

REDUCE

REDUCE { <i>dynamic-clause</i> } [GIVING <i>operand5</i>] <i>array-clause</i> }
--

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 | RESIZE

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

Function

The REDUCE statement is used to reduce:

- the allocated length of a dynamic variable (*dynamic-clause*), or
- the number of occurrences of X-arrays (*array-clause*).

For further information, see also the sections *Using Dynamic Variables, X-Arrays, Storage Management of X-Group Arrays* in the *Programming Guide*.

Syntax Description

Operand Definition Table:

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

Syntax Element Description:

Syntax Element	Description
<i>dynamic-clause</i>	<p>Dynamic Clause:</p> <p>The REDUCE DYNAMIC VARIABLE statement reduces the allocated length of a dynamic variable (<i>operand1</i>) to the length specified (<i>operand2</i>).</p> <p>For further information, see <i>Dynamic Clause</i> below.</p>
<i>operand1</i>	<p>Dynamic Variable:</p> <p><i>operand1</i> is the dynamic variable for which the length is to be reduced.</p>
<i>operand2</i>	<p>Target Length of Dynamic Variable:</p> <p><i>operand2</i> is used to specify the length to which the dynamic variable is to be reduced.</p> <p>The value specified must be a non-negative integer constant or a variable of type Integer4 (I4).</p>
<i>array-clause</i>	<p>Array Clause:</p> <p>The REDUCE ARRAY statement reduces the number of occurrences of the X-array (<i>operand3</i>) to the upper and lower bound specified with (<i>dim[,dim[,dim]]</i>).</p> <p>For further information, see <i>Array Clause</i> below.</p>
<i>operand3</i>	<p>X-Array:</p> <p><i>operand3</i> is the X-array. The occurrences of the X-array can be reduced.</p> <p>The index notation of the array is optional. As index notation only the complete range notation * is allowed for each dimension.</p>
<i>dim operand4</i>	<p>Dimension:</p> <p>The lower and upper bound notation (<i>operand4</i> or asterisk) to which the X-array should be reduced is specified here. If the current value of the upper or lower bound should be used, an asterisk (*) must be specified instead of <i>operand4</i>.</p> <p>For further information, see <i>Dimension</i> below.</p>
GIVING <i>operand5</i>	<p>GIVING Clause:</p> <p>If the GIVING clause is not specified, Natural runtime error processing is triggered if an error occurs.</p> <p>If the GIVING clause is specified, <i>operand5</i> contains the Natural message number if an error occurred, or zero upon success.</p>

Dynamic Clause

```
[SIZE OF] DYNAMIC [VARIABLE] operand 1 TO operand 2
```

The REDUCE DYNAMIC VARIABLE statement reduces the allocated length of a dynamic variable (*operand1*) to the length specified (*operand2*). The allocated memory of the dynamic variable which is beyond the given length is released immediately, i.e., when the statement is executed.

If the currently allocated length (*LENGTH) of the dynamic variable is greater than the given length, *LENGTH is set to the given length and the content of the variable is truncated (but not modified). If the given length is larger than the currently allocated length of the dynamic variable, the statement will be ignored.

Array Clause

```
[OCCURRENCES OF] ARRAY operand3 TO { 0  
                                  (dim[,dim[,dim]]) }
```

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

If REDUCE TO 0 (zero) is specified, all occurrences of the X-array are released. In other words, the whole array is reduced.

An upper or lower bound used in a REDUCE 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
...

*/ reducing #a (1:10)

REDUCE ARRAY #a TO (1:10)      /* #a is reduced
REDUCE ARRAY #a TO (*:10)     /* to 10 occurrences.

*/ reducing #ga (1:10,1:20)

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

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

```

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

REDUCE 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

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

The lower and upper bound notation (*operand4* or asterisk) to which the X-array should be reduced 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).

When using the REDUCE statement, it is only possible to decrease the number of occurrences. If the requested number is larger than the currently allocated number of occurrences, it will simply be ignored.