

# Information technology – Programming languages – Fortran

## TECHNICAL CORRIGENDUM 1

Technical corrigendum 1 to international Standard ISO/IEC 1539:1991 (E) was prepared by Joint Technical Committee ISO/IEC JTC1, Information technology.

---

*Page xvi*

### **Lines 1 and 2**

Delete the sentence: “The section concludes ... names.”

*Page 3*

### **Subclause 1.4.1**

After item (4), add item (5) as follows:

- (5) A value of 0 for a list item in a formatted output statement will be formatted in a different form for some G edit descriptors. In addition, the Fortran 90 standard specifies how rounding of values will affect the output field form, but FORTRAN 77 did not address this issue: therefore, some FORTRAN 77 processors may produce a different output form than Fortran 90 processors for certain combinations of values and G edit descriptors.

*Page 15*

### **Subclause 2.4.6**

Third sentence, change “of” to “or”.

Page 24

#### Subclause 3.3.2.4

Lines 5 and 6, replace the text “and no other statement in the program unit may have an initial line that appears to be a program unit END statement” with “. A statement whose initial line appears to be a program unit END statement must not be continued.”.

Page 33

#### Subclause 4.4.1

Replace the text of the fourth constraint after R429 with:

The character length specified by the *char-length* in a *component-decl* or the *char-selector* in a *type-spec* (5.1, 5.1.1.5) must be a constant specification expression (7.1.6.2).

Pages 39 and 40

#### Clause 5.1

In R504, change “*function-name* [ ( *array-spec* ) ]” to “*function-name*”.

On page 40, lines 12 and 13, change “Constraint: An *array-spec* for a *function-name* that does not have the POINTER attribute must be an *explicit-shape-spec-list*” to “Constraint: An *array-spec* for an *object-name* that is a function result that does not have the POINTER attribute must be an *explicit-shape-spec-list*”.

On page 40, lines 14 and 15, delete the constraint “An *array-spec* for a *function-name* that does have the POINTER attribute must be a *deferred-shape-spec-list*”.

Page 42

#### Subclause 5.1.1.5

Replace the five lines of text following the constraints for R508 with:

The *char-selector* in a CHARACTER *type-spec* and the \* *char-length* in an *entity-decl* or in a *component-decl* of a type definition specify character length. The \* *char-length* in an *entity-decl* or a *component-decl* specifies an individual length and overrides the length specified in the *char-selector*, if any. If a \* *char-length* is not specified in an *entity-decl* or a *component-decl*, the *length-selector* or *type-param-value* specified in the *char-selector* is the character length. If the length is not specified in a *char-selector* or a \* *char-length*, the length is 1.

Page 43

#### Subclause 5.1.1.7

Line 4, change “is specified” to “is declared explicitly”.

Pages 54 and 55

#### Clause 5.3

In line 4 of the second paragraph after the constraints, in the phrase: “the default is the mapping ...”, after “default” add “for a program unit or an interface body is default integer if the letter is I, J, ... , or N and default real otherwise, and the default for an internal or module procedure”. Delete “A program ... O-Z”.

In line 3 of the following paragraph after “provided the mapping is not null.”, insert the new sentence: “Note that the mapping can be to a derived type that is inaccessible in the local scope if the derived type is accessible to the host scope.”

Change lines 2 to 4 of page 55 (the interface for function FUN) to:

```

FUNCTION FUN (I)           ! Not all entities need be
  INTEGER FUN             ! declared explicitly
END FUNCTION FUN

```

*Pages 77 and 78*

#### **Subclause 7.1.6.1**

In line 4 of item (6) on page 77, change “not assumed or” to “not assumed, are not defined by an expression that is not a constant expression, and are not”.

In line 4 of item (6) on page 78, change “not assumed or” to “not assumed, are not defined by an expression that is not an initialization expression, and are not”.

*Page 93*

#### **Subclause 7.5.3.1**

Delete the first constraint following rule R743.

#### **Subclause 7.5.3.2**

Add a new first paragraph:

In each *assignment-stmt*, the *mask-expr* and the variables being defined must be arrays of the same shape.

In line 1 of the second paragraph after “*expr*”, add “or *variable*”.

In line 1 of the third paragraph after “*expr*”, add “or *variable*”.

*Page 120*

#### **Subclause 9.4.1**

Add to the end of the list of constraints:

Constraint: If a SIZE= specifier is present, an ADVANCE= specifier also must appear.

*Page 123*

#### **Subclause 9.4.2**

In the second constraint, replace the word “scalar” with “named scalar variable”.

*Page 144*

#### **Subclause 10.5.4.1.2**

In the second paragraph, end of line 2, after “– 0.5,” add “ or N is identically 0 and *d* is 0,”.

In the second paragraph, line 5, after “*N* is identically 0” add “ and *d* is not zero”.

*Page 149*

#### **Subclause 10.8.1**

Add “ and” to the end of item (4), and add an additional item to the list after item (4):

(5) The character constant contains at least one character,

*Page 158*

**Subclause 11.3.2**

In the fifth paragraph following the constraints, lines 3 and 4, change “no entity is referenced by this name” to “the name is not used to refer to an entity”.

