

Edit Masks - EM Parameter

This chapter describes how you can specify an edit mask for an alphanumeric or numeric field.

The following topics are covered:

- Use of EM Parameter
 - Edit Masks for Numeric Fields
 - Edit Masks for Alphanumeric Fields
 - Length of Fields
 - Edit Masks for Date and Time Fields
 - Customizing Separator Character Displays
 - Examples of Edit Masks
 - Further Examples of Edit Masks
-

Use of EM Parameter

With the session parameter EM you can specify an edit mask for an alphanumeric or numeric field, that is, determine character by character the format in which the field values are to be output. Using the session parameter EMU, you can define edit masks with Unicode characters in the same way as described below for the EM session parameter.

Example:

```
DISPLAY NAME (EM=X^X^X^X^X^X^X^X^X^X)
```

In this example, each X represents one character of an alphanumeric field value to be displayed, and each ^ represents a blank. If displayed via the DISPLAY statement, the name JOHNSON would appear as follows:

```
J O H N S O N
```

You can specify the session parameter EM

- at report level (in a FORMAT statement),
- at statement level (in a DISPLAY, WRITE, INPUT, MOVE EDITED or PRINT statement) or
- at element level (in a DISPLAY, WRITE or INPUT statement).

An edit mask specified with the session parameter EM will override a default edit mask specified for a field in the DDM; see *Using the DDM Editor Windows, Specifying Extended Field Attributes*.

If EM=OFF is specified, no edit mask at all will be used.

An edit mask specified at statement level will override an edit mask specified at report level.

An edit mask specified at element level will override an edit mask specified at statement level.

Edit Masks for Numeric Fields

An edit mask specified for a field of format N, P, I, or F must contain at least one 9 or Z. If more nines or Zs exist, the number of positions contained in the field value, the number of print positions in the edit mask will be adjusted to the number of digits defined for the field value. If fewer nines or Zs exist, the high-order digits before the decimal point and/or low-order digits after the decimal point will be truncated.

For further information, see session parameter EM, *Edit Masks for Numeric Fields* in the *Parameter Reference* documentation.

Edit Masks for Alphanumeric Fields

Edit masks for alphanumeric fields must include an X for each alphanumeric character that is to be output.

With a few exceptions, you may add leading, trailing and insertion characters (with or without enclosing them in apostrophes).

The circumflex character (^) is used to insert blanks in edit mask for both numeric and alphanumeric fields.

For further information, see session parameter EM, *Edit Masks for Alphanumeric Fields* in the *Parameter Reference* documentation.

Length of Fields

It is important to be aware of the length of the field to which you assign an edit mask.

- If the edit mask is longer than the field, this will yield unexpected results.
- If the edit mask is shorter than the field, the field output will be truncated to just those positions specified in the edit mask.

Examples:

Assuming an alphanumeric field that is 12 characters long and the field value to be output is JOHNSON, the following edit masks will yield the following results:

Edit Mask	Output
EM=X.X.X.X.X	J.O.H.N.S
EM=***XXXXXX***	***JOHNSO**

Edit Masks for Date and Time Fields

Edit masks for date fields can include the characters D (day), M (month) and Y (year) in various combinations.

Edit masks for time fields can include the characters H (hour), I (minute), S (second) and T (tenth of a second) in various combinations.

In conjunction with edit masks for date and time fields, see also the date and time system variables.

Customizing Separator Character Displays

Natural programs are used in business applications all over the world. Depending on the local conventions, it is usual to present numeric data fields and those with a date or time content in a special output style, when displayed in I/O statements. The different appearance should not be realized by alternate program coding that is processed selectively as a function of the locale where the program is being executed, but should be carried out with the same program image in conjunction with a set of runtime parameters to specify the decimal point character and the "thousands separator character".

The following topics are covered below:

- Decimal Separator
- Dynamic Thousands Separator
- Examples

Decimal Separator

The Natural parameter DC is available to specify the character to be inserted in place of any characters used to represent the decimal separator (also called "radix" character) in edit masks. This parameter enables the users of a Natural program or application to choose any (special) character to separate the integer positions from the decimal positions of a numeric data item and enables, for example, U.S. shops to use the decimal point (.) and European shops to use the comma (,).

Dynamic Thousands Separator

To structure the output of a large integer values, it is common practice to insert separators between every three digits of an integer to separate groups of thousands. This separator is called a "thousands separator". For example, shops in the United States generally use a comma for this purpose (1,000,000), whereas shops in Germany use the period (1.000.000), in France a space (1 000 000), etc.

In a Natural edit mask, a "dynamic thousands separator" is a comma (or period) indicating the position where thousands separator characters (defined with the THSEPCH parameter) are inserted at runtime. At compile time, the Natural profile parameter THSEP or the option THSEP of system command COMPOPT enables or disables the interpretation of the comma (or period) as dynamic thousands separator.

