DEC STANDARD 145

TITLE: DEC Representation of Data Values in ASCII Character Strings for Information Interchange Standard

ABSTRACT: This standard defines the representation of data in character strings for interchange among DEC systems. It is an extension of ANSI X3.42. American National Standard for the Representation of Numeric Values in Characters Strings for Information Interchange.

DATE	ECO #	AUTHOR	APPROVED	REV	SEC	PAGES	TITLE
27 MAY76	----	R. Abramson	C Noelcke	A	--		DEC Representation of Data Values in ASCII Character Strings for Information Interchange Standard DEC STD 145

TYPE	CODE	NUMBER	REV
A	DS	EL00145	A

INDEX: DEC Representation of Data Values In ASCII Character Strings for Information Interchange Standard

PAGE REVISION CONTROL

PAGE NO.	PAGE REVISIONS													PAGE NO.	PAGE REVISIONS												
1.0	A													19.0	A												
2.0	A													20.0	A												
3.0	A													21.0	A												
4.0	A													22.0	A												
5.0	A													23.0	A												
6.0	A																										
7.0	A																										
8.0	A																										
9.0	A																										
10.0	A																										
11.0	A																										
12.0	A																										
13.0	A																										
14.0	A																										
15.0	A																										
16.0	A																										
17.0	A																										
18.0	A																										
ECO NO.														ECO NO.													
DATE	27 MAY 76													DATE	27 MAY 76												
SEC. REV	-													SEC. REV	-												
STD REV	A													STD REV	A												

1.0 INTRODUCTION

1.1 Motivation

As customers increasingly purchase multiple different systems for the same installation, or upgrade to larger systems to satisfy expanding data processing requirements, or interconnect systems of different architectures or running different software, or run different software on the same system, it is necessary that we offer the ability for data to be moved between systems easily and efficiently. For this type of interchange to be possible, it is necessary that there be a standard representation of data written and read by software running on DEC systems.

1.2 Goals

The goals of this standard are to:

1. Identify the classes of data values which must be interchangeable among systems and between software within one system.

2. Specify the ASCII character string representation of data values on media selected for interchange.

This Standard does not specify:

1. The maximum number of characters in a data value or any of its constituent parts: this does not prohibit fixing the maximum number of characters in a data value in connection with any product or other standard.

2. The method of delimiting a data value, the organization of data values into larger aggregates (such as complex numbers, arrays or records), or units of measure, scaling, or the application associated with the data value.

3. The mechanism to communicate the accuracy of numbers being represented by a data value.

4. The means for communicating between originator and recipient sufficient information for the recovery of a data value from the carrying media: there are in existence a variety of programming language, industry oriented, and application-specific conventions which may apply.

5. The interpretation of a character string. Simply because a field has the appearance of a particular data value, the originator and recipient are not required to construe it as representing that type of data value.

6. The programming language or extra-lingual elements that permit or cause the production of a data value by a data processing system.

7. The programming language or extra-lingual elements that permit or cause the acceptance of a data value by a system independent of the originating system.

8. The representation of data values within source statements of a program.

9. The diagnostic or remedial procedures to be followed upon encountering a field which does not conform to the syntax specified in this standard.

1.3 Scope

This standard applies to all software products which produce or read character data files including terminal files. This class of products includes the object time systems of language translators and interpreters as well as editors.

1.4 History of Previous Standardization Efforts

There are no present DEC standards and no recent projects attempting the definition of standard data representations.

1.5 Related Current Standards

DEC 005-003-187-03	Draft DEC Magnetic Tape Labels and File Structure Standard
ANSI X3.4-1968	Code for Information Interchange
ANSI X3.9-1966	FORTRAN
ANSI X3.23-1974	COBOL
ANSI X3.27-1969	Magnetic Tape Labels and File structure for Information Interchange
ANSI BSRX3.42	The Representation of Numeric Values in Character Strings for Information Interchange Note: The extensions made to BSRX3.42 to create this current DEC standard are: 1) optional radix-indications in numeric data values, 2) string data values, 3) logical data values.

