NAME

mbm − format of MBITMAP files and structures

SYNOPSIS

#include "mjsu.h"

DESCRIPTION

The MBITMAP structure is used to represent rectangular monochrome bitmaps with upto 65535 rows and 65535 columns.

In-Memory Structure

The **mbm_*** functions in the library utilise the following structure to represent bitmaps:

```
typedef struct
        {
        USHORT height;
        USHORT width;
        BYTE *bits;
        } MBITMAP;
```

height and *width* are in units of "dots", "pixels" or "bits". *bits* is a pointer to dynamically-allocated storage holding the raw bits of the bitmap.

The raw bits are stored consecutively, row-major, in consecutive bytes (arranged most-sigificant-bit first), with no per-row padding.

Thus the first column of the top row of the bitmap is stored in the most-siginificant-bit of *bits*[0], the second column of the top row is stored in the second-most-significant-bit of *bits*[0], and so on.

The number of bytes allocated for the raw bits is given by the formula:

(((*height* * *width*) + 7) / 8)

External Files

MBITMAP structures can be saved in or loaded from external files, using **mbm_save(3)** and **mbm_load(3)**. In such a file, each bitmap consists of (in order);

the four ASCII bytes "<MBM"
an ASCII digit specifying the file-format version
an ASCII carriage-return character
an ASCII linefeed character
8 bits of "flags", currently not used (should be zero)
height as a 16-bit unsigned integer
width as a 16-bit unsigned integer
the raw bits of the bitmap, ordered as above

height and *width* are stored "least-significant-byte-first" (ie: in 8086 byte-order).

By convention, files containing only MBITMAPs usually have the filename extension ".mbm", and are referred to as "M-B-M files". Such files can contain multiple bitmaps, one after the other.

mbm_save() and **mbm_load()** can be used to embed and extract MBITMAPs from arbitrary types of files.

VERSION

The external file-format described above is MBM version 1. Future versions may utilise compression techniques.

SEE ALSO

mbm_load(3), **mbm_save(3)**.

mbm_bits(3), et al.