



INTERNATIONAL STANDARD ISO/IEC 1539-1:2010
TECHNICAL CORRIGENDUM 1

Published 2012-10-01

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 1

Technologies de l'information — Langages de programmation — Fortran

Partie 1: Langage de base

RECTIFICATIF TECHNIQUE 1

Technical Corrigendum 1 to ISO/IEC 1539-1:2010 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 22, *Programming languages, their environments and system software interfaces*.

TECHNICAL CORRIGENDUM 1**Introduction**

At the end of the fourth item in the bulleted list (Data declaration), append the sentence: “An array or an object with a nonconstant length type parameter can have the VALUE attribute.”.

In the last item in the main bulleted list (Programs and procedures), before “An impure” insert the new sentence: “An argument to a pure procedure can have default INTENT if it has the VALUE attribute.”.

In the same paragraph, before “The FUNCTION and SUBROUTINE” insert the new sentence: “The PROTECTED attribute can be specified by the procedure declaration statement.”

Subclause 1.6.2

In the first sentence of the first paragraph of the subclause change “This” to “Except as identified in this subclause, this”.

In the second sentence of the first paragraph change “Any” to “Except as identified in this subclause, any”.

Following the first paragraph of the subclause, add new paragraphs:

Fortran 2003 specified that array constructors and structure constructors of finalizable type are finalized. This part of ISO/IEC 1539 specifies that these constructors are not finalized.

Fortran 2003 permitted an INTENT(OUT) argument of a pure subroutine to be polymorphic; that is not permitted by this part of ISO/IEC 1539.

Subclause 4.5.6.3

Move paragraph 9 of the subclause and Note 4.49 to precede paragraph 1. In addition, edit the paragraph by changing “the variable is” to “if the variable is not an unallocated allocatable variable, it is” and by appending a new sentence at the end of the paragraph: “If the variable is an allocated allocatable that would be deallocated by intrinsic assignment, the finalization occurs before the deallocation.”.

In paragraph 1 of the subclause, after “it is finalized” insert “unless it is the variable in an intrinsic assignment (7.2.1.3) or a component thereof”

Delete paragraphs 5 and 7 of the subclause.

Replace paragraph 8 of the subclause with:

When a procedure is invoked, an object that becomes argument associated with a nonpointer, nonallocatable INTENT(OUT) dummy argument of that procedure is finalized. The finalization caused by INTENT(OUT) is considered to occur within the invoked procedure; so for elemental procedures, an INTENT(OUT) argument will be finalized only if a scalar or elemental final subroutine is available, regardless of the rank of the actual argument.

Subclause 4.5.7.3

In the first paragraph of the subclause, change “as a type-bound” to “as an accessible type-bound”.

Subclause 5.5

In the fourth paragraph of the subclause, delete the sentence “The mapping may ... scoping unit.” and replace “in the outermost inclusive scope in which it appears” by “; if the outermost inclusive scope in which it appears is not a type definition, it is declared in that scope, otherwise it is declared in the host of that scope”.

Subclause 5.6

In the first sentence of the fifth paragraph, replace “type parameters, and shape” by “kind type parameters, and rank”.

Subclause 6.5.3.3.2

Replace the second paragraph of the subclause by:

A vector-subscripted array section shall not be finalized by a nonelemental final subroutine.

In the third paragraph of the subclause, replace “shall ... (16.6.7)” with “is not definable and shall not be defined or become undefined”.

Subclause 6.7.3.2

Add the following sentence to the end of the first paragraph: “An allocatable variable shall not be deallocated if it or any subobject of it is argument associated with a dummy argument or construct associated with an associate name.”.

Subclause 6.7.3.3

Add the following sentence to the end of the first paragraph: “A pointer shall not be deallocated if its target or any subobject thereof is argument associated with a dummy argument or construct associated with an associate name.”.

Subclause 7.1.11

Replace the ninth paragraph of the subclause by:

A generic entity referenced in a specification expression in the *specification-part* of a scoping unit shall have no specific procedures defined in that scoping unit, or its host scoping unit, subsequent to the specification expression.

Subclause 7.1.12

Replace the third paragraph of the subclause by:

A generic entity referenced in a constant expression in the *specification-part* of a scoping unit shall have no specific procedures defined in that scoping unit, or its host scoping unit, subsequent to the constant expression.

Subclause 8.1.6.6.4

In the first paragraph of the subclause replace the fourth item in the bulleted list with the following:

- A branch occurs within the range of a DO construct and the branch target statement is neither the *end-do* nor within the range of the same DO construct.

Subclause 8.1.6.7

In the first paragraph of the subclause, in the second item in the bulleted list replace the first sentence by: “A pointer that is used in an iteration other than as the pointer in pointer assignment, allocation, or nullification, either shall be previously pointer-assigned, allocated, or nullified in that iteration or shall not have its pointer association changed during any iteration.”.

In the third item in the bulleted list replace the second sentence by: “An allocatable object that is referenced, defined, deallocated, or has its allocation status, dynamic type, or a deferred type