# _MEMIMAGE

The _MEMIMAGE function returns a _MEM value referring to an image's memory using a designated image handle.

## Syntax

*imageBlock* = _MEMIMAGE[(*imageHandle&*)]

## Parameters

- The *imageBlock* _MEM type variable holds the read-only elements .OFFSET, .SIZE, .TYPE and .ELEMENTSIZE.
- If the optional *imageHandle&* isn't passed, it is assumed to be the current _DESTination program screen image.

## Description

- Use the function to place images into memory access blocks for faster data access.
- All values created by this function must be freed using _MEMFREE with a valid _MEM type variable.
- Image handle values and the memory used must still be freed using _FREEIMAGE when no longer required.

## Examples

*Example 1:* Darkening an image using memory with $CHECKING:OFF for greater speed. Use any 24 bit image file name on the second code line.

```
SCREEN _NEWIMAGE(1024, 768, 32)
i& = _LOADIMAGE("turtle.jpg") '<<<<<<<<<<<<< use any 24 bit image file

FOR n! = 1 TO 0.01 STEP -0.01
    i2& = _COPYIMAGE(i&)
    DarkenImage i2&, n!
    _PUTIMAGE (0, 0), i2&
    _FREEIMAGE i2&
    _DISPLAY
NEXT

SUB DarkenImage (Image AS LONG, Value_From_0_To_1 AS SINGLE)
```

## Contents

```
IF Value_From_0_To_1 <= 0 OR Value_From_0_To_1 >= 1 OR _PIXELSIZE(Image) <> 4 THEN EXIT SUB
DIM Buffer AS _MEM: Buffer = _MEMIMAGE(Image) 'Get a memory reference to our image
DIM Frac_Value AS LONG: Frac_Value = Value_From_0_To_1 * 65536 'Used to avoid slow floating point calculations
DIM O AS _OFFSET, O_Last AS _OFFSET
O = Buffer.OFFSET 'We start at this offset
O_Last = Buffer.OFFSET + _WIDTH(Image) * _HEIGHT(Image) * 4 'We stop when we get to this offset
'use on error free code ONLY!
$CHECKING:OFF
DO
    _MEMPUT Buffer, O, _MEMGET(Buffer, O, _UNSIGNED _BYTE) * Frac_Value \ 65536 AS _UNSIGNED _BYTE
    _MEMPUT Buffer, O + 1, _MEMGET(Buffer, O + 1, _UNSIGNED _BYTE) * Frac_Value \ 65536 AS _UNSIGNED _BYTE
    _MEMPUT Buffer, O + 2, _MEMGET(Buffer, O + 2, _UNSIGNED _BYTE) * Frac_Value \ 65536 AS _UNSIGNED _BYTE
    O = O + 4
LOOP UNTIL O = O_Last
'turn checking back on when done!
$CHECKING:ON
_MEMFREE Buffer
END SUB
```

Code by Galleon

*Explanation:* The second value passed to DarkenImage is a value from 0.0 to 1.0 where 0.0 is full darkness and 1 is none.

*Example 2:* Reading information stored in an image with _MEMIMAGE to print ASC text characters to the screen.

```
SCREEN 13
_FULLSCREEN
PSET (0, 0), ASC("H")
PSET (1, 0), ASC("E")
PSET (2, 0), ASC("L")
PSET (3, 0), ASC("L")
PSET (4, 0), ASC("O")
PSET (5, 0), 32
PSET (6, 0), ASC("W")
PSET (7, 0), ASC("O")
PSET (8, 0), ASC("R")
PSET (9, 0), ASC("L")
PSET (10, 0), ASC("D")
DIM m AS _MEM
m = _MEMIMAGE
x1$ = _MEMGET(m, m.OFFSET, STRING * 11) 'convert numbers to ASCII text characters
_MEMFREE m 'free memory when done
LOCATE 10, 1: PRINT LEN(x1$) 'prints 11 as byte length
PRINT x1$ 'prints HELLO WORLD
END
```

*Notes:* The colors in the upper left corner are the text data used. An image could hold a hidden text message this way.

## See also

- _MEM
- _MEMNEW
- _MEMGET, _MEMPUT
- _MEMFREE
- $CHECKING

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