

## Bitmap file header

This block of bytes is at the start of the file and is used to identify the file. A typical application reads this block first to ensure that the file is actually a BMP file and that it is not damaged. The first 2 bytes of the BMP file format are the character "B" then the character "M" in ASCII encoding. All of the integer values are stored in little-endian format (i.e. least-significant byte first).

| Offset hex | Offset dec | Size    | Purpose  |
|------------|------------|---------|--|
| 00         | 0          | 2 bytes | <p>The <u>header field</u> used to identify the BMP and DIB file is 0x42 0x4D in hexadecimal, same as <code>BM</code> in ASCII. The following entries are possible:</p> <ul style="list-style-type: none"> <li>▪ <b>BM</b> – Windows 3.1x, 95, NT, ... etc.</li> <li>▪ <b>BA</b> – OS/2 struct bitmap array</li> <li>▪ <b>CI</b> – OS/2 struct color icon</li> <li>▪ <b>CP</b> – OS/2 const color pointer</li> <li>▪ <b>IC</b> – OS/2 struct icon</li> <li>▪ <b>PT</b> – OS/2 pointer</li> </ul> |
| 02         | 2          | 4 bytes | The size of the BMP file in bytes  |
| 06         | 6          | 2 bytes | Reserved; actual value depends on the application that creates the image   |
| 08         | 8          | 2 bytes | Reserved; actual value depends on the application that creates the image   |
| 0A         | 10         | 4 bytes | The offset, i.e. starting address, of the byte where the bitmap image data (pixel array) can be found.   |

## DIB header (bitmap information header)

This block of bytes tells the application detailed information about the image, which will be used to display the image on the screen. The block also matches the header used internally by Windows and OS/2 and has several different variants. All of them contain a dword (32-bit) field, specifying their size, so that an application can easily determine which header is used in the image. The reason that there are different headers is that Microsoft extended the DIB format several times. The new extended headers can be used with some GDI functions instead of the older ones, providing more functionality. Since the GDI supports a function for loading bitmap files, typical Windows applications use that functionality. One consequence of this is that for such applications, the BMP formats that they support match the formats supported by the Windows version being run. See the table below for more information.

## Windows and OS/2 bitmap headers

| Size | Header name                           | OS support  | Features   | Written by      |
|------|---------------------------------------|---|--|-----------------|
| 12   | BITMAPCOREHEADER<br>OS21XBITMAPHEADER | <u>Windows 2.0</u> or later<br><u>OS/2 1.x</u> <sup>[2]</sup>   |  |                 |
| 64   | OS22XBITMAPHEADER                     | <u>OS/2</u> BITMAPCOREHEADER2   | Adds halftoning.<br>Adds RLE and Huffman 1D compression. |                 |
| 16   | OS22XBITMAPHEADER                     | This variant of the previous header contains only the first 16 bytes and the remaining bytes are assumed to be zero values. <sup>[2]</sup><br><br>An example of such a case is the graphic <code>pal8os2v2-16.bmp</code> <sup>[9]</sup> of the BMP Suite. <sup>[10]</sup> |  |                 |
| 40   | BITMAPINFOHEADER                      | <u>Windows NT</u> , <u>3.1x</u> or later <sup>[1]</sup>   | Adds 16 bpp and 32 bpp formats.<br>Adds RLE compression. |                 |
| 52   | BITMAPV2INFOHEADER                    | Undocumented  | Adds RGB bit masks.                                      | Adobe Photoshop |
| 56   | BITMAPV3INFOHEADER                    | Not officially documented, but this documentation was posted on Adobe's forums, by an employee of Adobe with a statement that the standard was at one point in the past included in official MS documentation <sup>[11]</sup>   | Adds <u>alpha channel</u> bit mask.                      | Adobe Photoshop |
| 108  | BITMAPV4HEADER                        | <u>Windows NT 4.0</u> , <u>95</u> or later  | Adds color space type and <u>gamma correction</u>        |                 |
| 124  | BITMAPV5HEADER                        | <u>Windows NT 5.0</u> , <u>98</u> or later  | Adds <u>ICC color profiles</u>                           |                 |

| Offset (hex)   | Offset (dec) | Size (bytes) | OS/2 1.x BITMAPCOREHEADER <sup>[2]</sup>      |
|--|--------------|--------------|---|
| 0E   | 14           | 4            | The size of this header (12 bytes)            |
| 12   | 18           | 2            | The bitmap width in pixels (unsigned 16-bit)  |
| 14   | 20           | 2            | The bitmap height in pixels (unsigned 16-bit) |
| 16   | 22           | 2            | The number of color planes, must be 1         |
| 18   | 24           | 2            | The number of bits per pixel                  |
| OS/2 1.x bitmaps are uncompressed and cannot be 16 or 32 bpp |              |              |   |

Versions after BITMAPCOREHEADER only add fields to the end of the header of the previous version. For example: BITMAPV2INFOHEADER adds fields to BITMAPINFOHEADER, and BITMAPV3INFOHEADER adds fields to BITMAPV2INFOHEADER.

An integrated alpha channel has been introduced with the undocumented BITMAPV3INFOHEADER and with the documented BITMAPV4HEADER (since [Windows 95](#)) and is used within [Windows XP](#) logon and theme system as well as [Microsoft Office](#) (since v2000); it is supported by some [image editing software](#), such as [Adobe Photoshop](#) since version 7 and [Adobe Flash](#) since version MX 2004 (then known as [Macromedia Flash](#)). It is also supported by [GIMP](#), [Google Chrome](#), [Microsoft PowerPoint](#) and [Microsoft Word](#).

For compatibility reasons, most applications use the older DIB headers for saving files. With OS/2 no longer supported after [Windows 2000](#), for now the common Windows format is the BITMAPINFOHEADER header. See next table for its description. All values are stored as unsigned integers, unless explicitly noted.

| Offset (hex) | Offset (dec) | Size (bytes) | Windows BITMAPINFOHEADER <sup>[1]</sup>   |
|--------------|--------------|--------------|---|
| 0E           | 14           | 4            | the size of this header (40 bytes)  |
| 12           | 18           | 4            | the bitmap width in pixels (signed integer)   |
| 16           | 22           | 4            | the bitmap height in pixels (signed integer)  |
| 1A           | 26           | 2            | the number of color planes (must be 1)  |
| 1C           | 28           | 2            | the number of bits per pixel, which is the color depth of the image. Typical values are 1, 4, 8, 16, 24 and 32. |
| 1E           | 30           | 4            | the compression method being used. See the next table for a list of possible values                             |
| 22           | 34           | 4            | the image size. This is the size of the raw bitmap data; a dummy 0 can be given for BI_RGB bitmaps.             |
| 26           | 38           | 4            | the horizontal resolution of the image. (pixel per metre, signed integer)                                       |
| 2A           | 42           | 4            | the vertical resolution of the image. (pixel per metre, signed integer)   |
| 2E           | 46           | 4            | the number of colors in the color palette, or 0 to default to 2 <sup>n</sup>                                    |
| 32           | 50           | 4            | the number of important colors used, or 0 when every color is important; generally ignored                      |

# The Structure of the Bitmap Image File (BMP)



