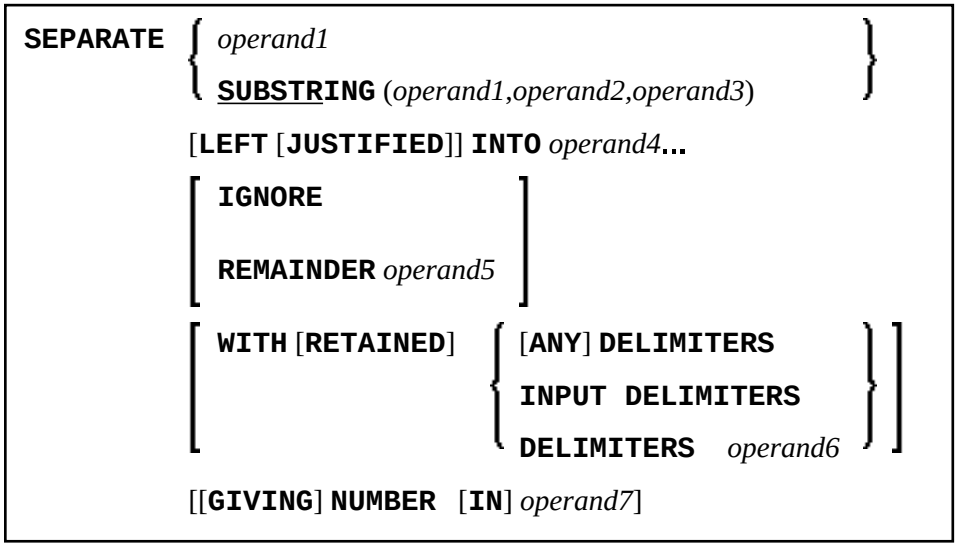
# SEPARATE




This chapter covers the following topics:

- Function
- Syntax Description
- Examples

For an explanation of the symbols used in the syntax diagram, see *Syntax Symbols*.

Related Statements: COMPRESS | COMPUTE | EXAMINE | MOVE | MOVE ALL | RESET

Belongs to Function Group: *Arithmetic and Data Movement Operations*

---

## Function

The SEPARATE statement is used to separate the content of an alphanumeric or binary operand into two or more alphanumeric or binary operands (or into multiple occurrences of an alphanumeric or binary array).

## Syntax Description

Operand Definition Table:

| Operand | Possible Structure | | | | | Possible Formats | | | | | | | | | | | | Referencing Permitted | Dynamic Definition |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *operand1* | C | S | | | | A | U | | | | B | | | | | | | yes | no |
| *operand2* | C | S | | | | | | N | P | I | B* | | | | | | | yes | no |
| *operand3* | C | S | | | | | | N | P | I | B* | | | | | | | yes | no |
| *operand4* | | S | A | G | | A | U | | | | B | | | | | | | yes | yes |
| *operand5* | | S | | | | A | U | | | | B | | | | | | | yes | yes |
| *operand6* | C | S | | | | A | U | | | | B | | | | | | | yes | no |
| *operand7* | | S | | | | | | N | P | I | | | | | | | | yes | yes |

* Format B of *operand2* and *operand3* may be used only with a length of less than or equal to 4.

Syntax Element Description:

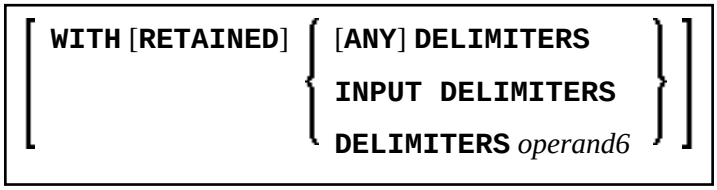
| Syntax Element | Description |
|---|---|
| *operand1* | **Source Operand:**<br><br>*operand1* is the alphanumeric/binary constant or variable whose content is to be separated.<br><br>Trailing blanks in *operand1* are removed before the value is processed (even if the blank is used as a delimiter character; see also the DELIMITERS option). |
| SUBSTRING | **SUBSTRING Option:**<br><br>Normally, the whole content of a field is separated, starting from the beginning of the field.<br><br>The SUBSTRING option allows you to separate only a certain part of the field. After the field name (*operand1*) in the SUBSTRING clause you specify first the starting position (*operand2*) and then the length (*operand3*) of the field portion to be separated. For example, if a field #A contained CONTRAPTION, SUBSTRING(#A,5,3) would contain RAP.<br><br>**Note:**<br>If you omit *operand2*, the starting position is assumed to be 1. If you omit *operand3*, the length is assumed to be from the starting position to the end of the field. |
| LEFT JUSTIFIED | **LEFT JUSTIFIED Option:**<br><br>This option causes leading blanks which may occur between the delimiter character and the next non-blank character to be removed from the target operand. |

