RESIZE

```
RESIZE  ⎰ dynamic-clause ⎱ [GIVING operand5]
        ⎱ array-clause   ⎰
```

This chapter covers the following topics:

- Function

- Syntax Description

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

Related Statements: EXPAND | REDUCE

Belongs to Function Group: *Memory Management Control for Dynamic Variables or X-Arrays*.

Function

The RESIZE statement is used to adjust:

- the size of a dynamic variable (*dynamic-clause*), or

- the number of occurrences of X-arrays (*array-clause*).

For further information, see also the following sections in the *Programming Guide*:

Using Dynamic Variables
Allocating/Freeing Memory Space for a Dynamic Variable
X-Arrays
Storage Management of X-Group Arrays

Syntax Description

Operand Definition Table:

Operand	Possible Structure			Possible Formats													Referencing Permitted	Dynamic Definition
operand1		S	A		A	U				B								no	no
operand2	C	S						I										no	no
operand3			A	G	A		N	P	I	F	B	D	T	L	C	G	O	yes	no
operand4	C	S					N	P	I									no	no
operand5		S						I4										no	yes

Syntax Element Description:

dynamic-clause	The RESIZE DYNAMIC statement adjusts the allocated length of the currently allocated storage of a dynamic variable (*operand1*) to the value specified with *operand2*. For more information, see *Dynamic Clause* below.
operand1	*operand1* is the dynamic variable for which the length is to be adjusted.
operand2	*operand2* is used to specify the new length of the dynamic variable. The value specified must be a non-negative numeric integer constant or a variable of type Integer4 (I4).
array-clause	The RESIZE ARRAY statement adjusts the number of occurrences of the X-array (*operand3*) to the upper and lower bound specified with (`dim[,dim[,dim]]`). For more information, see *Array Clause* below.
operand3	*operand3* is the X-array. The occurrences of the X-array can be expanded or reduced. The index notation of the array is optional. As index notation only the complete range notation `*` is allowed for each dimension.
dim *operand4*	The lower and upper bound notation (*operand4* or asterisk) to which the X-array should be expanded is specified here. If the current value of the upper or lower bound should be used, an asterisk (*) must be specified in place of *operand4*. For more information, see *Dimension* below.
GIVING *operand5*	If the GIVING clause is not specified, Natural runtime error processing is triggered if an error occurs. If the GIVING clause is specified, *operand5* contains the Natural message number if an error occurred, or zero upon success.

Dynamic Clause

```
[SIZE OF] DYNAMIC [VARIABLE] operand1 TO operand2
```

The RESIZE DYNAMIC statement adjusts the allocated length of a dynamic variable (*operand1*) to the value specified with *operand2*.

When the RESIZE statement is used, the currently allocated storage size will be adjusted to the requested values, regardless wether it must be increased or decreased.

Array Clause

```
[AND RESET] [OCCURRENCES OF] ARRAY operand3 TO (dim[,dim[,dim]])
```

The RESIZE ARRAY statement adjusts the number of occurrences of the X-array (*operand3*) to the upper and lower bound specified with (`dim[,dim[,dim]]`).

The RESET option resets all occurrences of the resized X-array to its default zero value. By default (no RESET option), the actual values are kept and the resized (new) occurrences are reset.

An upper or lower bound used in an RESIZE statement must be exactly the same as the corresponding upper or lower bound defined for the array.
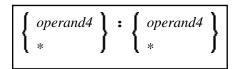
Example:

```
DEFINE DATA LOCAL
1 #a(I4/1:*)
1 #g(1:*)
  2 #ga(I4/1:*)

1 #i(i4)
END-DEFINE
...

*/ resizing #a (1:10)
RESIZE ARRAY #a TO (1:10)        /* #a is resized to
RESIZE ARRAY #a TO (*:10)        /* 10 occurrences.


/* resizing #ga (1:10,1:20)
RESIZE ARRAY #g TO (1:10)        /* 1st dimension is set to (1:10)
RESIZE ARRAY #ga TO (*:*,1:20)   /* 1st dimension is dependent and
                                 /* therefore kept with (*:*)
                                 /* 2nd dimension is set to (1:20)

RESIZE ARRAY #a TO (5:10)        /* This is rejected because the lower index
                                 /* must be 1 or *
RESIZE ARRAY #a TO (#i:10)       /* This is rejected because the lower index
                                 /* must be 1 or *

RESIZE ARRAY #ga TO (1:10,1:20)  /* (1:10) for the 1st dimension is rejected
                                 /* because the dimension is dependent and
                                 /* must be specified with (*:*).
```

For further information, see

- *Storage Management of X-Arrays*

- *Storage Management of X-Group Arrays*

Dimension

$$
\left\{ \begin{array}{c} operand4 \\ * \end{array} \right\} : \left\{ \begin{array}{c} operand4 \\ * \end{array} \right\}
$$

The lower and upper bound notation (*operand4* or asterisk) to which the X-array should be expanded is specified here. If the current value of the upper or lower bound should be used, an asterisk (*) may be specified in place of *operand4*. In place of * . *, you may also specify a single asterisk.

The number of dimensions (*dim*) must exactly match the defined number of dimensions of the X-array (1, 2, or 3).

If the number of occurrences for a specified dimension is less than the number of the currently allocated occurrences, the number of occurrences is not changed for the corresponding dimension.