Subclause 7.5.2

At the end of the paragraph that begins “The *target* shall”, append: “If the *target* is an allocatable array, it shall be currently allocated.”.

Page 176

Subclause 10.8.1

In the fourth paragraph of page 176, replace item (4) with

- (4) The leading characters are not *digits* followed by an asterisk, and

Page 188

Subclause 11.3.2

In the second paragraph following NOTE 11.8, delete: “and public accessibility”.

In the paragraph preceding NOTE 11.9, delete: “with public accessibility”

Page 192

Subclause 12.2.2

In line 3 of the subclause, change “a constant” to “an initialization”.

Page 266

Subclause 13.14.95

After “**Result Value.**”, insert “If P or R is absent, the result value is as would have been obtained with the argument present with the value 0.”

In lines 4-7 of the **Result Value** paragraph, replace “the result is –1 ... is supported.” with “the result is –1 if the processor does not support a real data type with a precision greater than or equal to P but does support a real data type with an exponent range greater than or equal to R, –2 if the processor does not support a real data type with an exponent range greater than or equal to R but does support a real data type with a precision greater than or equal to P, –3 if the processor supports no real data type with either of these properties, and –4 if the processor supports real data types for each separately but not together.”