1.6 Future Standards Activities

The structuring of data values into records and records into files will be subjects of the Record I/O Standard sub-committee.

2.0 TERMINOLOGY

2.1 Binary Notation:

A fixed radix notation where the radix of representation is two.

2.2 Channel:

A path along which signals can be sent, e.g., data channel, output channel.

2.3 Character:

A letter, digit, or other symbol that is used as part of the organization, control or representation of data.

2.4 Character String:

A string consisting solely of characters.

2.5 Decimal Notation:

A fixed radix notation where the radix of representation is ten.

2.6 Digit:

A symbol that represents one of the non-negative integers smaller than the radix. For example, in decimal notation, a digit is one of the characters from 0 to 9. For radices greater than ten, the additional digits are the upper case alphabet with A immediately following 9.

2.7 Explicit-Point-Scaled Representation:

A number representation system in which each number, as represented by a significand and exrad, equals the value represented by the significand times the radix of representation raised to the power of the value represented by the exrad. For example, the value represented by the common notation 6.1902X10(3) contains a significand of 6.1902 and an exrad of 3, and may be represented in engineering notation by the sequence of characters 6.1902E3. Following the specifications of this standard, this representation would be of the form +0.61902E+04. Loosely, floating-point form.

2.8 Explicit-Point-Unscaled Representation:

A positional notation in which each number is represented by a sequence of characters, the position of this radix point being explicitly indicated by a special character at that position. Loosely, variable-point form.

2.9 Exrad:

In an explicit-point-scaled representation, the representation of the power to which the radix of representation is to be raised. For example, in the common notation 6.1902X10(3), the string of characters 3 is the exrad.

2.10 Field:

In a record, a specified area used for a particular category of data, e.g., a group of consecutive card columns used to represent a wage rate.

2.11 Fixed Radix Notation:

A positional representation in which the significances of successive digit positions are successive integral powers of a single radix.

2.12 Hexadecimal Notation:

A fixed radix notation where the radix of representation is 16.

2.13 Implicit-Point Representation:

A positional notation in which each number is represented by a sequence of digits, the position of the implicit radix point being fixed with respect to one end of the sequence according to some convention or specification. Loosely, fixed-point form.

2.14 Interchange:

Carrying a volume of, or transmitting by communication channel, data produced by one system (hardware, operating system, programming language, compiler, distributed library routines), associated with either a source program, a human readable description of the data, or a machine readable description of the data, to another system of possibly different architecture; the second system reading, understanding, meaningfully processing, modifying and adding to the data and then returning the data to the originating system or another system; the originating system or other system continuing to process the volume of data or data transmitted as if it had not been involved in interchange. Not all of these functions need be exercised in every interchange, but the capability to perform them should exist. In so far as the system is capable of accepting or producing data on volumes or channels. In this definition "data" specifically excludes an object program or any appurtenant table or control code.

2.15 Medium:

The material or configuration thereof, on which data are recorded, e.g., paper tape, cards, magnetic tape.

2.16 Normalized Form:

An explicit-point-scaled representation, in which the significand is a proper fraction in the range

```
          r(-1) .LE.  ABS(s) .LT.  r(Ø)
```

where r is the radix of representation and ABS(s) is the unsigned value of the significand. This condition may be met by appropriate selection of the value represented by the exrad. The significand has the property that while ABS(s) .LT. 1, ABS(s) X r .GE. 1. Any given number will be represented by a unique normalized form. For example, the normalized form representation of the common rotation 6.19Ø2X1Ø(3) would include a significand Ø.619Ø2 and an exrad of 4, that is, following the specifications of this standard, this representation would be of the form +Ø.619Ø2E+Ø4.

2.17 Number:

A mathematical entity that may indicate the quantity or amount of units. Synonymous with numeric value. Compare with numeral; numeric representation.

