EXPAND

```
EXPAND ⎰ 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: REDUCE | RESIZE

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

Function

The EXPAND statement is used to expand:

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

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

For further information, see 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	U	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 EXPAND DYNAMIC VARIABLE statement expands the allocated length 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 size is to be expanded.
operand2	*operand2* is used to specify the length to which the dynamic variable is to be expanded. The value specified must be a non-negative integer constant or a variable of type Integer4 (I4).
array-clause	The EXPAND ARRAY statement increases 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 for which the number of occurrences may be increased. 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 (*) may 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 EXPAND DYNAMIC VARIABLE statement expands the allocated size of a dynamic variable (*operand1*) to the value specified with *operand2*.

If *operand2* is less than the currently allocated length of *operand1*, the statement will be ignored for this dynamic variable. The currently allocated length (*LENGTH) of the dynamic variable is not modified.

Array Clause

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

The EXPAND ARRAY statement increases the number of occurrences of the X-array (*operand3*) to the upper and lower bound specified with **TO** (*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.

When using the EXPAND statement, it is only possible to increase the number of occurrences. If the requested number is smaller than the currently allocated number of occurrences, it will simply be ignored.

An upper or lower bound used in an EXPAND 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
...
/* allocating #a(1:10)
EXPAND ARRAY #a TO (1:10)        /* #a is allocated 10
EXPAND ARRAY #a TO (*:10)        /* occurrences.


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

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

EXPAND 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 (*:*).
```
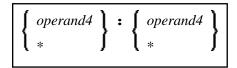
For further information, see

- *Storage Management of X-Arrays*

- *Storage Management of X-Group Arrays*

Dimension

Each of the dimensions (*dim*) specified in the *Array Clause* is defined using the following syntax:

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

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 place of *operand4*. Instead 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.