# EM - Edit Mask

With this session parameter, you can specify an edit mask for an input and/or output field that is used in a `DEFINE DATA`, `DISPLAY`, `INPUT`, `MOVE EDITED`, `PRINT`, `PROCESS PAGE` or `WRITE` statement.

The parameter `EM` can also be used with U format fields. For information on Unicode format, see also *Unicode and Code Page Support in the Natural Programming Language*, *Session Parameters*, `EMU`, `ICU`, `LCU`, `TCU` versus `EM`, `IC`, `LC`, `TC`.

| Possible settings | | See below. |
|---|---|---|
| Default setting | none | |
| Applicable statements: | FORMAT | |
| | DEFINE DATA DISPLAY INPUT PRINT WRITE | Parameter may be specified at statement level and/or at element level. |
| | MOVE EDITED | Parameter may be specified at element level. |
| Applicable command: | none | |

The following topics are covered below:

- Syntax

- Blanks in Edit Masks

- Default Edit Masks

- Edit Masks for Numeric Fields

- Edit Masks for Alphanumeric Fields

- Edit Masks for Binary Fields - Format B

- Hexadecimal Edit Masks

- Edit Masks for Date and Time Fields - Formats D and T

- Edit Masks for Logical Fields - Format L

See also *Edit Masks - EM Parameter* in the *Programming Guide*.

# Syntax

For input fields, values must be entered exactly matching the edit mask. If you would like to display the edit mask for an input field, the field should be defined as modifiable (`AD=M`).

For a database field, a default edit mask may have been defined in the DDM. If you specify with the `EM` parameter an edit mask for a database field, this edit mask specified will be used instead of any default edit mask which may be defined for the field in the DDM.

If you specify `EM=OFF` for a field, no edit mask will be used for the field, not even one that may be defined in the DDM.

At statement level of a `DISPLAY`, `FORMAT`, `INPUT` or `WRITE` statement, no detail field edit mask may be specified, except `EM=OFF`.

An edit mask overrides any settings for the session parameters `AL`, `NL` and `SG`.

The characters `9`, `H`, `X` and `Z` represent significant print positions in numeric (`9` , `Z`), hexadecimal (`H`), and alphanumeric (`X`) edit masks. For the difference between `9` and `Z`, see *Edit Masks for Numeric Fields*, below.

### Examples:

```
DISPLAY AA(EM=OFF) AB(EM=XX.XX)
WRITE SALARY (EM=ZZZ,ZZ9)
```

An abbreviated notation can be used for long edit masks. The following examples demonstrate the abbreviated notation which may be used for numeric, hexadecimal, and alphanumeric edit masks:

```
EM=9(4)-9(5)  is equivalent to: EM=9999-99999
EM=H(10)      is equivalent to: EM=HHHHHHHHHH
EM=X(6)..X(3) is equivalent to: EM=XXXXXX..XXX
```

# Blanks in Edit Masks

Blanks within an edit mask are represented by the character on your keyboard that in hexadecimal code corresponds to `H'20'` (ASCII) or `H'5F'` (EBCDIC), that is, the character `^` (or ¬).

# Default Edit Masks

If no edit mask is specified for a field, a default edit mask is assigned to the field depending on the field format:

| Field Format | Default Edit Mask |
|---|---|
| A | X |
| B | H |
| N, P, I | Z9 |
| F | scientific representation |
| D | depends on default date format (as set with the profile parameter DTFORM) |
| T | HH:II:SS |
| L | blank / X |

# 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 9s or Zs exist than 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 9s or Zs exist, the high-order digits before the decimal point and/or low-order digits after the decimal point will be truncated.