2.18 Numeral:

A discrete representation of a number. For example, twelve, 12, XII and 1100 are four different numerals that represent the same number. Synonymous with numeric representation. Compare with number, numeric value. Contrast with numeric word, which is an identifier.

2.19 Numeric Representation:

Same as numeral.

2.23 Numeric Value:

Same as number.

2.21 Numeric Word:

A word consisting of digits and possibly space characters and special characters, used as an identifier. For example, in the universal decimal classification system, the numeric word 621.39 + 897 is used as an identifier for a class of literature. Contrast with numeral.

2.22 Octal Notation:

A fixed radix notation where the radix of representation is 8.

2.23 Positional Notation:

A numeration system in which a number is represented by means of an ordered set of digits, such that the value contributed by each digit depends on its position as well as upon its value. Synonymous with positional representation.

2.24 Positional Representation:

Same as positional notation.

2.25 Product:

A computing system or any component of a computing system including hardware, software, communication channel and media.

2.26 Radix:

In positional representation, that integer, if it exists, by which the significance of the digit place must be multiplied to give the significance of the next higher digit place. For example, in decimal notation, the radix of each place is ten.

2.27 Radix point:

In radix notation, the real or implied character that separates the digits associated with the integral part of a numeral f- those associated with the fractional part.

2.28 Record:

A collection of related items of data, which for operating system logic purposes is treated as a unit of information. Conceptually, a record corresponds (in the context of business data) to a transaction, a customer's account, etc. In other contexts, the delineation of a record may be relatively arbitrary, and is determined by the designer of the information formats. (X3.27-1969)

A collection of related items of data, treated as a unit, for example, one line of an invoice may form a record; a complete set of such records may form a file. (X3.12-1970)

2.29 Recorded Medium:

A recorded medium is a removable device and all the machine processable information recorded upon it. (The specifications of this standard apply to the recorded information rather than to the device upon which the information is recorded.)

2.30 Sequence:

An arrangement of items according to a specified set of rules.

2.31 Significand:

In an explicit-point-scaled representation, that explicit-point-unscaled representation which is to be multiplied by; the radix of representation raised to the power represented by the exrad. For example, in the common notation 6.1902X10(3), the string of characters 6.1902 is the significand.

2.32 String:

A linear sequence of entities such as characters or physical elements.

3.0 SYNTACTIC METALANGUAGE

Syntactic specifications are linguistic equivalents of algebraic equations, being a concise means for describing the construction of elements of a language. The syntactic metalanguage used in this standard is defined in the table below.

SYMBOL(S)	MEANING
< >	The string of characters delimited by these symbols is the name assigned to a phrase or element of the language being described; the component name.
::=	The component named on the left of this symbol is to be formed by the evaluation of the syntactic expression to the right of the symbol. This symbol may be read as "is defined by ".
|	Exclusive or. A dyadic operator of a syntactic expression describing the alternative forms of a phrase or element of the language.
[...]	Optional element. The contents of the brackets may be absent from the data value representation.
[...]n	Repetition. The operation of catenation is to be performed over n occurrences of the contents of the brackets; n may assume a constant value or a (bounded) range of values.
(N)	An exrad of N.

Symbols which are not included in the set defined in the table above, and which are not enclosed in corner braces ('<' and '>') represent themselves and together constitute the alphabet (character set) of the language being described.

A syntactic expression is composed of component names, elements of the alphabet and the operators of alternation or concatenation. Concatenation has a higher hierarchical strength than alternation. Hence the expression

```
     A|BC|D
```

has the meaning "the string 'A' or the concatenation of the strings 'B' and 'C', or the string 'D'," where each 'or' is exclusive and only one element may be chosen.

Similiarly the syntactic expression

```
     <A>|<B><C>|<D>
```

defines a set of phrases which is composed of the subsets of phrases which are the totality of instances of the components <A>, <B><C> and <D>.

