EXPAND

EXPAND { *dynamic-clause* | *array-clause* } [**GIVING** *operand5*]

Operand	Possible Structure					Possible Formats											Referencing Permitted	Dynamic Definition
Operand1		S	A			A					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

Related statements: REDUCE | RESIZE

Function

The EXPAND statement is used to expand:

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

Note:
The array clause is only available under Windows and UNIX.

dynamic-clause

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

The EXPAND DYNAMIC statement expands the size of the currently allocated storage of a dynamic variable (operand1) to the value specified in operand2.

When using the EXPAND statement it is only possible to increase the number of occurrences. If operand 2 is less than the size of the currently allocated storage of operand1, the statement will be ignored for this dynamic variable. The currently used size (*LENGTH) of the dynamic variable is not modified.

operand1

Operand1 is the dynamic variable for which the size is to be expanded.

operand2

Operand2 is used to specify the size 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

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

Note:
The clause is only valid under Windows and UNIX.

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]]).

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 as 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 #i(i4)
END-DEFINE
...
EXPAND ARRAY #a TO (1:10)   /* THIS IS ALLOWED
EXPAND ARRAY #a TO (*:10)   /* THIS IS ALLOWED
EXPAND ARRAY #a TO (5:10)   /* THIS IS REJECTED
EXPAND ARRAY #a TO (#i:10)  /* THIS IS REJECTED
```

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* | * } : { *operand4* | * }

The lower and upper bound notation (operand4 or asterisk) to which the X-array should be expanded is specified here. If the upper or lower bound must not be changed, an asterisk (*) must be specified instead of operand4.

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.

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.