*Page 164*

**Subclause 12.1.2.2.1**

Add “; or” to the end of item (15), and add a new item to the list after (15):

(16) The name of a named construct

After the next paragraph, following “prior to the DATA statement.”, insert a new paragraph:

If a derived type name of a host is inaccessible, data entities of that type or subobjects of such data entities still can be accessible.

*Page 167*

**Subclause 12.3.2.1**

In the paragraph following the constraints, line 3, change “explicit interface” to “explicit specific interface”.

*Page 170*

**Subclause 12.3.2.2**

At the end of the fourth paragraph following R1207, add

A name that appears in an EXTERNAL statement must not also appear as a specific procedure name in an interface block in the scoping unit.

*Page 172*

**Subclause 12.4.1.1**

In the first paragraph, last line; Change “of the the dummy” to “of the dummy”.

At the end of the first paragraph, add: “If the dummy argument is an assumed-shape array, the rank of the dummy argument must agree with the rank of the actual argument.”

*Page 177*

**Subclause 12.5.2.4**

At end of the first paragraph, add a new sentence:

When a statement function is invoked, an instance of that statement function is created.

*Page 178*

**Subclause 12.5.2.5**

In lines 8 and 9 of the second paragraph following the constraints, in the sentence that begins, “Otherwise, they ...”, delete “all be scalars ... length or”.

Page 179

### Subclause 12.5.2.8

Add the following item to the numbered list:

- (5) If it is an array, it must not be supplied as an actual argument to an elemental procedure unless an array of the same rank is supplied as an actual argument corresponding to a nonoptional dummy argument of that elemental procedure.

In the paragraph following the numbered list, replace “It” with “Except as noted in (5) above, it”.

Page 182

### Subclause 12.5.4

In the first constraint, first sentence, change “references to scalar variables and array elements” to “references to variables” and change “intrinsic operators” to “intrinsic operations”. After first sentence add:

If *scalar-expr* contains a reference to a function or a function dummy procedure, the reference must not require an explicit interface, the function must not require an explicit interface or be a transformational intrinsic, and the result must be scalar. If an argument to a function or a function dummy procedure is array valued, it must be an array name.

Split the second constraint into two constraints. The first of the new constraints ends on line 4 with “... use or host association.” The second is the remainder of the original constraint with “scalar variable, array element” replaced by “*dummy-arg-name*, variable”.

In the fifth constraint, delete the second word: “scalar” and replace “local to” with “accessible in”.

Page 198

### Subclause 13.13.13

#### Result Type.

Add “scalar” after “default logical”.

Page 203

### Subclause 13.13.25

#### Result Value.

Case (ii), at the beginning of the third line, change “1,*sh*” to “*sh*,1”.

Page 220

### Subclause 13.13.66

#### Result Value.

Case (iii), line 4, change “, [” to “[” (i.e. delete comma).

Page 226

### Subclause 13.13.80

#### Argument.

Replace the description with “A must be the name of an optional dummy argument that is accessible in the procedure in which the PRESENT function reference appears.”

Page 234

### Subclause 13.13.100

#### Result Value.

Change “The” to “If X is not zero, the”, and change “; otherwise ” to “. Otherwise”.

Page 245

### Subclause 14.1.3

Replace the last paragraph with the following two paragraphs:

Except for a common block name or a scalar variable name, a name that identifies a global entity or local entity of class 1 (14.1.2) accessible in the scoping unit of a statement must not be the name of a statement entity of that statement. Within the scope of a statement entity, another statement entity must not have the same name.

If the name of a global or local entity accessible in the scoping unit of a statement is the same as the name of a statement entity in that statement, the name is interpreted within the scope of the statement entity as that of the statement entity. Elsewhere in the scoping unit, including parts of the statement outside the scope of the statement entity, the name is interpreted as that of the global or local entity.

Page 247

### Subclause 14.6.2.1

In item (3), subitem (c), delete “or”. In subitem (d), delete “6.3.3.2,” and change the period at the end to “, or”. Add a new subitem as follows:

- (e) After the execution of a RETURN or END statement in a procedure where the pointer was either declared or, with the exceptions described in 6.3.3.2, accessed.

Pages 255 to 261

## Annex A

**conformable**; Change “2.4.7” to “2.4.5”.

**constant**; Change “2.4.4” to “2.4.3.1.2”.

**derived type**; Change “2,” to “2)”.

**extent**; Change “2.4.7” to “2.4.5”.

**literal constant**; Change “2.4.4” to “2.4.3.1.2”.

**main program**; Change “2,” to “2)”.

**module**; Change “4,” to “4)”.

**named constant**; Change “2.4.4” to “2.4.3.1.2”.

**procedure**; Change “3,” to “3)”.

**rank**; Change “2.4.7” to “2.4.5”.

**shape**; Change “2.4.7” to “2.4.5”.

**size**; Change “2.4.7” to “2.4.5”.

**subobject**; Change “2.4.3.2” to “2.4.3.1”.

**target**; Change “specified in a” to “specified in a TARGET statement or”.

**variable**; Change “2.4.5” to “2.4.3.1.1”.