International Standard 6093

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION•МЕЖДУНАРОДНАЯ ОРГАНИЗАЦИЯ ПО СТАНДАРТИЗАЦИИ•ORGANISATION INTERNATIONALE DE NORMALISATION

Information processing — Representation of numerical values in character strings for information interchange

Traitement de l'information — Représentation des valeurs numériques dans les chaînes de caractères pour l'échange d'information

First edition — 1985-11-01

UDC 681.3.04 **Ref. No. ISO 6093-1985 (E)**

Descriptors : data processing, information interchange, numeric representations, character sets, coded representation.

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work.

Draft International Standards adopted by the technical committees are circulated to the member bodies for approval before their acceptance as International Standards by the ISO Council. They are approved in accordance with ISO procedures requiring at least 75 % approval by the member bodies voting.

International Standard ISO 6093 was prepared by Technical Committee ISO/TC 97, *Information processing systems.*

Users should note that all International Standards undergo revision from time to time and that any reference made herein to any other International Standard implies its latest edition, unless otherwise stated.

Printed in Switzerland

Contents

This page intentionally left blank

Information processing — Representation of numerical values in character strings for information interchange

1 Scope and field of application

This International Standard specifies three presentations of numerical values, which are represented in character strings in a form readable by machine, for use in interchange between data processing systems. This International Standard also provides guidance for developers of programming language standards and implementors of programming products. These representations are recognizable by humans, and thus may be useful in communication between humans.

The base of representation is 10.

This International Standard applies only to numerical values consisting of a finite number of digits with or without the decimal mark. It does not specify the mechanism to communicate the accuracy of the number being represented or the method of delimiting the numerical representations or the organization of the numerical representations into larger aggregates.

2 Conformance

A representation of a numerical value is in conformance with this International Standard if it is one of the three representations specified herein.

A conformance statement shall identify the representation and, where applicable, specify whether COMMA or FULL STOP is used as the decimal mark. In the absence of such a statement, the FULL STOP is deemed to be the decimal mark.

3 References

ISO 646, *Information processing — ISO 7-bit coded character set for information interchange.*

ISO 2022, *Information processing — ISO 7-bit and 8-bit coded character sets — Code extension techniques.*

ISO 4873, *Information processing — 8-bit code for information interchange — Structure and rules for implementation.*

4 Definitions

For the purpose of this International Standard the following definitions apply.

4.1 decimal mark: The character that separates the digits forming the integral part of a number from those forming the fractional part.

4.2 field: A continuous string of character positions on a data carrier.

4.3 field description: The set of characteristics possessed by the field to ensure that its contents have an unique numerical interpretation to the interchange parties. For each field within a set of interchanged data the field description is specified in documentation associated with the interchange agreement between the parties. The field description includes the specification of the length of the field.

4.4 length of a field: The number of character positions of a field.

4.5 positional notation: A numeration system in which a real number is represented by a string of characters in such a way that the value contributed by a character depends on its position as well as on its value.

5 Character set

5.1 Description

The character set for the representation of numerical values shall be a sub-set of the ISO 646 coded character set.

5.2 Syntax

The following syntatic objects are defined using the method of syntax specification described in annex A.

a) digit = 0/1/2/3/4/5/6/7/8/9

b) sign = + / –

c) decimal-mark = , / .

d) space = SPACE

e) exponent-mark = E / e

5.3 Semantics

The digits shall be the characters coded in positions 3/0 to 3/9 of ISO 646.

The remaining characters shall correspond to positions 2/0 (SPACE), 2/11 (PLUS SIGN), 2/12 (COMMA), 2/13 (MINUS SIGN), 2/14 (FULL STOP), 4/5 (CAPITAL LETTER E) and 6/5 (SMALL LETTER e).

5.4 Coding

The coding of the characters is specified in ISO 646. Table 4 is reproduced from the code table for the IRV of the 7-bit coded character set in ISO 646. Additional markings in table 4 identify the sub-set of characters specified above.

6 First numerical representation (NR1)

The first numerical representation shall be a positional notation in which each number shall be represented by a string of digits, the decimal mark is implicit and its position fixed.

