

Array object format.

0	(0)	-1												
4	(4)	0												
8	(8)	OL												
12	(C)	BA												
16	(10)	QUALIF												
20	(14)	LIND												
24	(18)	UIND												
28	(1C)	<table> <tr> <td>n</td><td>type</td><td>d4</td></tr> <tr> <td colspan="2"></td><td></td></tr> <tr> <td colspan="2">dn-1</td><td></td></tr> <tr> <td colspan="3">array elements</td></tr> </table>	n	type	d4				dn-1			array elements		
n	type	d4												
dn-1														
array elements														

- 1 in the first word indicates that this is an array object.
- OL is the array object length.
- BA is the address of the element A[0,0,...,0].
- QUALIF is the address of the prototype of the class qualifying the array, or unused if type is not ref (...).
- n number of subscripts.
- d dope vector
- LIND lower index
- UIND upper index
- see next page.

Dope vector and index checking.

Assume the array declaration

$A [l_1 : u_1, \dots, l_n : u_n];$

Then

$$d_1 = u_1 - l_1 + 1$$

$$d_i = d_{i-1}(u_i - l_i + 1), i = 2, \dots, n - 1$$

$$d_0 = 1 \text{ (not present in object)}$$

$$LIND = \sum_{i=1}^n l_i d_{i-1}$$

$$UIND = \sum_{i=1}^n u_i d_{i-1}$$

The computation of the address of $A[i_1, \dots, i_n]$ is described by the following algorithm:

$$t := \sum_{k=1}^n i_k d_{k-1}; \text{ if } t > UIND \text{ or } t < LIND \text{ then}$$

error ("subscriptbounds");

address := t * elementlength + BA;