INTERNATIONAL STANDARD ISO/IEC 1539-1:2010
TECHNICAL CORRIGENDUM 3

Published 2014-09-15

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 3

Technologies de l'information — Langages de programmation — Fortran

Partie 1: Langage de base

RECTIFICATIF TECHNIQUE 3

Technical Corrigendum 2 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*.

ICS 35.060 **Ref. No. ISO/IEC 1539-1:2010/Cor.3:2014(E)**

© ISO/IEC 2014 – All rights reserved

Published in Switzerland

Documents
Document Sharing Hub

Full document details: ISO_IEC 1539-1_2010_Cor 3_2014

ISO/IEC/JTC1/SC22/WG5-N2003

ISO/IEC 1539-1:2010 - TECHNICAL CORRIGENDUM 3

Introduction
In the last item in the main bulleted list (Programs and procedures), after "An empty CONTAINS section is allowed.", insert new sentence: "A PROCEDURE statement can have a double colon before the first procedure name.".

Subclause 1.3.77
Following subclause 1.3.77 add new item:

> **1.3.77a**
> **function result**
> entity that returns the value of a function

Subclause 1.3.121
Delete term **1.3.121 result variable**.

Subclause 1.3.147.6
Replace the definition of **extensible type** with:

> type that may be extended using the EXTENDS clause (4.5.7.1)

Subclause 1.6.3
In the first paragraph of the subclause, replace "Any" by "Except as identified in this subclause, any".

Delete the final sentence of the first paragraph, "The following …1539." and insert two new paragraphs:

> Fortran 95 permitted defined assignment between character strings of the same rank and different kinds. This part of ISO/IEC 1539 does not permit that if both of the different kinds are ASCII, ISO 10646, or default kind.
>
> The following Fortran 95 features might have different interpretations in this part of ISO/IEC 1539.

Subclause 1.6.4
Following the third paragraph of the subclause, insert a new paragraph:

> Fortran 90 permitted defined assignment between character strings of the same rank and different kinds. This part of ISO/IEC 1539 does not permit that if both of the different kinds are ASCII, ISO 10646, or default kind.

Subclause 2.2.3
In the second paragraph of the subclause, after "data objects" insert "or procedure pointers".

Subclause 2.3.5
In the fifth paragraph of the subclause, before Note 2.7, insert new note:

> **NOTE 2.6a**
> If the processor supports the concept of a process exit status, it is recommended that error termination initiated other than by an ERROR STOP statement supplies a processor-dependent nonzero value as the process exit status.

Subclause 4.3.1.1
Following constraint C406, insert new constraint:

> C406a (R403) In TYPE(*intrinsic-type-spec*) the *intrinsic-type-spec* shall not end with a comma.

AVA Documents
Document Sharing Hub

Full document details: ISO_IEC 1539-1_2010_Cor 3_2014

Subclause 4.3.1.2
In the second paragraph of the subclause, in the final sentence, change “function result variable” to “function result”.

Subclause 4.4.3.2
In the fifth paragraph of the subclause, in the fifth bulleted item in the list change “result variable in the function” to “function result”.

Subclause 4.5.2.1
After constraint C427 insert new constraint:

C427a (R426) The same t*ype-param-name* shall not appear more than once in a *derived-type-stmt*.

Subclause 4.5.2.3
In constraint C436, after “appears,” insert “the type shall have at least one component,”.

Subclause 4.5.3.1
In constraint C438, after “shall appear” insert “exactly once”.

Subclause 4.5.7.1
In the first paragraph of the subclause, after “A derived type” insert “, other than the type C_PTR or C_FUNPTR from the intrinsic module ISO_C_BINDING,”.

Subclause 5.1
In the second paragraph of the subclause, change “its result variable” to “the function result”.

Subclause 5.2.1
In the second paragraph of the subclause, replace constraint C507 by:

C507 (R501) If the PARAMETER keyword appears, *initialization* shall appear in each *entity-decl.*

Add new constraint:

C507a An expression that specifies a length type parameter or array bound of a named constant shall be a constant expression.

Subclause 5.2.1
Following the final paragraph of the subclause, insert a new paragraph:

If *initialization* appears for a nonpointer entity,
- its type and type parameters shall conform as specified for intrinsic assignment (7.2.1.2);
- if the entity has implied shape, the rank of *initialization* shall be the same as the rank of the entity;
- if the entity does not have implied shape, *initialization* shall either be scalar or have the same shape as the entity.

Subclause 5.3.8.1
In syntax rule R515, change “*implied-shape-spec-list*” to “*implied-shape-spec*” and insert new production:

or *implied-shape-or-assumed-size-spec*

Subclause 5.3.8.5
In the first paragraph of the subclause, replace the final sentence “An assumed-size array is declared with an *assumed-size-spec*.” with “A dummy argument is declared to be an assumed-size array by an *assumed-size-spec* or an *implied-shape-or-assumed-size-spec*.”.

Before syntax rule R521 insert new BNF term:

R520a *assumed-implied-spec* **is** [*lower-bound* :] *

Documents
Document Sharing Hub
Full document details: ISO_IEC 1539-1_2010_Cor 3_2014