NOTE — This representation is also called: implicit-point representation.

6.1 Description

Each instance of an NR1 shall be composed of optional leading SPACEs followed by a sign (in the signed representation) and a string of digits. There shall be at least one digit. No embedded or trailing SPACEs shall be contained in the field.

6.2 Syntax

NR1 = unsigned-NR1/signed-NR1

unsigned-NR1 = space* digit digit*

signed-NR1 = space* (sign/space) digit digit*

6.3 Semantics

Each representation shall be contained in a field the length of which shall be equal to the sum of the number of SPACEs and the number of digits, plus 1 if a sign is present. At least one digit shall be present.

In an unsigned NR1 the value represented shall be greater than, or equal to, zero.

In a signed NR1 the PLUS SIGN can be replaced by a SPACE.

The implied decimal mark shall follow the right-most digit in the NR1, unless a scaling factor to be applied to the field is specified in accompanying documentation.

The signed representation of the numerical value zero shall contain a PLUS SIGN or a SPACE, but not a MINUS SIGN.

6.4 Examples

In the following examples the field length is assumed to be seven. The character SPACE is represented by △.

Table 1 — Examples of NR1

Common notation	Unsigned NR1	Signed NR1
4902	0004902 △△04902 △△△4902	+004902 △+04902 △△+4902 △△△4902
+1234	0001234 △△△1234	+001234 △△+1234 △△△1234
−56780	no representation	−56780 △−56780
0	0000000 △△△△△△0	+000000 △△△△△+0 △△△△△△0
1234567	1234567	no representation

7 Second numerical representation (NR2)

The second numerical representation shall be a positional notation in which each number shall be represented by a string of characters, the decimal mark is explicity indicated by a specific character.

NOTE — This representation is also called: explicit-point unscaled representation.

7.1 Description

Each instance of an NR2 shall be composed of optional leading SPACEs followed by a sign (in the signed representation) and a string of digits. There shall be at least one digit. No embedded or trailing SPACEs shall be contained in the field.

It is recommended that there is at least one digit to the left of the decimal mark even when there is at least one to the right.

7.2 Syntax

NR2 = unsigned-NR2/signed-NR2

unsigned-NR2 = (space* digit digit* decimal-mark digit*)/(space* digit* decimal-mark digit digit*)

signed-NR2 = (space* (sign/space) digit digit* decimal-mark digit*)/(space* (sign/space) digit* decimal-mark digit digit*)

7.3 Semantics

Each representation shall be contained in a field the length of which shall be equal to the sum of the number of SPACEs and of the number of digits, plus 1; or plus 2 in the signed NR2, if the sign is present. At least one digit and the decimal mark shall be present.

In an unsigned NR2 the value represented shall be greater than, or equal to, zero.

In a signed representation the PLUS SIGN can be replaced by a SPACE.

The position of the decimal mark shall represent the position of the actual decimal mark in the value, unless a scaling factor to be applied to the field is specified in accompanying documentation.

The signed representation of the numerical value zero shall contain a PLUS SIGN or a SPACE, but not a MINUS SIGN.

7.4 Examples

In the following examples the field length is assumed to be eight.

Table 2 — Examples of NR2

Common notation	Unsigned-NR2	Signed-NR2
1327.	1327.000 0001327. △△△1327.	+1327.00 △△+1327. △△△1327.
123,45	00123,45 △△123,45	△+123,45 △△123,45
1237,0	△△1237,0	△+1237,0 △△1237,0
.00001	00.00001	+0.00001
−5,678	no representation	−5,67800 −05,6780
1234,567	1234,567	no representation
0	000,0000 △△△△△0,0	+0,00000 △△△△+0,0 △△△△△0,0 △△△△△△0,

8 Third numerical representation (NR3)

The third numerical representation shall be a notation in which a number is represented by two strings of digits called significand and exponent. The value of the number equals the value of the significand multiplied by 10 raised to the power represented by the exponent.

NOTE — This representation is also called: explicit-point scaled representation.

8.1 Description

NR3 shall consist of representations of numerical values of the general form (A) E (B) which represent the value

$$A \times 10^{B}$$

where B is an integer.