The following topics are covered below:

- Characters for the Definition of Numeric Edit Masks

- Sign Characters

- Literal Leading Characters

- Literal Insertion and Trailing Characters

- Trailing Sign Characters

- Examples of Numeric Edit Masks

## Characters for the Definition of Numeric Edit Masks

| Character | Function |
|---|---|
| 9 | Position to be displayed (one digit of the field value). |
| . (period) | The first period inserted is used as a decimal point. Subsequent periods are treated as literal characters.<br><br>**Note:**<br>At this point, the period represents the sign currently defined as decimal point character. If another character is choosen (for example, a comma) with the session or profile parameter DC, this character is to be used instead. |
| Z | Zero suppression for leading zeros. This is the default for numeric fields. The letter Z may be repeatedly specified to represent floating zero suppression. Z must not be specified to the right of the decimal point. A zero value may be displayed as blanks using all Zs in the edit mask (see also session parameter ZP). |

The 9s or Zs can be preceded by one or more other characters.

## Sign Characters

If the first character before the 9s or Zs is +, -, S or N, a sign may be displayed:

| Character | Function |
|---|---|
| + | A floating sign is to be displayed preceding (leading sign character) or following (trailing sign character) the number. The sign may be generated as a plus or minus depending on the value of the field. |
| - | A floating minus is to be displayed preceding (leading sign character) or following (trailing sign character) the number if the value of the field is negative. |
| S | A sign is to be displayed to the left of the column. A plus sign is displayed for a positive value and a minus sign is displayed for a negative value. |
| N | A minus sign is to be displayed to the left of the column if the value of the field is negative. |

## Literal Leading Characters

Any number of literal leading characters can appear before the first displayable position (as indicated by Z or 9). These must follow any sign character. If there is no sign character and the first literal leading character is +, -, S or N, it must be enclosed in apostrophes. If a literal leading character is H, X, Z or 9, it must be enclosed in apostrophes.

The first literal leading character specified will appear in the output only if the value contains leading zeros and the edit mask is defined with Z (leading zero suppression). This character will then be used as a filler character displayed instead of a blank for leading zeros. Subsequent literal leading characters will be displayed as they are input.

## Literal Insertion and Trailing Characters

Literal insertion and trailing characters can also be used. The symbol (^) can be used to represent a leading, inserted, or trailing blank. By enclosing significant characters (9, H, Z, X) in apostrophes, it is possible to use any characters as leading, insertion, or trailing characters. Insignificant edit mask characters need not be enclosed in apostrophes. Within the same edit mask notation, it is possible to have groups of leading, insertion, and/or trailing character strings, some of which are bounded by apostrophes and some of which are not.

## Trailing Sign Characters

A trailing sign character can be specified for numeric edit masks by using the + or − character as the last character in the edit mask. A + will produce a trailing + or − sign depending on the value of the field. A − will produce a trailing space or − sign depending on the value of the field. If a leading and trailing sign are specified in the edit mask, both will be produced.

## Examples of Numeric Edit Masks

The table below lists the results obtained from the original values shown at the top of each column as they are output without editing mask. All values used as column headings represent format N fields. The lines below the top column represent the formats obtained using the different editing masks:

| Value | 0000.03 (N4.2) | −0054 (N4) | +0087 (N4) | 0962 (N4) | 1830 (N4) |
|---|---|---|---|---|---|
| **Edit Mask** | | | | | |
| EM=9.9 | 0.0 | 4. | 7. | 2. | 0. |
| EM=99 | 00 | 54 | 87 | 62 | 30 |
| EM=S99 | +00 | −54 | +87 | +62 | +30 |
| EM=+Z9 | +0 | −54 | +87 | +62 | +30 |
| EM=−9.99 | 0.03 | −4. | 7. | 2. | 0. |
| EM=N9 | 0 | −4 | 7 | 2 | 0 |
| EM=*9.99 | 0.03 | 4. | 7. | 2. | 0. |
| EM=Z99 | 00 | 54 | 87 | 962 | 830 |
| EM=*DMZZ9.9 | DM**0.0 | DM*54. | DM*87. | DM962. | DM830. |
| EM=999+ | 000+ | 054− | 087+ | 962+ | 830+ |
| EM=999− | 000 | 054− | 087 | 962 | 830 |
| IC=$ EM=ZZZ.99 | $.03 | $54. | $87. | $962. | $830. |
| **EM=H(6)** | | | | | |
| - *ASCII:* | 303030303033 | 30303574 | 30303837 | 30393632 | 31383330 |
| - *EBCDIC:* | F0F0F0F0F0F3 | F0F0F5D4 | F0F0F8F7 | F0F9F6F2 | F1F8F3F0 |

By combining edit masks with the parameters `IC` and `TC`, negative numbers can be displayed in varying formats using a `DISPLAY` statement.

