INTERNATIONAL STANDARD

ISO/IEC 1539-2

Second edition
2000-06-01

Information technology — Programming languages — Fortran —

Part 2: Varying length character strings

Technologies de l'information — Langages de programmation — Fortran — Partie 2: Chaînes de caractères de longueur variable

Reference number
ISO/IEC 1539-2:2000(E)

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 734 10 79
E-mail copyright@iso.ch
Web www.iso.ch

Printed in Switzerland

CONTENTS

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 3.

In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this part of ISO/IEC 1539 may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

International Standard ISO/IEC 1539-2 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 22, *Programming languages, their environments and system software interfaces*.

This second edition cancels and replaces the first edition (ISO/IEC 1539-2:1994), which has been technically revised. The following summarises the changes made to the facilities provided by this part of ISO/IEC 1539:

The assignment, concatenation, and comparison operations are extended to describe elemental semantics.

`CHAR` is extended to describe pure semantics.

`ADJUSTL`, `ADJUSTR`, `EXTRACT`, `IACHAR`, `ICHAR`, `INDEX`, `INSERT`, `LEN`, `LEN_TRIM`, `LGE`, `LGT`, `LLE`, `LLT`, `REMOVE`, `REPEAT`, `REPLACE`, `SCAN`, `SPLIT`, `TRIM`, `VAR_STR`, and `VERIFY` are all extended to describe elemental semantics.

ISO/IEC 1539 consists of the following parts, under the general title *Information technology — Programming languages — Fortran*:

— *Part 1: Base language*

— *Part 2: Varying length character strings*

Annexes A and B of this part of ISO/IEC 1539 are for information only.

Introduction

This part of ISO/IEC 1539 has been prepared by ISO/IEC JTC1/SC22/WG5, the technical working group for the Fortran language. This part of ISO/IEC 1539 is an auxiliary standard to ISO/IEC 1539-1 : 1997, which defines the latest revision of the Fortran language, and is the first part of the multipart Fortran family of standards; this part of ISO/IEC 1539 is the second part. The revised language defined by ISO/IEC 1539-1 : 1997 is informally known as Fortran 95.

This part of ISO/IEC 1539 defines the interface and semantics for a module that provides facilities for the manipulation of character strings of arbitrary and dynamically variable length. Annex A refers to a possible implementation, in Fortran 95, of a module that conforms to this part of ISO/IEC 1539. It should be noted, however, that this is purely for purposes of demonstrating the feasibility and portability of this standard. The actual code is not intended in any way to prescribe the method of implementation, nor is there any implication that this is in any way an optimal portable implementation. The module is merely a fairly straightforward demonstration that a portable implementation is possible.

This standard is a development from a previous version known as ISO/IEC 1539-2: 1994 that takes account of the improvements introduced in Fortran 95. The most significant improvements in Fortran 95 for the present standard were the introduction of pure and elemental procedures. Since pure and elemental functions can be used in specification expressions, their introduction in this standard enhances the usability of the standard for the end user. The ability to define many of the functions specified in this standard to be elemental improves the compatibility of these functions with similar intrinsic functions defined by the main standard.

The improvements in type initialization provided in Fortran 95 have also enabled the sample implementation referred to in Annex A to be written in such a way that significant leakage of memory is less likely to occur.

Information technology – Programming languages – Fortran – Part 2: Varying length character strings

1 Scope

This part of ISO/IEC 1539 defines facilities in Fortran for the manipulation of character strings of dynamically variable length. This part of ISO/IEC 1539 provides an auxiliary standard for the version of the Fortran language specified by ISO/IEC 1539-1: 1997 and informally known as Fortran 95.

A program that conforms with 1539-2: 1994 also conforms with this standard.

This part of ISO/IEC 1539 is an auxiliary standard to that defining Fortran 95 in that it defines additional facilities to those defined intrinsically in the primary language standard. A processor conforming to the Fortran 95 standard is not required also to conform to this part of ISO/IEC 1539. However, conformance to this part of ISO/IEC 1539 assumes conformance to the primary Fortran 95 standard.