A construct or production takes the form of a component name, the symbol ::= and a syntactic expression. The production

```
     <X> ::= <Y>|A<Z>
```

has the meaning "the component named <X> is composed of an instance of the syntactic expression <Y>|A<Z>," if the productions

```
     <Y> ::= B|C
```

and <Z> ::= P|Q|R

were to be defined in addition, then the syntactic expression <Y>|A<Z> describes the set [B, C, AP, AQ, AR]. Hence the component named <X> has the valid forms contained in the set [B, C, AP, AQ, AR].

The number of syntactic elements associated by repeated operations of concatenation is signified by a postscript notation. In this form, the definition of a 3 digit numeral would be expressed as

```
     <3 digit integer> ::= [<digit>]3
```

In such a syntactic expression, it is not required that each instance of the similiarly named component is identical, or else it would be possible only to describe sequences of the same digit such as 333 or 888.

The index of repetition is not explicitly defined in the syntactic metalanguage hence providing for a generality of definition which is needed in this standard, and providing the means by which the application may define the index. Throughout the syntactic specifications used in this standard, the indices of repetition are related to the number of characters in the field which is to contain the value and apart from the restrictions relating to the minimum number of digits or characters in a field, the sign and minimum number of digits in an exrad, and the mandatory presence of the sign character in signed representations; the number of characters in a data value is defined by the user of the standard rather than by the standard itself.

4.0 STANDARD DATA VALUE REPRESENTATIONS

4.1 General

4.1.1 Description

A standard data value representation consists of one member of the set which is the union of the members of the sets of implicit-point, explicit-point-unscaled, explicit-point-scaled, string, and logical values.

The implicit-point and explicit-point-unscaled are further subdivided into signed and unsigned fields; only a signed form is allowed for the explicit-point-scaled set.

The sequence of characters within the representation of a data value as specified by this standard is such that the characters as read left to right by a human reader are stored into ascending memory addresses by a receiving system.

4.1.2 Character Sets

4.1.2.2 Numeric Values Representation

Implicit-point, explicit-point-unscaled, and explicit-point-scaled data values, representing numeric values, are composed of the following ASCII characters according to the syntactic rules in the remainder of this section:

```
     <Numeric-representation-characters>::=<space>|+|-|.|E|R|1|2|3
                                           |4|5|6|7|8|9|0|A
                                           |B|C|D|E|F
```

4.1.2.3 String Values Representation

String values are composed of the following characters:

```
     <String-representation-characters>::= Any ASCII
                                   character, ASCII positions
                                   40-176
```

4.1.2.4 Logical Values Representation

Logical values are composed of the following ASCII characters:

```
     <logical-representation-characters>::= F!T
```

4.1.3 Definitions of Syntactic Elements

The following syntactic elements are defined here for use in the remainder of this standard:

```
     <text>::=              Any ASCII character, ASCII positions 40-176.
     <space>::=             The   ASCII   character   whose   octal
                            representation is 040.
     <sign>::=              <space>!+!-
     <positive-sign>::=     <space>!+
     <expnt-sign>::=        +!-
     <radix-indicator>::= R<radix-digits>
     <exp>::=               E
     <digit>::=             <decimal-digit>!<octal-digit>!<hex-digit>
                            !<binary-digit>
     <decimal-digit>::=     Ø!1!2!3!4!5!6!7!8!9
     <octal-digit>::=       Ø!1!2!3!4!5!6!7
     <hex-digit>::=         Ø!1!2!3!4!5!6!7!8!9!A!B!C!D!E!F
     <binary-digit>::=      Ø!1
     <radix-point>::=       .
     <radix-digits>::=      2!8!10!16
```

4.1.4 Further Definitions

4.1.4.1 Length of <radix-indicator>

Certain data values described in this standard may include an optional <radix-indicator> component. In describing the field length of such data values, the function "LENGTH" is specified. The LENGTH function gives the current character string length of the immediately following component name within the data value under discussion.

