

MEMELEMENT

The MEMELEMENT function returns a MEM block referring to a variable's memory, but not past it.

Syntax

```
memoryBlock = _MEMELEMENT(referenceVariable)
```

- The *referenceVariable* parameter designates the existing variable name using the memory block.
- MEMELEMENT is the same as MEM but in an array it returns the specifications of an element, not the entire array.
- All values created by memory functions MUST be freed using MEMFREE with a valid MEM variable type.
- The MEMELEMENT type contains the following **read-only** elements where *name* is the variable name:

name.**OFFSET** is the beginning offset of the memory block AS OFFSET
name.**SIZE** returns the largest available region of memory of the ELEMENT in bytes AS OFFSET
name.**ELEMENTSIZE** is the BYTE size of the elements within the block AS OFFSET

- 2 = INTEGER values have an element size of 2 bytes
- 4 = LONG integer and SINGLE float values have an element size of 4 bytes
- 8 = DOUBLE float and INTEGER64 values have an element size of 8 bytes
- 32 = FLOAT values have an element size of 32 bytes
- LEN = STRING or OFFSET byte sizes vary so use LEN(variable) for the number of bytes.

name.**TYPE** is the type (represented as bits combined to form a value) AS LONG (see below).

Contents

Syntax

.TYPE values (version 1.000 and up)

Versions prior to 1.000

Examples

See also

.TYPE values (version 1.000 and up)

- 🍷 0 = UDT (user defined type) or memory created by `_MEMNEW`
- 🍷 1 = 1 bit `ELEMENT.SIZE=1` *Only used along with specific types (currently integers or floats)
- 🍷 2 = 2 bit. `ELEMENT.SIZE=2` *
- 🍷 4 = 4 bit. `ELEMENT.SIZE=4` *
- 🍷 8 = 8 bit. `ELEMENT.SIZE=8` *
- 🍷 16 = 16 bit. `ELEMENT.SIZE=16` *
- 🍷 32 = 32 bit. `ELEMENT.SIZE=32` *
- 🍷 64 = 64 bit. `ELEMENT.SIZE=64` *
- 🍷 128 = 128 bit. `ELEMENT.SIZE=128` *
- 🍷 256 = 256 bit. `ELEMENT.SIZE=256` *
- 🍷 512(+ bit*) = integer types only(ie. whole numbers)
- 🍷 1024(+ bit*) = floating point types only(ie. numbers that can have a decimal point)
- 🍷 2048 = `STRING` type only
- 🍷 4096(+ 512 + bit*) = `_UNSIGNED` integer type only
- 🍷 8192 = `_MEM` type only
- 🍷 16384(+ 512 + bit*)= `_OFFSET` type only

Note: If a future integer, float or other type doesn't have a size that is 1,2,4,8,16,32,64,128 or 256 it won't have a size-bit set.

Versions prior to 1.000

- 🍷 1 = Integer types such as `_BYTE`, `INTEGER`, `LONG`, `_INTEGER64` or `_OFFSET`
- 🍷 2 = `_UNSIGNED` variable types. Value must be added to the variable type value.(2 cannot be used by itself)
- 🍷 3 = ALL `_UNSIGNED INTEGER` type values.(add 1 + 2)
- 🍷 4 = Floating point types such as `SINGLE`, `DOUBLE` or `_FLOAT`
- 🍷 8 = `STRING`
- 🍷 0 = unknown(eg. created with `_MEMNEW`) or user-defined-types

Note: `_MEM` and `_OFFSET` values cannot be cast to other variable types.

Examples

Example: Comparing the specifications returned by `_MEM` and `_MEMELEMENT` from an array.

```
DIM a(1 TO 100) AS _UNSIGNED _BYTE

DIM m1 AS _MEM
DIM m2 AS _MEMELEMENT

m1 = _MEM(a(50)) 'function returns information about array up to specific element
PRINT m1.OFFSET, m1.SIZE, m1.TYPE, m1.ELEMENTSIZE

m2 = _MEMELEMENT(a(50)) 'function returns information about the specific element
PRINT m2.OFFSET, m2.SIZE, m2.TYPE, m2.ELEMENTSIZE

END
```

Output using VERSION .954 ONLY .TYPE values: 1 (integer) + 2 (unsigned)

```
28377205      51      3      1
28377205      1      3      1
```

Explanation: `_MEM` returns the info about the array to that element while `_MEMELEMENT` returns info about that element only.

- 🔴 `_MEM` value returns the available array .SIZE as 51 bytes from the designated array element.
- 🔴 `_MEMELEMENT` value returns the available element .SIZE as one byte.

See also

- 🔴 `_MEM`
- 🔴 `_MEMNEW`
- 🔴 `_MEMGET`, `_MEMPUT`

Navigation:
Main Page with Articles and Tutorials
Keyword Reference - Alphabetical
Keyword Reference - By usage

Retrieved from "<https://qb64phoenix.com/qb64wiki/index.php?title=MEMELEMENT&oldid=7511>"

This page was last edited on 29 January 2023, at 10:39.