| Syntax Element | Description |
|---|---|
| *operand4* | **Target Operand:**<br><br>*operand4* represents the target operands. If an array is specified as target operand, it is filled occurrence by occurrence with the separated values.<br><br>The number of target operands corresponds to the number of delimiter characters (including trailing delimiter characters) in *operand1*, plus 1.<br><br>If *operand4* is a dynamic variable, its length may be modified by the SEPARATE operation. The current length of a dynamic variable can be ascertained by using the system variable *LENGTH.<br><br>For general information on dynamic variables, see the section *Using Dynamic and Large Variables*. |
| IGNORE /<br><br>REMAINDER *operand5* | **IGNORE / REMAINDER Options:**<br><br>If you do not specify enough target fields for the source value to be separated into, you will receive an appropriate error message.<br><br>To avoid this, you have two options:<br><br>● **IGNORE Option:**<br><br>If you specify IGNORE, Natural will ignore it if there are not enough target operands to receive the source value.<br><br>● **REMAINDER Option:**<br><br>If you specify REMAINDER *operand5*, that section of the source value which could not be placed into target operands will be placed into *operand5*. You may then use the content of *operand5* for further processing, for example in a subsequent SEPARATE statement.<br><br>See also *Example 3*. |
| DELIMITERS | **DELIMITERS Option:**<br><br>See *DELIMITERS Option* below. |

| Syntax Element | Description |
|---|---|
| RETAINED | **RETAINED Option:**<br><br>Normally, the delimiter characters themselves are not moved into the target operands.<br><br>When you specify RETAINED, however, each delimiter (that is, either default delimiters and blanks, or the delimiter specified with *operand6*) will also be placed into a target operand.<br><br>Example:<br><br>The following SEPARATE statement would place 150 into #B, + into #C, and 30 into #D:<br><br>`...`<br>`MOVE '150+30' TO #A`<br>`SEPARATE #A INTO #B #C #D WITH RETAINED DELIMITER '+'`<br>`...`<br><br>See also *Example 3*. |
| GIVING NUMBER *operand7* | **GIVING NUMBER Option:**<br><br>This option causes the number of filled target operands (including those filled with blanks) to be returned in *operand7*. The number actually obtained is the number of delimiters plus 1.<br><br>If you use the *IGNORE Option*, the maximum possible number returned in *operand7* will be the number of target operands (*operand4*).<br><br>If you use the *REMAINDER Option*, the maximum possible number returned in *operand7* will be the number of target operands (*operand4*) plus *operand5*. |

## DELIMITERS Option:

Delimiter characters within *operand1* indicate the positions at which the value is to be separated.




Syntax Element Description:

| Syntax Element | Description |
|---|---|
| `WITH [ANY] DELIMITERS` | If you omit the `DELIMITERS` option or specify `WITH ANY DELIMITERS`, a blank and any character which is neither a letter nor a numeric character will be treated as delimiter character. |
| `WITH INPUT DELIMITERS` | Indicates that the blank and the default input delimiter character (as specified with the session parameter `ID`) is to be used as delimiter character. |
| `WITH DELIMITERS` *`operand6`* | Indicates that each of the characters specified in *`operand6`* is to be treated as delimiter character.<br><br>If *`operand6`* contains trailing blanks, these will be ignored. |

# Examples

- Example 1 - Various Samples
- Example 2 - Using an Array
- Example 3 - Using REMAINDER/RETAINED Options

## Example 1 - Various Samples

```
** Example 'SEPEX1': SEPARATE
************************************************************************
DEFINE DATA LOCAL
1 #TEXT1   (A6) INIT <'AAABBB'>
1 #TEXT2   (A7) INIT <'AAA BBB'>
1 #TEXT3   (A7) INIT <'AAA-BBB'>
1 #TEXT4   (A7) INIT <'A.B/C,D'>
1 #FIELD1A (A6)
1 #FIELD1B (A6)
1 #FIELD2A (A3)
1 #FIELD2B (A3)
1 #FIELD3A (A3)
1 #FIELD3B (A3)
1 #FIELD4A (A3)
1 #FIELD4B (A3)
1 #FIELD4C (A3)
1 #FIELD4D (A3)
1 #NBT     (N1)
1 #DEL     (A5)
END-DEFINE
*
WRITE NOTITLE 'EXAMPLE A (SOURCE HAS NO BLANKS)'
SEPARATE #TEXT1 INTO #FIELD1A #FIELD1B GIVING NUMBER #NBT
WRITE     / '=' #TEXT1 5X '=' #FIELD1A 4X '=' #FIELD1B 4X '=' #NBT
*
WRITE NOTITLE /// 'EXAMPLE B (SOURCE HAS EMBEDDED BLANK)'
SEPARATE #TEXT2 INTO #FIELD2A #FIELD2B GIVING NUMBER #NBT
WRITE     / '=' #TEXT2 4X '=' #FIELD2A 7X '=' #FIELD2B 7X '=' #NBT
*
WRITE NOTITLE /// 'EXAMPLE C (USING DELIMITER ''-'')'
SEPARATE #TEXT3 INTO #FIELD3A #FIELD3B WITH DELIMITER '-'
WRITE     /    '=' #TEXT3 4X '=' #FIELD3A 7X '=' #FIELD3B
*
MOVE ',/' TO #DEL
```

```
WRITE NOTITLE /// 'EXAMPLE D USING DELIMITER' '=' #DEL
*
SEPARATE #TEXT4 INTO #FIELD4A #FIELD4B
         #FIELD4C #FIELD4D WITH DELIMITER #DEL
WRITE     /    '=' #TEXT4 4X '=' #FIELD4A 7X '=' #FIELD4B
          /              19X '=' #FIELD4C 7X '=' #FIELD4D
*
END
```