4.1.4.2 Effect of <radix-indicator>

The occurence of a terminal radix-indicator in the representation of a numeric value determines the set of digits which may appear in the remainder of the represented value. If a radix-indicator of R2 appears, then only binary digits may occur in the associated string. If R8 appears, only octal digits may occur, etc. In the absence of an explicit radix-indicator, decimal notation is the default. A default notation other than decimal may be used on prior agreement of sender and receiver.

4.2 Implicit-Point Data Value Representation

4.2.1 Description

Each instance of an implicit-point data value is composed of optional leading spaces followed by a sign (in the signed representation) and a string of digits and an optional radix-indicator. There is at least one digit, and no embedded or trailing spaces are contained in the field.

The signed representation of the numeric value zero contains a positive-sign.

4.2.2 Syntactic Specification of Implicit-Point Data Values

```
<unsigned-implicit-point>::=[<space>]s[<digit>]b[<radix-indicator>]

<signed-implicit-point>::=[<space>]s<sign>[<digit>]b[<radix-indicator>]
```

4.2.3 Permitted Values of Repetition Indices in Implicit-Point Data Values

Each instance of an implicit-point data value conforms to all of the following conditions:

In the following section, the field length (number of character positions in the field) is defined to be w, an integer.

4.2.3.1 In an unsigned implicit-point value:

```
s+b+LENGTH<radix-indicator>=w
LENGTH<radix-indicator>=Ø!2!3
s .GE. Ø
b .GE. 1
w .GE. 1
```

4.2.3.2 In a signed implicit-point value:

```
s+b+LENGTH<radix-indicator>+1=w
LENGTH<radix-indicator>=Ø!2!3
s .GE. Ø
b .GE. 1
w .GE. 2
```

4.2.4 The Represented Value

The following attributes of an implicit-point value are assumed in the following examples.

4.2.4.1 The radix of the common notation is 10.

4.2.4.2 In an unsigned implicit-point-value the value being represented and transmitted is greater than or equal to zero.

4.2.4.3 The implied radix point follows the right-most <digit>. This does not preclude specification in accompanying documentation of a scaling factor to be applied to each occurrence of the field.

4.2.5 Examples

In the following examples, the field length (number of character positions in the field) is assumed to be eight. These examples are not intended to be explusive of all other possible representations of the intended values. An asterisk (*) appears next to the preferred representation of each value.

Common Notation	Unsigned	Signed
4901	00004901	+0004901
	004901	+04901
	4901*	+4901
		4901*
	11445R8	+11445R8
9.0	00000009	+0000009
	9*	+9
		9*
	001001R2	+01001R2
1,000,000	01000000	+1000000
	1000000*	1000000*
+57821.	00057821	+0057821
	57821*	+57821
		57821*
	E1DDR16	+E1DDR16
-49330	no representation	-0049330
		-49330*
0	00000000	+0000000
	0000	0000000
	0*	+000

```
                                   000
                                    +0
                                     0+
                              but not
                              +0000000
```

4.2.6 Maximum Value of Implicit-Point Data Values

The maximum value of an implicit-point data value which can be represented on an interchange medium is implementation-dependent. However, for systems conforming to the standard, the absolute value of this maximum must not be less than 2047 (decimal).

4.3 Explicit-Point-Unscaled Data Value Representation

4.3.1 Description

Each instance of an explicit-point-unscaled data value is composed of optional leading spaces followed by a sign (in the signed representation) and a string of digits and an optional radix-indicator. There is at least one digit, the location of the .adix point is explicitly specified in the character string in the same character position in each occurence of the field, and no embedded or trailing spaces are contained in the field. When not otherwise mandatory, it is a recommended practice that there always exist at least one digit to the left of the radix point.

The signed representation of the numeric value zero contains a positive sign (+ or <space>).