If THSEP is set to OFF (default), any character used as thousands separator in the edit mask is treated as literal and displayed unchanged at runtime. This setting retains downwards compatibility.

If THSEP is set to ON, any comma (or period) in the edit mask is interpreted as dynamic thousands separators. In general, the dynamic thousands separator is a comma, but if the comma is already in use as decimal character (DC), the period is used as dynamic thousands separator.

At runtime the dynamic thousands separators are replaced by the current value of the THSEPCH parameter (thousands separator character).

Examples

A Natural program that is cataloged with parameter settings DC= ' . ' and THSEP=ON uses the edit mask (EM=ZZ , ZZZ , ZZ9 . 99).

Parameter Settings at Runtime	Displays as
DC= ' . ' and THSEPCH= ' , '	1 , 234 , 567 . 89
DC= ' , ' and THSEPCH= ' . '	1 . 234 . 567 , 89
DC= ' , ' and THSEPCH= ' / '	1 / 234 / 567 , 89
DC= ' , ' and THSEPCH= ' ' '	1 234 567 , 89
DC= ' , ' and THSEPCH= ' ' ' ' '	1 ' 234 ' 567 , 89

Examples of Edit Masks

Some examples of edit masks, along with possible output they produce, are provided below.

In addition, the abbreviated notation for each edit mask is given. You can use either the abbreviated or the long notation.

Edit Mask	Abbreviation	Output A	Output B
EM=999 . 99	EM=9 (3) . 9 (2)	367 . 32	005 . 40
EM=ZZZZZ9	EM=Z (5) 9 (1)	0	579
EM=X ^ XXXXX	EM=X (1) ^ X (5)	B LUE	A 19379
EM=XXX . . . XX	EM=X (3) . . . X (2)	BLU . . . E	AAB . . . 01
EM=MM . DD . YY	*	01 . 05 . 87	12 . 22 . 86
EM=HH . II . SS . T	**	08 . 54 . 12 . 7	14 . 32 . 54 . 3

* Use a date system variable.

** Use a time system variable.

For further information about edit masks, see the session parameter EM in the *Parameter Reference*.

Example Program without EM Parameters

```

** Example 'EDITMX01': Edit mask (using default edit masks)
*****
DEFINE DATA LOCAL
1 VIEWEMP VIEW OF EMPLOYEES
  2 NAME
  2 JOB-TITLE
  2 SALARY (1:3)
  2 CITY
END-DEFINE
*
READ (3) VIEWEMP BY NAME STARTING FROM 'JONES'
  DISPLAY 'N A M E'      NAME          /
          'OCCUPATION'  JOB-TITLE
          'SALARY'      SALARY (1:3)
          'LOCATION'     CITY

  SKIP 1
END-READ
END
    
```

Output of Program EDITMX01:

The output of this program shows the default edit masks available.

```

Page          1                               04-11-11  14:15:54

          N A M E                SALARY          LOCATION
          OCCUPATION
-----
JONES
MANAGER                46000 TULSA
                      42300
                      39300

JONES
DIRECTOR                50000 MOBILE
                      46000
                      42700

JONES
PROGRAMMER              31000 MILWAUKEE
                      29400
                      27600
    
```

Example Program with EM Parameters

```

** Example 'EDITMX02': Edit mask (using EM)
*****
DEFINE DATA LOCAL
1 VIEWEMP VIEW OF EMPLOYEES
  2 NAME
  2 FIRST-NAME
  2 JOB-TITLE
  2 SALARY (1:3)
END-DEFINE
*
READ (3) VIEWEMP BY NAME STARTING FROM 'JONES'
  DISPLAY 'N A M E'      NAME          (EM=X^X^X^X^X^X^X^X^X^X^X^X^X^X^X^X^X) /
          FIRST-NAME    (EM=...X(10)...)
          'OCCUPATION'  JOB-TITLE    (EM=' ____ 'X(12))
    
```

```
'SALARY'      SALARY (1:3) (EM=' USD 'ZZZ,999)
SKIP 1
END-READ
END
```

Output of Program EDITMX02:

Compare the output with that of the previous program (*Example Program without EM Parameters*) to see how the EM specifications affect the way the fields are displayed.

Page 1 04-11-11 14:15:54

N A M E FIRST-NAME	OCCUPATION	SALARY
J O N E S ..VIRGINIA ...	___ MANAGER	USD 46,000 USD 42,300 USD 39,300
J O N E S ..MARSHA ...	___ DIRECTOR	USD 50,000 USD 46,000 USD 42,700
J O N E S ..ROBERT ...	___ PROGRAMMER	USD 31,000 USD 29,400 USD 27,600

Further Examples of Edit Masks

See the following example programs:

- *EDITMX03 - Edit mask (different EM for alpha-numeric fields)*
- *EDITMX04 - Edit mask (different EM for numeric fields)*
- *EDITMX05 - Edit mask (EM for date and time system variables)*