ECMA

EUROPEAN COMPUTER MANUFACTURERS ASSOCIATION

STANDARD ECMA-63

REPRESENTATION OF NUMERICAL VALUES IN CHARACTER STRINGS FOR INFORMATION INTERCHANGE

September 1980

Free copies of this document are available from ECMA,
European Computer Manufacturers Association
114 Rue du Rhône – 1204 Geneva (Switzerland)

ECMA

EUROPEAN COMPUTER MANUFACTURERS ASSOCIATION

STANDARD ECMA-63

REPRESENTATION OF NUMERICAL VALUES IN CHARACTER STRINGS FOR INFORMATION INTERCHANGE

September 1980

BRIEF HISTORY

Technical Committee ECMA/TC15 for Labelling and File Structure also considered the problem of the representation of numerical values in character strings. This work was undertaken in co-operation with ISO/TC97/SC15 and ANSI. It led to the development of a draft for an international standard (ISO/DIS 6093). In the USA a corresponding standard (ANSI X.3-42) was adopted and issued.

The present Standard ECMA-63 is technically identical with those mentioned above. The three representations specified are the same. It differs from these other standards in that the character COMMA has been included as alternative representation of the decimal mark. This inclusion was considered essential in view of the wide-spread usage of COMMA in Europe. The fact that some widely used programming languages do not provide explicit facilities to accommodate the use of COMMA was not considered sufficient, as it is true also for some of these programming languages and some of the agreed standard representations. Also the technical means to overcome this difficulty are felt to be relatively simple.

Furthermore, this Standard has been edited in a simple and direct form, easy to read and to understand also for non-specialists.

TABLE OF CONTENTS

1. SCOPE

This Standard ECMA-63 specifies three different representations of numerical values, which are represented in form readable by machine, for use in interchange between data processing systems. This 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 Standard applies only to numerical values consisting of a finite number of digits with or without decimal mark.

2. CONFORMANCE

A set of interchanged data is in conformance with this Standard if it uses one or more of the representations specified herein to the exclusion of any others.

3. REFERENCES

ECMA-6 7-Bit Input/Output Coded Character Set

ECMA-14 Rules for the Definition of 4-Bit Sets derived from ECMA 7-Bit Coded Character Set

ECMA-43 8-Bit Coded Character Set

4. DEFINITIONS

For the purposes of this Standard the following definitions apply.

4.1 Decimal Mark

The real or implied character that separates the digits associated with the integral part of a number from those associated with the fractional part.

4.2 Field

A continuous string of character positions on a data carrier. The number of character positions is called length of the field.

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 shall be 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 Occurence of a Field

If a set of interchanged data contains several fields which have the same field description associated with them, then these fields are considered to be multiple occurences of a single field of defined type.

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 is a sub-set of the ECMA 7-bit coded character set. 4-bit and 8-bit coded sets could be defined according to Standards ECMA-14 and ECMA-43.

5.2 Syntax (see APPENDIX I)

digit	= 0/1/2/3/4/5/6/7/8/9
sign	= +/-
decimal-mark	= ,/.
space	= SPACE
exponent	= E

5.3 Semantics

The digits are the set of arabic digits contained in the ECMA 7-bit coded character set in position 3/0 through 3/9.

The remaining characters correspond to the positions 2/0 (SPACE), 2/11 (PLUS SIGN), 2/12 (COMMA), 2/13 (MINUS SIGN), 2/14 (FULL STOP) and 4/5 (capital letter E). See also Appendi

5.4 Coding

The coding of the characters is specified in Standard ECMA-6. The Table herewith reproduces Table 2 of that Standard. Additional markings identify the sub-set of characters specified above.

5.5 Decimal Mark

The character chosen for the representation of the decimal mark shall be the same in all occurences of a field containin an explicit decimal mark.

5.6 Notation

In the examples SPACE is represented by Δ.

6. FIRST NUMERICAL REPRESENTATION (NR1)

The first numerical representation is positional notation in which each number shall be represented by a string of digits, the decimal mark being 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 always 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.

Common notation	Unsigned NR1	Signed NR1
4902	0004902	+004902
	ΔΔ04902	Δ+04902
	ΔΔΔ4902	ΔΔ+4902
		ΔΔΔ4902
+1234	0001234	+001234
	ΔΔΔ1234	ΔΔ+1234
		ΔΔΔ1234
-56780	no representation	-056780
		Δ-56780
0	0000000	+000000
	ΔΔΔΔΔΔ0	ΔΔΔΔΔ+0
		ΔΔΔΔΔΔ0
1234567	1234567	no representation

7. SECOND NUMERICAL REPRESENTATION (NR2)

The second numerical representation is a positional notation in which each number is represented by a string of characters, the decimal mark being explicitly 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 (see APPENDIX I)

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 sume 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 each occurence of the field the number of digits to the right of the decimal mark shall be the same.