4.3.2 Syntactic Specification of Explicit-Point-Unscaled Data Values

```
<unsigned-explicit-point-unscaled>::=[<space>]a[<digit>]b<radix-point>
                                     [<digit>]d[<radix-indicator>]

<signed-explicit-point-unscaled>::=[<space>]a<sign>[<digit>]b
                                   <radix-point>[<digit>]d
                                   [<radix-indicator>]
```

4.3.3 Permitted Values of Repetition Indices in Explicit-Point-Unscaled Data Values

Each instance of an explicit-point-unscaled data value conforms to all of the following conditions:

In the following section, the field length (number of character positions in the field) is defined to be w, an integer.

4.3.3.1 In an unsigned explicit-point-unscaled data value:

```
s+b+d+LENGTH<radix-indicator>+1=w
LENGTH<radix-indicator>=0|2|3
s .GE. 0
b .GE. 0
d .GE. 0
b+d .GE. 1
w .GE. 2
```

4.3.3.2 In a signed explicit-point-unscaled data value:

```
s+b+d+LENGTH<radix-indicator>+2=w
LENGTH<radix-indicator>=0|2|3
s .GE. 0
b .GE. 0
d .GE. 0
b+d .GE. 1
w .GE. 3
```

4.3.3.3 - The value of d is a parameter of the field; that is, it is the same in each occurrence of the field.

4.3.3.4 - It is not mandatory, but it is a recommended practice that b .GE. 1, even when d .GE. 1.

4.3.4 The Represented Value

The following attributes of an explicit-point-unscaled value are assumed in the following examples:

4.3.4.1 The radix of the common notation is 10.

4.3.4.2 - In an unsigned explicit-point-unscaled value the value being represented and transmitted is greater than or equal to zero.

4,3,4,3 = The position of the explicit radix point represents the position of the actual radix point in the explicit=point=unscaled value. This does not preclude specification in accompanying documentation of a scaling factor to be applied to each occurrence of the field.

4,3,5 Examples

In the following examples, the field length (number of character positions in the field) is assumed to be eight. These examples are not intended to be exclusive of all other possible representations of the intended values. An asterisk (*) appears next to the preferred representation of each value.

Common Notation	Unsigned	Signed
4901	4901,000	+4901,00
	0004901,	+4901,
	4901,*	4901,*
	11445,R8	
9,01	9,010000	+9,01000
	00009,01	+9,010
	9,0100	+9,01
	9,01*	9,01*
,000001	,0000010	+,000001
	0,000001*	,000001
	,000001	0,000001*
=4,933	no representation	=4,93300
		=004,933
		=4,9330
		=4,933*
0	,0000000	+,000000
	0,000000	+0,00000
	0,0	+00000,0
	,0	+000000,
	0,*	+0,0
		0,*

4,3,6 Maximum Value of Explicit=point=Unscaled Data Values

The maximum value of an explicit=point=unscaled data value which can be represented on an interchange medium is implementation=dependent. However, for systems conforming to this Standard, the absolute value of this maximum must not be less than 2047 (decimal).

4.4 Explicit-Point-scaled Data Value Representation

4.4.1 Description

Each instance of an explicit-point-scaled value is composed of optional leading spaces, followed by a sign character and a string of digits. The number of digits in the significand is greater than or equal to one, the location of the radix point in the significand is explicitly specified in the character string in the same character position in each occurrence of the field, the character E following the significand in the same character position in each occurrence of the field, and the sign of the exrad immediately follows the character E. The number of digits in the exrad is greater than or equal to one, and no embedded or trailing spaces are contained in the field. A radix-indicator may optionally follow the string of exrad digits. When not otherwise mandatory, it is a recommended practice that there always exist at least one digit to the left of the radix point.

The representation of the numeric value zero contains a positive sign, and all zero digits in the significand; and a plus sign (+) and all zero digits in the exrad.

4.4.2 Syntactic Specification of Explicit-point-scaled Data Values

```
<explicit-point-scaled>::=[<space>]s<sign>[<digit>]b
                          <radix-point>[<digit>]d<exp><exrad-sign>
                          [<digit>]f[<radix-indicator>]
```

4.4.3 Permitted Values of Repetition Indices in Explicit-Point-Scaled Data Values

Each instance of an explicit-point-scaled data value conforms to all of the following conditions:

In the following section, the field length (number of character positions in the field) is defined to be w, an integer.

