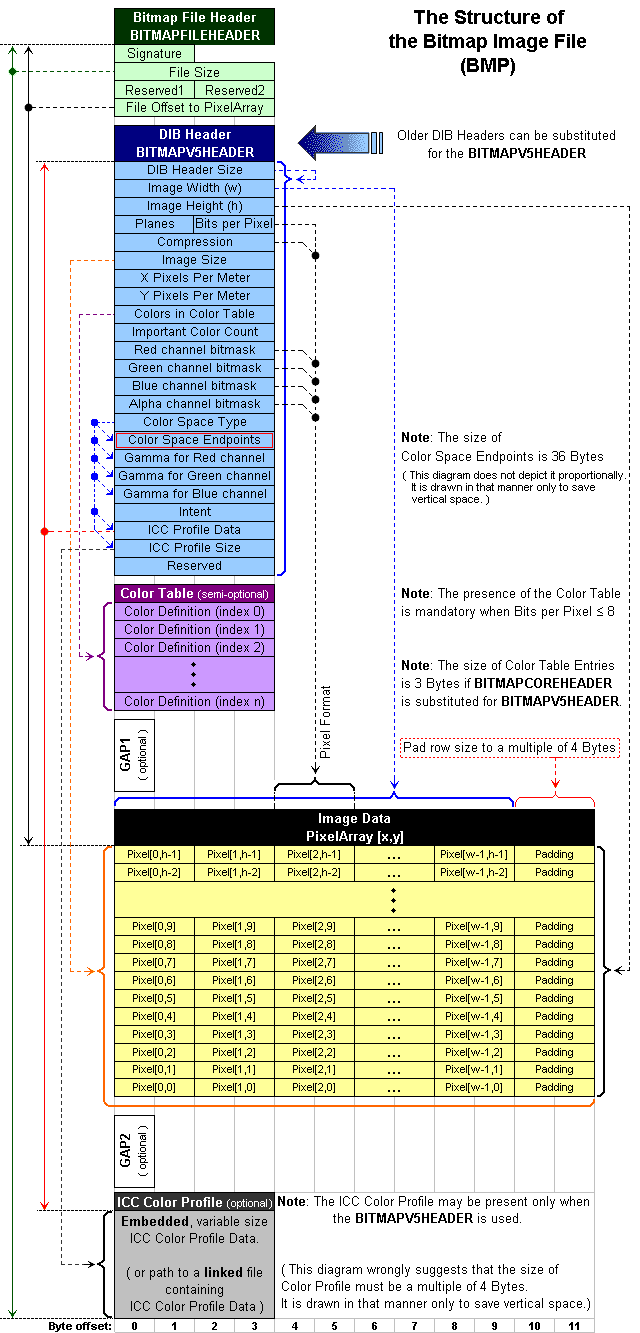
BMP:

Old file format, lots of variation

Structure:

Picture is worth 1000 words - see below

I try to summarize everything here, more detail is available on wikipedia.



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Structure Name** | **Optional** | **Size** | **Purpose** | **Comments** |
| **Bitmap File Header** | No | 14 Bytes | To store general information about the Bitmap Image File | Not needed after the file is loaded in memory |
| **DIB Header** | No | Fixed-size (however 7 different versions exist) | To store detailed information about the bitmap image and define the pixel format | Immediately follows the Bitmap File Header |
| **Extra bit masks** | Yes | 3 or 4 DWORDs[[2]](http://en.wikipedia.org/wiki/BMP_file_format#cite_note-AlphaBitFields-2) (12 or 16 Bytes) | To define the pixel format | Present only in case the DIB Header is the BITMAPINFOHEADER |
| **Color Table** | Semi-optional | Variable-size | To define colors used by the bitmap image data (Pixel Array) | Mandatory for [color depths](http://en.wikipedia.org/wiki/Color_depth) <= 8 |
| **Gap1** | Yes | Variable-size | Structure alignment | An artifact of the File Offset to PixelArray in the Bitmap File Header |
| **Pixel Array** | No | 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 in the Pixel Array is padded to a multiple of 4 bytes in size |
| **Gap2** | Yes | Variable-size | Structure alignment | An artifact of the ICC Profile Data offset field in the DIB Header |
| **ICC Color Profile** | Yes | Variable-size | To define the color profile for color management | Can also contain a path to an external file containing the color profile. When loaded in memory as "non-packed DIB", it is located between the color table and gap1.[[3]](http://en.wikipedia.org/wiki/BMP_file_format#cite_note-DIBHeaderTypes-3) |

Bitmap loaded into memory

* Becomes a DIB data structure. Almost the same as BMP format, but without the 14-byte bitmap file header

Bitmap File Header

* Identifies the file (file type, size, offset, etc) . Application reads this first to learn that it is a BMP file and that it isn’t damaged

DIB Header

* Gives the user application details about the image, which is then used by the application to display the image on the screen
* Several variations, all have a 32 bit field to specify the size

Color Table

* A block of bytes listing the colors used by the image, can be 2, 3, or 4 byte entries depending on
* Each pixel in the image is described by a set of bits that references a single color described by this table
* Not used when the pixels are in 16-bit per pixel (16bgpp) format

Pixel Storage

* Bits representing pixels are packed into rows. The size of each row is rounded up to the nearest multiple of 4 bytes and uses padding
* Formulas for calculating rowsize, number of rows, etc. can be found on wikipedia’s website

Pixel Array

* A block of DWORDs that describes the image pixel by pixel
* Typically stored starting in lower left corner, then left to right, row by row from bottom to top
* Paddding bytes must be appended to the end of the rows to bring each row up to a multiple of 4 bytes

Pixel Format

* Gets super complicated because there are so many options ranging from 2 bits per pixel to 32 bits per pixel. We will have to figure out how many bbp our test files use, then return to this to sort it out for our specific test files.