_MEMCOPY

The _MEMCOPY statement copies a block of bytes from one memory offset to another offset in memory.

Syntax

_MEMCOPY *sourceBlock*, *sourceBlock.OFFSET*, *sourceBlock.SIZE* TO *destBlock*, *destBlock.OFFSET*

Contents

- Syntax
- Parameters
- Description
- Examples
- See also

Parameters

- *sourceBlock* is the source memory block name created AS _MEM.
- *sourceBlock.OFFSET* is the dot _OFFSET within the source memory block to read from.
- *sourceBlock.SIZE* is the total number of bytes to transfer based on actual size.
- *destBlock* is the destination _MEM memory block name to transfer data to.
- *destBlock.OFFSET* is the dot _OFFSET within the dest _MEM memory block to write to.

Description

- The dot OFFSET is the memory block's start location in memory. Add bytes to place data further into the block.
- The dot SIZE is the total byte size of the memory block to transfer. You can transfer all or a portion of the data bytes.
- The memory block regions may overlap.
- **Always free memory blocks after values have been transferred to variables and are no longer required.**

Examples

Example: Swapping data from one STRING variable to another. Fixed length strings are recommended for speed.

```
DIM m AS _MEM
DIM n AS _MEM

m = _MEMNEW(10)
n = _MEMNEW(100)

_MEMPUT m, m.OFFSET, "1234567890"

s$ = SPACE$(10) 'to load into a variable length string set its length first
_MEMGET m, m.OFFSET, s$
PRINT "in:[" + s$ + "]"

_MEMCOPY m, m.OFFSET, m.SIZE TO n, n.OFFSET 'put m into n

b$ = SPACE$(10)
_MEMGET n, n.OFFSET, b$
PRINT "out:[" + b$ + "]"
_MEMFREE m: _MEMFREE n 'always clear the memory when done
```

Snippet: Instead of copying each array element, one at a time in nested FOR loops, _MEMCOPY does it in one statement instantly.

```
'copy array a to array b one index at a time:
FOR i1 = 0 TO 100
    FOR i2 = 0 TO 100
        b(i1, i2) = a(i1, i2)
    NEXT
NEXT

'copy array a to array b in memory instantly:
DIM ma AS _MEM: ma = _MEM(a()) 'place array data into blocks
DIM mb AS _MEM: mb = _MEM(b())
_MEMCOPY ma, ma.OFFSET, ma.SIZE TO mb, mb.OFFSET
_MEMFREE ma: _MEMFREE mb 'clear the memory when done
```

See also

- _MEM, _MEM (function)
- _MEMNEW, _MEMGET (function)
- _MEMIMAGE, _MEMELEMENT
- _MEMGET, _MEMPUT
- _MEMFILL, _MEMFREE

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