4.4.3.1 In a signed explicit-point-scaled data value:

```
s+b+d+f+LENGTH<radix-indicator>+4=w
LENGTH<radix-indicator>=0|2|3
s .GE. 0
b .GE. 0
d .GE. 0
b+d .GE. 1
f .GE. 1
w .GE. 6
```

4.4.3.2 - The value of d is a parameter of the field; that is, it is the same in each occurrence of the field.

4.4.3.3 - The value of f is a parameter of the field; that is, it is the same in each occurrence of the field.

4.4.3.4 - It is not mandatory, but it is a recommended practice that b .GE. 1, even when d .GE. 1.

4.4.4 The Represented Values

The following attributes of an explicit-point-scaled value are assumed in the following examples:

4.4.4.1 - The radix of representation of the common notation of both the significand and the exrad is 10.

4.4.4.2 - An unsigned explicit-point-scaled value does not appear.

4.4.4.3 - The position of the explicit radix point in the significand represents the position of the actual radix point in the explicit-point-scaled item. This does not preclude specification in accompanying documentation of an additional scaling factor to be applied to each occurrence of the field.

4.4.4.4 - The implied radix point of the exrad follows the right-most digit in the exrad.

4.4.4.5 - The representation of the numeric value zero must contain a positive sign, and all zero digits in the significand; and a plus sign (+) and all zero digits in the exrad.

4.4.5 Examples

In the following examples, the field length (number of character positions in the field) is assumed to be twelve and the number of digits in the exrad is assumed to be two.

These examples are not intended to be exclusive of all other possible representations of the intended values. An asterisk (*) appears next to the preferred representation of each value.

Common Notation	Signed Explicit-Point-Scaled
49	+49.E+00* +.49E+02 .49E+02 +31.E+00R16
9.1	.91E+01 9.1E+00*
.00001	+0.1E-04 .1E-04*
-4.9	-4.9E+00* -.49E+01
0	+0.0E+00* but not -0.0E+00 +0.0E-00 +0.0E-99

4.4.6 Maximum Value of Explicit-point-Scaled Data Values

The maximum value of an explicit-point-scaled data value which can be represented on an interchange medium is implementation-dependent. However, systems conforming to this Standard must support at a minimum a range of values of 1.97 X 10(-31) to 1.7 X 10(38).

4.5 String Data Value Representation

4.5.1 Description

Each instance of a string-data-item is composed of text string-representation-characters. There is at least one such character. Note that space is a string-representation-character and hence allowed in the field.

By prior agreement between sender and receiver, string-data-values may be delimited by double quotes. In such cases, every occurence of the double quote character other than those initiating and terminating will be indicated by two concatenated double quote characters.

4,5,2 Syntactic Specification of String Data Values

```
<string-data-value>::=[<text>]b
```

In all instances of string-data-values, the field length (number of character positions in the field) w is equal to b.

4,5,4 Examples Without Double Quote Delimiters

Common Notation	String Representation
NEW JERSEY	NEW JERSEY
+57821	+57821
"LITERAL"	"LITERAL"

4,5,5 Examples With Double Quote Delimiters

Common Notation	String Representation
New Jersey	"New Jersey"
"LITERAL"	"""LITERAL"""

4,5,6 Maximum Length of String Data Values

The maximum length of a string-data value which may be represented on an interchange medium is implementation-independent. However, this maximum must not be less than 72.

4,6 Logical Data Item Representation

4,6,1 Description

Each instance of a logical data item is composed of optional leading spaces followed by a single logical-representation-character.

4,6,2 Syntactic Specification of Logical Data Items

```
<logical-data-item>::=[<space>]s<logical-representation-character>
```

4.6.3 Permitted Values of Repetition Indices

```
S .GE. 0

W .EQ. S+1

W .GE. 1
```

4.6.4 Examples

```
                T
                F
```