This part of ISO/IEC 1539 prescribes the name of a Fortran module, the name of a derived data type to be used to represent varying-length strings, the interfaces for the procedures and operators to be provided to manipulate objects of this type, and the semantics that are required for each of the entities made accessible by this module.

This part of ISO/IEC 1539 does not prescribe the details of any implementation. Neither the method used to represent the data entities of the defined type nor the algorithms used to implement the procedures or operators whose interfaces are defined by this part of ISO/IEC 1539 are prescribed. A conformant implementation may use any representation and any algorithms, subject only to the requirement that the publicly accessible names and interfaces conform to this part of ISO/IEC 1539, and that the semantics are as required by this part of ISO/IEC 1539 and those of ISO/IEC 1539-1 : 1997.

It should be noted that a processor is not required to implement this part of ISO/IEC 1539 in order to be a standard conforming Fortran processor, but if a processor implements facilities for manipulating varying

length character strings, it is recommended that this be done in a manner that is conformant with this part of ISO/IEC 1539.

A processor conforming to this part of ISO/IEC 1539 may extend the facilities provided for the manipulation of varying length character strings as long as such extensions do not conflict with this part of ISO/IEC 1539 or with ISO/IEC 1539-1 : 1997.

A module, written in standard conforming Fortran, is referenced in Annex A. This module illustrates one way in which the facilities described in this part of ISO/IEC 1539 could be provided. This module is both conformant with the requirements of this part of ISO/IEC 1539 and, because it is written in standard conforming Fortran, it provides a portable implementation of the required facilities. This module is referenced for information only and is not intended to constrain implementations in any way. This module is a demonstration that at least one implementation, in standard conforming and hence portable Fortran, is possible.

It should be noted that this part of ISO/IEC 1539 defines facilities for dynamically varying length strings of characters of default kind only. Throughout this part of ISO/IEC 1539 all references to intrinsic type CHARACTER should be read as meaning characters of default kind. Similar facilities could be defined for non-default kind characters by a separate, if similar, module for each such character kind.

This part of ISO/IEC 1539 has been designed, as far as is reasonable, to provide for varying length character strings the facilities that are available for intrinsic fixed length character strings. All the intrinsic operations and functions that apply to fixed length character strings have extended meanings defined by this part of ISO/IEC 1539 for varying length character strings. Also a small number of additional facilities are defined that are appropriate because of the essential differences between the intrinsic type and the varying length derived data type.

2 Normative References

The following normative documents contain provisions which, through reference in this text, constitute provisions of this part of ISO/IEC 1539. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. However, parties to agreements based on this part of ISO/IEC 1539 are encouraged to investigate the possibility of applying the most recent editions of the normative documents indicated below. For undated references, the latest edition of the normative document referred to applies. Members of ISO and IEC maintain registers of currently valid International Standards.

ISO/IEC 646 : 1991 *Information technology – ISO 7-bit Coded character set for information interchange.*

ISO/IEC 1539-1 : 1997 *Information technology – Programming languages – Fortran – Part 1: Base language.*

3 Requirements

3.1 The Name of the Module

The name of the module shall be

`ISO_VARYING_STRING`

Programs shall be able to access the facilities defined by this part of ISO/IEC 1539 by the inclusion of `USE` statements of the form

```
USE ISO_VARYING_STRING
```

3.2 The Type

The type shall have the name

`VARYING_STRING`

Entities of this type shall represent values that are strings of characters of default kind. These character strings may be of any non-negative length and this length may vary dynamically during the execution of a program. There shall be no arbitrary upper length limit other than that imposed by the size of the processor and the complexity of the programs it is able to process. The characters representing the value of the string have positions 1,2,...,N, where N is the length of the string. The internal structure of the type shall be `PRIVATE` to the module.

3.3 Extended Meanings for Intrinsic Operators

The meanings for the intrinsic operators of:

assignment =

concatenation //

comparisons ==, /=, <, <=, >=, >

shall be extended to accept any combination of operands of type `VARYING_STRING` and type `CHARACTER`. Note that the equivalent comparison operator forms `.EQ.`, `.NE.`, `.LT.`, `.LE.`, `.GE.`, and `.GT.` also have their meanings extended in this manner.