In each instance of an NR3 the significand shall be composed of optional leading SPACEs, followed by an optional sign (in the signed representation) and a string of digits. There shall be at least one digit in the significand; the location of the decimal mark in the significand is explicitly specified in the character string. The character E (or e) shall follow the significand, and the exponent, preceded by its sign, shall immediately follow the character E (or e).

The exponent shall be composed of a leading sign followed by at least one digit.

No embedded or trailing SPACEs shall be contained in the field. It is recommended that there is at least one digit to the left of the decimal mark, even when there is at least one to the right.

8.2 Syntax

NR3 = unsigned-NR3/signed-NR3

unsigned-NR3 = space* significand exponent-mark exponent

signed-NR3 = space* (sign/space) significand exponent-mark exponent

significand = (digit digit* decimal-mark digit*)/(digit* decimal-mark digit digit*)

exponent = sign? digit digit*

8.3 Semantics

Each representation shall be contained in a field the length of which shall be equal to the sum of the number of SPACEs and of the number of digits, plus 4; or only 3 if the sign of the significand is represented by SPACE; or only 2 if the latter SPACE has been omitted.

In an unsigned NR3 the value represented shall be greater than, or equal to, zero.

In a signed NR3 the PLUS SIGN of the significand can be replaced by a SPACE.

If the exponent has the value zero, its sign shall be a PLUS SIGN. If the exponent is not equal to zero and if its sign is omitted, then the exponent is positive.

The representation of the numerical value zero shall contain a PLUS SIGN or a SPACE, only ZEROs in the significand, and a PLUS SIGN and only ZEROs in the exponent.

8.4 Examples

In the following examples, the field length is assumed to be eight.

Table 3 — Examples of NR3

Common notation	Signed-NR3
5600	+0,56E+4 +5.6e+03
.00003	+0,3E−04 △0,3e−04
−2,8	−2,8E+00
0	+0,0E+00 △△△0.e+0

8.5 Normalized form

An NR3 representation, in which the significand shall be a proper fraction in the range

$0,1 \leqslant \text{ABS (s)} < 1$

where ABS (s) shall be the unsigned value of the significand, is said to be normalized form. This condition may be met by appropriate selection of the value represented by the exponent.

Any given number can be represented by a unique normalized form. For example, the normalized representation of the common notation

$6,1902 \times 10^3$

includes:

— a significand 0,61902

and

— an exponent of 4.

Following the specification of NR3, this representation would be of the form

+ 0,61902E + 04

Table 4 — Code table

b7	0	0	0	0	1	1	1	1
b6	0	0	1	1	0	0	1	1
b5	0	1	0	1	0	1	0	1
b4 b3 b2 b1	0	1	2	3	4	5	6	7
0 0 0 0	0	NUL	DLE	SP	0	@	P	`	p
0 0 0 1	1	SOH	DC1	!	1	A	Q	a	q
0 0 1 0	2	STX	DC2	"	2	B	R	b	r
0 0 1 1	3	ETX	DC3	#	3	C	S	c	s
0 1 0 0	4	EOT	DC4	¤	4	D	T	d	t
0 1 0 1	5	ENQ	NAK	%	5	E	U	e	u
0 1 1 0	6	ACK	SYN	&	6	F	V	f	v
0 1 1 1	7	BEL	ETB	'	7	G	W	g	w
1 0 0 0	8	BS	CAN	(	8	H	X	h	x
1 0 0 1	9	HT	EM	)	9	I	Y	i	y
1 0 1 0	10	LF	SUB	*	:	J	Z	j	z
1 0 1 1	11	VT	ESC	+	;	K	[	k	{
1 1 0 0	12	FF	IS4	,	<	L	\	l	|
1 1 0 1	13	CR	IS3	-	=	M	]	m	}
1 1 1 0	14	SO	IS2	.	>	N	^	n	~
1 1 1 1	15	SI	IS1	/	?	O	_	o	DEL

Annex A

Method of syntax specification

(This annex forms part of the standard.)

The syntax, through a series of substitution rules, defines syntactic objects of various types, such as "integer" or "digit", and describes which strings of symbols are objects of these types.