In an unsigned-NR2 the value represented shall be always 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 represents 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. These examples do not represent occurences of the same field.

Common notation	Unsigned-NR2	Signed-NR2
1327.	1327.000	+1327.00
	0001327.	ΔΔ+1327.
	ΔΔΔ1327.	ΔΔΔ1327.
123,45	00123,45	Δ+123,45
	ΔΔ123,45	ΔΔ123,45
1237,0	ΔΔ1237,0	Δ+1237,0
		ΔΔ1237,0
.00001	00.00001	+0.00001
-5,678	no representa-	-5,67800
	tion	-05,6780
1234,567	1234,567	no representation
0	000,0000	+0,00000
	ΔΔΔΔΔ0,0	ΔΔΔΔ+0,0
		ΔΔΔΔΔ0,0
		ΔΔΔΔΔΔ0,

8. THIRD NUMERICAL REPRESENTATION (NR3)

The third numerical representation is 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 multipled by 10 raised to the power represented by the exponent.

NOTE

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

8.1 Description

NR3 consists 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 a sign 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 shall follow the significand, and the exponent shall immediately follow the character 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 (see APPENDIX I)

Signed-NR3 = (space* (sign/space) digit digit* decimal-mark digit* Exponent)/(space* (sign/space) digit* decimal-mark digit digit* exponent)

Exponent = E 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.

In each occurence of the field the number of digits of the significand to the right of the decimal mark shall be the same. The number of digits of the exponent shall be the same in each occurence of the field.

There is no unsigned-NR3, however the PLUS SIGN of the significand can be replaced by a SPACE.

A scaling factor can be applied to each occurence of the field, if specified in accompanying documentation.

If the exponent has the value zero, its sign shall be a PLUS SIGN.

The representation of the numerical value zero shall contain a PLUS SIGN or a SPACE but not a MINUS SIGN, and 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. These examples do not represent occurences of the same field.

Common notation	Signed-NR3
5600	+0,56E+4
	+56.E+02
.00003	+0,3E-04
	Δ0,3E-04
-2,8	-2,8E+00
0	+0,0E+00
	ΔΔΔ0.E+0

8.5 Normalized Form

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

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

where ABS(s) is the unsigned value of the significand. 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, that is, following the specifications of NR3 of this Standard, this representation would be of the form +0,61902E+04.

TABLE

b_7	0	0	0	0	1	1	1	1
b_6	0	0	1	1	0	0	1	1
b_5	0	1	0	1	0	1	0	1
b_4 b_3 b_2 b_1	0	1	2	3	4	5	6	7
0 0 0 0	0	NUL	TC_7 (DLE)	SP	0	@	P	`	p
0 0 0 1	1	TC_1 (SOH)	DC_1	!	1	A	Q	a	q
0 0 1 0	2	TC_2 (STX)	DC_2	"	2	B	R	b	r
0 0 1 1	3	TC_3 (ETX)	DC_3	#	3	C	S	c	s
0 1 0 0	4	TC_4 (EOT)	DC_4	¤	4	D	T	d	t
0 1 0 1	5	TC_5 (ENQ)	TC_8 (NAK)	%	5	E	U	e	u
0 1 1 0	6	TC_6 (ACK)	TC_9 (SYN)	&	6	F	V	f	v
0 1 1 1	7	BEL	TC_{10} (ETB)	'	7	G	W	g	w
1 0 0 0	8	FE_0 (BS)	CAN	(	8	H	X	h	x
1 0 0 1	9	FE_1 (HT)	EM	)	9	I	Y	i	y
1 0 1 0	10	FE_2 (LF)	SUB	*	:	J	Z	j	z
1 0 1 1	11	FE_3 (VT)	ESC	+	;	K	[	k	{
1 1 0 0	12	FE_4 (FF)	IS_4 (FS)	,	<	L	\	l	|
1 1 0 1	13	FE_5 (CR)	IS_3 (GS)	-	=	M	]	m	}
1 1 1 0	14	SO	IS_2 (RS)	.	>	N	^	n	~
1 1 1 1	15	SI	IS_1 (US)	/	?	O	_	o	DEL

APPENDIX I

METHOD OF SYNTAX SPECIFICATION

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:

- the SOLIDUS, indicating that a metaname can be substituted in one of several ways,
- the ASTERISK, indicating that the object it follows may appear any number of times, including zero times,
- the LEFT and RIGHT PARENTHESES, used to group sequences together.

APPENDIX II

USE OF THE CHARACTER COMMA AS DECIMAL MARK

II 1. The character COMMA has been included in this ECMA 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.

II 2. 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 over-riding objection to the inclusion of COMMA in this ECMA Standard.

II 3. 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 programming language.