# Edit Masks for Alphanumeric Fields

An alphanumeric edit mask which is only to be used with A format fields must contain at least one `X` which represents a character to be displayed. An `H` as the first character designates a hexadecimal edit mask. A blank is represented by a (`^`) symbol. All other characters except parentheses are permissible including leading, trailing, and insertion characters. It is also possible to specify leading, insertion, or trailing characters enclosed within apostrophes. If the character `X`, a closing parenthesis, or a quotation mark is specified as an insertion character, it must be enclosed within apostrophes.

If leading characters are used before the first displayable position `X` of an alphanumeric edit mask, the first of these leading characters will not be displayed, but is used as filler character and replaces all leading blanks in the alphanumeric output field.

**Example:**

```
DEFINE DATA LOCAL
1 #X (A4)  INIT <'  34'>
END-DEFINE
WRITE #X (EM=*A:X:)
   6X #X (EM=*A:XX:)
   6X #X (EM=*A:XXX:)
   6X #X (EM=*A:XXXX:)
   6X #X (EM=1234XXXX5678)
END
```

**Output Produced:**

```
A:*:      A:**:      A:**3:      A:**34:      23411345678
```

Trailing characters which immediately follow the last permissible print position will be displayed.

If the number of positions specified with the mask is smaller than the field length, the overhanging field content is not displayed.

If the number of positions specified with the mask is higher than the field length, the mask is truncated on the first overhanging position.

**Example:**

```
DEFINE DATA LOCAL
 1 #TEXT  (A4) INIT <'BLUE'>
END-DEFINE
WRITE #TEXT (EM=X-X-X)      displays as 'B-L-U'       .. only three bytes of field displayed
WRITE #TEXT (EM=X-X-X-X-X)  displays as 'B-L-U-E-'    .. mask definition was truncated to (EM=X-X-X-X-)
```

## Example of Alphanumeric Edit Masks

The following program lists the alphanumeric edit masks for a field that is defined with format/length A4 and contains the value `BLUE`.

```
** Example 'EMMASK1': Edit mask
***********************************************************************
DEFINE DATA LOCAL
1 #TEXT  (A4)
END-DEFINE
*
ASSIGN #TEXT = 'BLUE'
WRITE NOTITLE 'MASK 1:' 5X #TEXT (EM=X.X.X.X)
       /       'MASK 2:' 5X #TEXT (EM=X^X^X^X)
       /       'MASK 3:' 5X #TEXT (EM=X--X--X)
       /       'MASK 4:' 5X #TEXT (EM=X-X-X-X-X-X)
       /       'MASK 5:' 5X #TEXT (EM=X' 'X' 'X' 'X)
       /       'MASK 6:' 5X #TEXT (EM=XX....XXX)
       /       'MASK 7:' 5X #TEXT (EM=1234XXXX)
END
```

Output of Program `EMMASK1`:

```
MASK 1:     B.L.U.E
MASK 2:     B L U E
MASK 3:     B--L--U
MASK 4:     B-L-U-E-
MASK 5:     B L U E
MASK 6:     BL....UE
MASK 7:     234BLUE
```

# Edit Masks for Binary Fields - Format B

Edit masks for binary fields may be set using X or H notation. For binary fields, the X notation is supported as if H had been specified instead of X.

# Hexadecimal Edit Masks

If the character H is specified as the first character in an edit mask, the content of an alphanumeric or numeric field will be displayed in hexadecimal format. Each H represents two print positions that will occur for each byte in the source field. Characters other than H serve as insertion or trailing characters in the mask. The number of positions to be displayed will be adjusted to the length of the edit mask if the mask is shorter than the field. The length of the edit mask will be adjusted to the length of the field if the field length is shorter than the edit mask.

