**BITMAP Image File:**

**BITMAP File Format:**

* BITMAP File consist following information:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Structure Name | Mandatory | Size | Purpose | Comments |
| BITMAP File Header | Yes | 14 Bytes | Store the general information about the BITMAP Image File. | Not needed after the file is loaded in the memory. |
| BITMAP Info Header | Yes | Fixed-size | Store the detailed information about BITMAP Image and define the pixel format. | Immediately follows the BITMAP File Header. |
| Extra-bit Masks | No | 3 or 4 DWORDs | To define the pixel format. | Present only if the DIB header is the BITMAPINFOHEADER. |
| Color-Table | Semi-Mandatory | Variable-size | To define the color used by the BITMAP image data (PixelArray). | Mandatory for the color depths <= 8. |
| GAP1 | No | Variable-size | Structure alignment | An artifact of the File offset to PixelArray in the BITMAP File header. |
| Pixel Array | Yes | Variable-size | To define the actual values of the pixels | The pixel format is defined by the DIB header or Extra-bit Masks. Each row of the Pixel Array is padded to a multiple of 4 bytes in size. |

**Color Table:**

* The colors in the Color-Table are usually specified 4-byte per entry 8.8.8.0.8 format (RGBAX notation).

**Pixel Storage:**

* The bits representing the BITMAP pixels are packed in rows. The size of each row is rounded up to a multiple of 4-bytes (32-bit DWORDs) by padding.

For Images with height > 1, multiple padded rows are stored consicutively, forming PixelArray.

Total number of bytes needed to store a row of pixels can be calculated as:

***RowSize = (((BitPerPixel \* ImageWidth) + 31)/32) \* 4***

The total amount of bytes necessary to store an array of pixels in an 'n' bits per pixel (bpp) image with 2n colors, can be calculated by accounting of the effect of rounding up the size of each row to a multiple of a 4 bytes, as follows;

***PixelArraySize = RowSize \* |ImageHeight|***

The total BITMAP File size can be approximated as:

***FileSize = 54 + 4\*2bpp + PixelArraySize***

Only images with 8 or fewer bits per pixel must account for the palette. 16Bpp images (or higher), may omit the palette part from the size calculation, as follows:

***FileSize = 54 + PixelArraySize***

**BITMAPFILEHEADER structure:**

* WORD Type : Type = "BM" (mainly)

DWORD Size : Total size of the BITMAP image file = sizeof(FileHeader) + sizeof(InfoHeader) + sizeof(PixelArray).

WORD Reserved1 : Optional = "0" (mainly)

WORD Reserved2 : Optional = "0" (mainly)

DWORD OffBits : The offset, in bytes, from the beginning of the BITMAPFILEHEADER structure to the bitmap bits = sizeof(FileHeader) + sizeof(InfoHeader).

**BITMAPINFOHEADER structure:**

* DWORD Size : Size of the structure = sizeof(InfoHeader).

LONG Width : Width of the BITMAP image in pixel.

LONG Height : Height of the BITMAP image in pixel.

WORD Planes : The number of planes for the target device. This value must be set to 1.

WORD BitCount : The number of bit-per-pixels.

DWORD Compression : The type of compression for a compressed bottom- up bitmap (top-down DIBs cannot be compressed).

Normally, its "BI\_RGB" (An uncompressed format).

DWORD SizeImage : The size, in bytes, of the image. This may be set to zero for BI\_RGB bitmaps.

LONG XPelsPerMeter : The horizontal resolution, in pixels-per-meter, of the target device for the bitmap.

LONG YPelsPerMeter : The vertical resolution, in pixels-per-meter, of the target device for the bitmap.

DWORD ClrUsed : The number of color indexes in the color table that are actually used by the bitmap.

DWORD ClrImportant : The number of color indexes that are required for displaying the bitmap. If this value is zero, all colors are required.

***typedef unsigned short WORD;***

***typedef unsigned int DWORD;***

***typedef signed long LONG;***