In the syntax, capital letters, digits and (possibly hyphenated) small-letter words are used as "metanames", i.e. as names of syntactical objects. Most of these metanames are defined by substitution rules in terms of other metanames. In order that this process terminates, certain metanames are designated as "terminal" metanames, and substitution rules for them are not included in the syntax. It should be noted in particular that SPACE and the capital letter E are terminal metanames which denote themselves.

The syntactic operators are

a) the SOLIDUS, indicating that a metaname can be substituted in one of several ways;

b) the ASTERISK, indicating that the object it follows may appear any number of times, including zero times;

c) the LEFT and RIGHT PARENTHESES, used to group sequences together;

d) the QUESTION MARK, indicating that the object it follows may be omitted.

Annex B

Use of the character COMMA as decimal mark

(This annex does not form part of the standard.)

The character COMMA has been included in this International Standard because of its widespread use in European countries to represent the decimal mark within written and printed numerical character strings, and because a direct correspondence between machine-readable and human-readable representation is desirable for data recorded on certain types of machine-readable media.

Although some widely used programming languages, including FORTRAN and BASIC, do not provide explicit facilities to permit the use of COMMA as a decimal mark in the data which can be processed by compiled programs, this fact does not appear to be an overriding objection to the inclusion of COMMA in this International Standard.

If a requirement is found to exist for the use of compiled programs to process data which includes COMMA as a decimal mark, and the programming language makes no provision for such a data representation, then it is in principle possible to incorporate into the compiling system a special parameter-controlled feature. This feature would permit substitute characters to be utilized in the data in place of the standard decimal mark and other separator symbols which are explicitly specified in the programing language.

Annex C

Application to programming languages

(This annex does not form part of the standard.)

C.1 General

When numerical values, the representation of which conforms to this International Standard, are required to appear in interchanged data, the operation of the processes which produce or accept such representations is frequently specified by means of statements in a programming language.

It is recommended that a programming language that is intended to permit the specification of such processes should permit the specification of processes which can produce, and of processes which can accept, at least one of the representations NR1, NR2 and NR3. Furthermore, if the programming language permits one or more forms of an identified NR to be produced by a specified process, then it should also permit each form of that NR to be accepted by a suitably specified process.

The examples of current common programming language techniques shown in clauses C.2 to C.6 can be used to produce and accept NRs. These examples are not intended to be exhaustive.

C.2 COBOL PICTURE character strings

C.2.1 Specifications for the application of this International Standard to PICTURE character strings in the programming language COBOL are shown in table 5 and discussed in C.2.2 to C.2.10.

Table 5 — Examples of the application of this International Standard to COBOL PICTURE character strings

Form	Prototypical specification	Example	
		Actual specification	Produced NR
Unsigned NR1	Z($w-1$)9	Z(5)9	△△△△53 △△△△△0
Signed NR1	+($w-1$)9	+(5)9	∧∧∧+53 △△△△+0
	S9($w-1$)	S9(5)	+00053 +00000
Unsigned NR2	Z($w-d-2$)9.9(d)	Z(5)9.9(2)	△△△△53.26 △△△△△0.15
	Z($w-2$)9.	Z(5)9.	△△△△53. △△△△△0.
	Z($w-d-1$).9(d)	Z(5).9(2)	△△△53.26 △△△△△.15
Signed NR2	+($w-d-2$)9.9(d)	+(5)9.9(2)	△△△+53.26 △△△△+0.15
	+($w-2$)9.	+(5)9.	△△△+53. △△△△+0.
	+($w-d-1$).9(d)	+(5).9(2)	△△+53.26 △△△△+.15
Signed NR3	(not available in COBOL)		

C.2.2 In the expressions within the parentheses, w is the number of characters in the field, and d is the number of digits to the right of the decimal point.

C.2.3 The expressions within the parentheses have to be written as unsigned nonzero numerals, otherwise the parentheses are not required.

C.2.4 The field width, w, has to be sufficient to contain the value being represented, including space for the sign and decimal point, as appropriate.

C.2.5 At least one 9 has to appear in the PICTURE character string to fulfil the requirements for an NR1 or NR2.

C.2.6 The number is right justified in the field.