Insertion or trailing characters may be optionally specified bounded by apostrophes.

All fields displayed with a hexadecimal edit mask are treated as alphanumeric. Therefore, if the edit mask is shorter than the field to be edited, numeric or alphanumeric positions will be displayed from left to right disregarding any decimal point positions.

If a hexadecimal edit mask is used as an input edit mask, every 0-9, a-f, A-F, blank and hex zero are accepted as a hex digit.

**Note:**
Blank and hex zero are regarded as 0 and a lower-case letter (a-f) is regarded as an upper-case letter.

## Edit Mask Examples for Hexadecimal Fields:

The tables below list the hexadecimal edit masks with results obtained from the original fields and values shown above each column. All numeric values (-10, +10, 01) to which edit masks have been applied originated in fields defined with N2 format. The alphanumeric value AB originated from a field defined with format/length A2.

### ASCII:

| Value => | AB | -10 | +10 | 01 |
|---|---|---|---|---|
| EM=HH | 4142 | 3170 | 3130 | 3031 |
| EM=H^H | 41 42 | 31 70 | 31 30 | 30 31 |
| EM=HH^H | 4142 | 3170 | 3130 | 3031 |
| EM=H-H | 41-42 | 31-70 | 31-30 | 30-31 |
| EM=H | 41 | 31 | 31 | 30 |

### EBCDIC:

| Value => | AB | -10 | +10 | 01 |
|---|---|---|---|---|
| EM=HH | C1C2 | F1D0 | F1F0 | F0F1 |
| EM=H:H | C1 C2 | F1 D0 | F1 F0 | F0 F1 |
| EM=HH:H | C1C2 | F1D0 | F1F0 | F0F1 |
| EM=H-H | C1-C2 | F1-D0 | F1-F0 | F0-F1 |
| EM=H | C1 | F1 | F1 | F0 |

### Example Program Using Hexadecimal Edit Masks:

```
** Example 'EMMASK2': Edit mask
************************************************************************
DEFINE DATA LOCAL
1 #TEXT1 (A2)
1 #TEXT2 (N2)
END-DEFINE
*
ASSIGN #TEXT1 = 'AB'
ASSIGN #TEXT2 =  10
*
WRITE NOTITLE
        'MASK (EM=HH)  :' 18T #TEXT1 (EM=HH)    30T #TEXT2 (EM=HH)
      / 'MASK (EM=H^H) :' 18T #TEXT1 (EM=H^H)   30T #TEXT2 (EM=H^H)
      / 'MASK (EM=HH^H):' 18T #TEXT1 (EM=HH^H)  30T #TEXT2 (EM=HH^H)
      / 'MASK (EM=H-H) :' 18T #TEXT1 (EM=H-H)   30T #TEXT2 (EM=H-H)
      / 'MASK (EM=H)   :' 18T #TEXT1 (EM=H)     30T #TEXT2 (EM=H)
END
```

Output of Program EMMASK2 (ASCII):

```
MASK (EM=HH)  :  4142        3130
MASK (EM=H^H) :  41 42       31 30
MASK (EM=HH^H):  4142        3130
MASK (EM=H-H) :  41-42       31-30
MASK (EM=H)   :  41          31
```

Output of Program EMMASK2 (EBCDIC):

```
MASK (EM=HH)  :  C1C2        F1F0
MASK (EM=H^H) :  C1 C2       F1 F0
MASK (EM=HH^H):  C1C2        F1F0
MASK (EM=H-H) :  C1-C2       F1-F0
MASK (EM=H)   :  C1          F1
```

# Edit Masks for Date and Time Fields - Formats D and T

In edit masks for fields which are defined with format D (date) or T (time), the characters described in the following sections can be specified.