3.3.1 Assignment

An elemental assignment of the form

var = expr

shall be defined with the following type combinations:

`VARYING_STRING` and `VARYING_STRING`
`VARYING_STRING` and `CHARACTER`
`CHARACTER` and `VARYING_STRING`

Action. The characters that are the value of the expression `expr` become the value of the variable `var`. There are two cases:

Case(i): Where the variable is of type `VARYING_STRING`, the length of the variable becomes that of the

expression.

Case(ii): Where the variable is of type `CHARACTER`, the rules of intrinsic assignment to a Fortran character variable apply. Namely, if the expression string is longer than the declared length of the character variable, only the left-most characters are assigned. If the character variable is longer than that of the string expression, it is padded on the right with blanks.

3.3.2 Concatenation

The elemental concatenation operation

`string_a // string_b`

shall be defined with the following type combinations:

`VARYING_STRING` and `VARYING_STRING`
`VARYING_STRING` and `CHARACTER`
`CHARACTER` and `VARYING_STRING`

The values of the operands are unchanged by the operation.

Result Characteristics. Of type `VARYING_STRING`.

Result Value. The result value is a new string whose characters are the same as those produced by concatenating the operand character strings in the order given.

3.3.3 Comparisons

Elemental comparisons of the form

```
string_a == string_b
string_a /= string_b
string_a < string_b
string_a <= string_b
string_a > string_b
string_a >= string_b
```

shall be defined for operands with the following type combinations:

`VARYING_STRING` and `VARYING_STRING`
`VARYING_STRING` and `CHARACTER`
`CHARACTER` and `VARYING_STRING`

The values of the operands are unchanged by the operation. Note that the equivalent operator forms `.EQ.`, `.NE.`, `.LT.`, `.LE.`, `.GE.`, and `.GT.` also have their meanings extended in this manner.

Result Characteristics. Of type default `LOGICAL`.

Result Value. The result value is true if `string_a` stands in the indicated relation to `string_b` and is false otherwise. The collating sequence used for the inequality comparisons is that defined by the processor for characters of default kind. If `string_a` and `string_b` are of different lengths, the comparison is done as if the shorter string were padded on the right with blanks.

3.4 Extended Meanings for Generic Intrinsic Procedures

The generic intrinsic procedures ADJUSTL, ADJUSTR, CHAR, IACHAR, ICHAR, INDEX, LEN, LEN_TRIM, LGE, LGT, LLT, LLE, REPEAT, SCAN, TRIM, and VERIFY shall have their meanings extended to include the appropriate argument type combinations involving VARYING_STRING and CHARACTER. Detailed descriptions of the extensions are given in this section.

3.4.1 ADJUSTL (string)

Description. Adjusts to the left, removing any leading blanks and inserting trailing blanks.

Class. Elemental function.

Argument. `string` shall be of type VARYING_STRING.

Result Characteristics. Of type VARYING_STRING.

Result Value. The result value is the same as `string` except that any leading blanks have been deleted and the same number of trailing blanks inserted.

3.4.2 ADJUSTR (string)

Description. Adjusts to the right, removing any trailing blanks and inserting leading blanks.

Class. Elemental function.

Argument. `string` shall be of type VARYING_STRING.

Result Characteristics. Of type VARYING_STRING.

Result Value. The result value is the same as `string` except that any trailing blanks have been deleted and the same number of leading blanks inserted.

3.4.3 CHAR (string [, length])

Description. Converts a varying string value to default CHARACTER.

Class. Pure transformational function.

Arguments.

`string` shall be scalar and of type VARYING_STRING.

`length` (optional) shall be scalar and of type default INTEGER.

Result Characteristics. Scalar of type default CHARACTER. If `length` is absent, the result has the same length as `string`. If `length` is present, the result has the length specified by the argument `length`.

Result Value.

Case(i): If `length` is absent, the result is a copy of the characters in the argument `string`.

Case(ii): If `length` is present, the result is a copy of the characters in the argument `string` that may have been truncated or padded. If `string` is longer than `length`, the result is truncated on the right. If `string` is shorter than `length`, the result is padded on the right with blanks. If `length` is less than one, the result is of zero length.

Note. This function is elemental in Fortran 95, where it has the form `CHAR(i [,kind])`, with `i` of type integer.