### Output of Program SEPEX1:

```
EXAMPLE A (SOURCE HAS NO BLANKS)

#TEXT1: AAABBB     #FIELD1A: AAABBB    #FIELD1B:           #NBT:  1



EXAMPLE B (SOURCE HAS EMBEDDED BLANK)

#TEXT2: AAA BBB    #FIELD2A: AAA       #FIELD2B: BBB       #NBT:  2



EXAMPLE C (USING DELIMITER '-')

#TEXT3: AAA-BBB    #FIELD3A: AAA       #FIELD3B: BBB



EXAMPLE D USING DELIMITER #DEL: ,/

#TEXT4: A.B/C,D    #FIELD4A: A.B       #FIELD4B: C
                   #FIELD4C: D         #FIELD4D:
```

## Example 2 - Using an Array

```
** Example 'SEPEX2': SEPARATE (using array variable)
************************************************************************
DEFINE DATA LOCAL
1 #INPUT-LINE (A60) INIT <'VALUE1,   VALUE2,VALUE3'>
1 #FIELD      (A20/1:5)
1 #NUMBER     (N2)
END-DEFINE
*
SEPARATE #INPUT-LINE LEFT JUSTIFIED INTO #FIELD (1:5)
                     GIVING NUMBER IN #NUMBER
*
WRITE NOTITLE #INPUT-LINE //
              #FIELD (1)  /
              #FIELD (2)  /
              #FIELD (3)  /
              #FIELD (4)  /
              #FIELD (5)  /
              #NUMBER
*
END
```

### Output of Program SEPEX2:

```
VALUE1,   VALUE2,VALUE3

VALUE1
VALUE2
VALUE3


  3
```

## Example 3 - Using REMAINDER/RETAINED Options

```
** Example 'SEPEX3': SEPARATE (with REMAINDER, RETAIN option)
************************************************************************
DEFINE DATA LOCAL
1 #INPUT-LINE (A60) INIT <'VAL1,   VAL2, VAL3,VAL4'>
1 #FIELD      (A10/1:4)
1 #REM        (A30)
END-DEFINE
*
WRITE TITLE LEFT 'INP:' #INPUT-LINE /
            '#FIELD (1)' 13T '#FIELD (2)' 25T '#FIELD (3)'
        37T '#FIELD (4)' 49T 'REMAINDER'
       /    '----------' 13T '----------' 25T '----------'
        37T '----------' 49T '------------------------------'
*
SEPARATE #INPUT-LINE INTO #FIELD (1:2)
         REMAINDER #REM WITH DELIMITERS ','
WRITE #FIELD(1) 13T #FIELD(2) 25T #FIELD(3) 37T #FIELD(4) 49T #REM
*
RESET #FIELD(*) #REM
SEPARATE #INPUT-LINE INTO #FIELD (1:2)
         IGNORE WITH DELIMITERS ','
WRITE #FIELD(1) 13T #FIELD(2) 25T #FIELD(3) 37T #FIELD(4) 49T #REM
*
RESET #FIELD(*) #REM
SEPARATE #INPUT-LINE INTO #FIELD (1:4) IGNORE
        WITH RETAINED DELIMITERS ','
WRITE #FIELD(1) 13T #FIELD(2) 25T #FIELD(3) 37T #FIELD(4) 49T #REM
*
RESET #FIELD(*) #REM
*
SEPARATE SUBSTRING(#INPUT-LINE,1,50) INTO #FIELD (1:4)
         IGNORE WITH DELIMITERS ','
WRITE #FIELD(1) 13T #FIELD(2) 25T #FIELD(3) 37T #FIELD(4) 49T #REM
*
END
```

### Output of Program SEPEX3:

```
INP: VAL1,   VAL2, VAL3,VAL4
#FIELD (1)  #FIELD (2)  #FIELD (3)  #FIELD (4)  REMAINDER
----------  ----------  ----------  ----------  ------------------------------
VAL1           VAL2                              VAL3,VAL4
VAL1           VAL2
VAL1        ,             VAL2     ,
VAL1           VAL2      VAL3       VAL4
```