- Date - Format D, and Time - Format T

- Syntactical Restrictions for Date Characters

- Hints for Input Edit Mask

- Hints for Week Display (WW or ZW) in Output Edit Mask

- Time - Format T - only

- Examples of Date and Time Edit Masks

## Date - Format D, and Time - Format T

| Character | Usage |
|---|---|
| DD | Day. |
| ZD | Day, with zero suppression. |
| MM | Month. |
| ZM | Month, with zero suppression. |
| YYYY | Year, 4 digits (see the section *Hints for Input Edit Mask*). |
| YY | Year, 2 digits (see the section *Hints for Input Edit Mask*). |
| Y | Year, 1 digit. Must not be used for input fields. |
| WW | Number of week (see the sections *Hints for Input Edit Mask* and *Hints for Week Display in Output Edit Mask*). |
| ZW | Number of week, with zero suppression (see the sections *Hints for Input Edit Mask* and *Hints for Week Display in Output Edit Mask*). |
| JJJ | Julian day. |
| ZZJ | Julian day with zero suppression. |
| NN... or N(*n*) | Name of day (language-dependent). The maximum length is determined by the number of Ns or by *n*. If the name is longer than the maximum length, it will be truncated; if it is shorter, the actual length of the name will be used. |
| O | Number of week day. The profile parameter DTFORM determines whether Monday or Sunday is considered the first day of the week. With DTFORM=U: (Sunday = 1, Monday = 2, etc.). With DTFORM=*other*: (Monday = 1, Tuesday = 2, etc.). |
| LL... or L(*n*) | Name of month (language-dependent). The maximum length is determined by the number of L characters or by *n*. If the name is longer than the maximum length, it will be truncated; if it is shorter, the actual length of the name will be used. |
| R | Year in Roman numerals (maximum 13 digits). Must not be used for input fields. The upper limit for displayable year values is 2887. |

## Syntactical Restrictions for Date Characters

For *Input* and *Output* edit masks, you *may not* use the following:

| text | | | characters | | |
|---|---|---|---|---|---|
| month | with | month name | MM or ZM | with | LL or L(*n*) |
| day name | with | week day number | NN or N(*n*) | with | O |

For *Input* edit masks, you *may not* use the following:

| text | | | characters | | |
|---|---|---|---|---|---|
| 1-digit year | nor | a year in Roman numerals | `Y` | nor | `R` |
| Day | without | month or month name | `DD` or `ZD` | without | `MM` or `ZM` or `LL` or `L(n)` |
| Week | without | year | `WW` or `ZW` | without | `YYYY` or `YY` |
| Month | without | year | `MM` or `ZM` | without | `YYYY` or `YY` |
| Julian day | without | year | `JJJ` or `ZZJ` | without | `YYYY` or `YY` |
| Day name | without | week | `NN` or `N(n)` | without | `WW` or `ZW` |
| Week day number | without | week | `O` | without | `WW` or `ZW` |
| Julian day | with | month | `JJJ` or `ZZJ` | with | `MM` or `ZM` |
| Julian day | with | week | `JJJ` or `ZZJ` | with | `WW` or `ZW` |
| Month | with | week | `MM` or `ZM` | with | `WW` or `ZW` |

## Hints for Input Edit Mask

The range of valid year values (`YYYY`) is `1582 - 2699`. If the profile parameter `MAXYEAR` is set to `9999`, the range of valid year values is `1582 - 9999`.

If only year (`YY` or `YYYY`) but no month or day is specified within an input edit mask, the values for month and day will both be set to `01`. If only year (`YY` or `YYYY`) and month (`MM`) but no day is specified within an input edit mask, the value for day will be set to `01`.

If a 2-digits year (`YY`) is used, the century used to fill up the year representation is the current century by default. However, this does not apply when a Sliding or Fixed Window is set. For more details, please refer to profile parameter `YSLW` in the Natural Parameter Reference documentation.

If a week number (`WW` or `ZW`) but no number of week day (`O`) or name of day (`NN...`) is specified, the first day of the week is assumed.

## Hints for Week Display (WW or ZW) in Output Edit Mask

When `DTFORM=U` (USA format) is set, the week starts on Sunday; whereas for all other `DTFORM` settings the first weekday is Monday. Whether a week is week 52/53 of the old year or week 01 of the new year depends on which year contains more days of the week. In other words, if Thursday (Wednesday for `DTFORM=U`) of that week is in the previous year, the week belongs to the previous year; if it is in the next year, the week belongs to the next year.

