



INTERNATIONAL STANDARD ISO/IEC 1539-1:1997 TECHNICAL CORRIGENDUM 2

Published 2002-06-15

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION • МЕЖДУНАРОДНАЯ ОРГАНИЗАЦИЯ ПО СТАНДАРТИЗАЦИИ • ORGANISATION INTERNATIONALE DE NORMALISATION
INTERNATIONAL ELECTROTECHNICAL COMMISSION • МЕЖДУНАРОДНАЯ ЭЛЕКТРОТЕХНИЧЕСКАЯ КОМИССИЯ • COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

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

TECHNICAL CORRIGENDUM 2

*Technologies de l'information — Langages de programmation — Fortran —
Partie 1: Langage de base*

RECTIFICATIF TECHNIQUE 2

Technical Corrigendum 2 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 3

Subclause 1.5.1

Add these extra items at the end on 1.5.1:

- (3) Earlier standards specified that if the second argument to MOD or MODULO was zero, the result was processor dependent. This standard specifies that the second argument shall not be zero.
- (4) The PAD= specifier in the INQUIRE statement in this standard returns the value UNDEFINED if there is no connection or the connection is for unformatted input/output. The previous standard specified YES.

Page 39

Subclause 4.4.1

Replace the second line of R429 by

R429a *pointer-initialization-expr* **or** *=> pointer-initialization-expr*
is *function-reference*

Constraint: *function-reference* shall be a reference to the intrinsic function NULL with no arguments.

Page 45

Subclause 4.4.4

Replace the paragraph that starts “Where a component” by

Where a component in the derived type is a pointer, the corresponding constructor expression shall be an allowable *target* for such a pointer in a pointer assignment statement (7.5.2).

Pages 45 and 46

Subclause 4.5

Add to end of the paragraph that follows the constraints: “The character length of an *ac-value* in an *ac-implied-do* whose iteration count is zero shall not depend on the value of the implied DO variable and shall not depend on the value of an expression that is not an initialization expression.”.

In the first line of page 46, replace “an *ac-value* sequence” with “a sequence of elements”.

Page 47

Subclause 5.1

Replace the second line of R505 by

or => *pointer-initialization-expr*

Page 48

Subclause 5.1

After line 26 of page 48, add:

Constraint: The *object-name* shall be the name of a data object.

Page 49

Subclause 5.1

In the paragraph that begins “If a *length-selector*”, replace “*length-selector*” by “*char-len-param-value*”.

In the paragraph that begins “If *initialization*”, replace “NULL()” by “*pointer-initialization-expr*” twice.

Page 55

Subclause 5.1.2.4.3

In the fifth line from the bottom of page 55, delete “statement” and replace “in a” with “by”.

Page 62

Subclause 5.2.10

Replace the fifth line of R540 by

or *pointer-initialization-expr*

In the penultimate line of page 62, replace “NULL()” by “pointer association status”.

Page 63

Subclause 5.2.10

In lines 1, 7 and 10 of page 63, replace “NULL()” by “*pointer-initialization-expr*” thrice.

Page 66

Subclause 5.4

In the paragraph that begins “Any *namelist-group-name*”, replace “in more than one NAMELIST statement” by “more than once in the NAMELIST statements”.

Page 80

Subclause 6.3.1.2

In item (1) in the list in clause 6.3.1.2, change “; it” to “. It shall not be supplied as an actual argument except to certain intrinsic inquiry functions. It”.

Page 91

Subclause 7.1.4.1

In the last line of page 91, add “The optional argument shall also be present if the reference appears as an actual argument corresponding to a dummy argument with assumed character length.”.

Page 94

Subclause 7.1.6.1

Add to end of list item (6) “where the argument is not of type character with a length that is assumed or defined by an expression that is not an initialization expression,”.

In the first line of the last paragraph of page 94, replace “for a type parameter” by “that depends on a type parameter”.

Replace the last sentence of page 94 by “The prior specification may be to the left of the inquiry function in the same statement, but shall not be within the same *entity-decl.*”.

Page 96

Subclause 7.1.6.2

In the first line of the last paragraph of the subclause, replace “for a type parameter” by “that depends on a type parameter”.

Replace the second sentence of the last paragraph of the subclause by “The prior specification may be to the left of the inquiry function in the same statement, but shall not be within the same *entity-decl.*”.

Page 97

Subclause 7.1.7

In the first line of page 97, replace “in an expression” with “of an intrinsic operation”.

In the fifth line of page 97, replace “all of its components” with “it”.