If the number of week (`WW` or `ZW`) and a year representation (`YYYY` or `YY` or `Y`) is in the same edit mask, the display for year always corresponds to the week number, regardless of the year in the underlying date field.

### Example:

```
DEFINE DATA LOCAL
1 D (D)
END-DEFINE
MOVE EDITED '31-12-2003' TO D(EM=DD-MM-YYYY)
DISPLAY D(EM=DD-MM-YYYY_N(10)) D(EM=DD-MM-YYYY/WW)
```

Although the underlying date is the 31 Dec. 2003, when the week number WW is contained in the edit mask, it displays as:

```
        D                D
--------------------  -------------
31-12-2003_Wednesday  31-12-2004/01
```

## Time - Format T - only

| Character | Usage |
|-----------|-------|
| T | Tenths of a second. |
| SS | Seconds. |
| ZS | Seconds, with zero suppression. |
| II | Minutes. |
| ZI | Minutes, with zero suppression. |
| HH | Hours. |
| ZH | Hours, with zero suppression. |
| AP | AM/PM element. |

## Examples of Date and Time Edit Masks

```
** Example 'EMDATI': Edit mask for date and time variables
************************************************************************
*
WRITE NOTITLE
  'DATE INTERNAL :' *DATX (DF=L) /
  '              :' *DATX (EM=N(9)' 'ZW.'WEEK 'YYYY)  /
  '              :' *DATX (EM=ZZJ'.DAY 'YYYY)        /
  '     ROMAN    :' *DATX (EM=R)   /
  '     AMERICAN :' *DATX (EM=MM/DD/YYYY)      12X 'OR  ' *DAT4U /
  '     JULIAN   :' *DATX (EM=YYYYJJJ)         15X 'OR  ' *DAT4J /
  '     GREGORIAN:' *DATX (EM=ZD.''L(10)''YYYY) 5X 'OR  ' *DATG ///
*
  'TIME INTERNAL :' *TIMX                       14X 'OR  ' *TIME /
  '              :' *TIMX (EM=HH.II.SS.T) /
  '              :' *TIMX (EM=HH.II.SS' 'AP) /
  '              :' *TIMX (EM=HH)
END
```

Output of Program EMDATI:

```
DATE INTERNAL : 2005-01-12
              : Wednesday  2.WEEK 2005
              :  12.DAY 2005
     ROMAN    : MMV
```

```
    AMERICAN : 01/12/2005            OR   01/12/2005
    JULIAN   : 2005012               OR   2005012
    GREGORIAN: 12.January2005        OR   12January  2005


TIME INTERNAL : 16:04:14            OR   16:04:14.8
              : 16.04.14.8
              : 04.04.14 PM
              : 16
```

# Edit Masks for Logical Fields - Format L

For fields of format L (logical fields), edit masks can be defined as follows:

---
(**EM**=[*false-string*/]*true-string*)

---

The *false-string* must not be longer than 31 characters.

## Example of Edit Masks for Logical Field

```
** Example 'EMLOGV': Edit mask for logical variables
************************************************************************
DEFINE DATA LOCAL
1 #SWITCH (L)  INIT <true>
1 #INDEX  (I1)
END-DEFINE
*
FOR #INDEX 1 5
  WRITE NOTITLE #SWITCH (EM=FALSE/TRUE) 5X 'INDEX =' #INDEX
  WRITE NOTITLE #SWITCH (EM=OFF/ON)     7X 'INDEX =' #INDEX
  IF #SWITCH
    MOVE FALSE TO #SWITCH
  ELSE
    MOVE TRUE TO #SWITCH
  END-IF
  /*
  SKIP 1
END-FOR
END
```

Output of Program EMLOGV:

```
TRUE       INDEX =    1
ON         INDEX =    1

FALSE      INDEX =    2
OFF        INDEX =    2

TRUE       INDEX =    3
ON         INDEX =    3

FALSE      INDEX =    4
OFF        INDEX =    4

TRUE       INDEX =    5
ON         INDEX =    5
```