

BMP 文件格式

1 简介

位图文件(Bitmap-File, BMP)格式是 Windows 采用的图像文件存储格式, 在 Windows 环境下运行的所有图像处理软件都支持这种格式。Windows 3.0 以前的 BMP 位图文件格式与显示设备有关, 因此把它称为设备相关位图(device-dependent bitmap, DDB)文件格式。Windows 3.0 以后的 BMP 位图文件格式与显示设备无关, 因此把这种 BMP 位图文件格式称为设备无关位图(device-independent bitmap, DIB)格式, 目的是为了让 Windows 能够在任何类型的显示设备上显示 BMP 位图文件。BMP 位图文件默认的文件扩展名是 BMP 或者 bmp。

2 文件结构

位图文件可看成由 4 个部分组成: 位图文件头(bitmap-file header)、位图信息头(bitmap-information header)、彩色表(color table)和定义位图的字节阵列, 它们的名称和符号如表 6-01 所示。

表 6-01 BMP 图像文件组成部分的名称和符号

位图文件的组成	结构名称	符号
位图文件头(bitmap-file header)	BITMAPFILEHEADER	bmfh
位图信息头(bitmap-information header)	BITMAPINFOHEADER	bmih
彩色表(color table)	RGBQUAD	aColors[]
图像数据阵列字节	BYTE	aBitmapBits[]

位图文件结构可综合在表 6-02 中。

表 6-02 位图文件结构内容摘要

	偏移量	域的名称	大小	内容
图像文件头	0000h	标识符 (Identifier)	2 bytes	两字节的内容用来识别位图的类型: ‘BM’ : Windows 3.1x, 95, NT, ... ‘BA’ : OS/2 Bitmap Array ‘CI’ : OS/2 Color Icon ‘CP’ : OS/2 Color Pointer ‘IC’ : OS/2 Icon ‘PT’ : OS/2 Pointer

	0002h	File Size	1 dword	用字节表示的整个文件的大小
	0006h	Reserved	1 dword	保留，设置为 0
	000Ah	Bitmap Data Offset	1 dword	从文件开始到位图数据开始之间的数据 (bitmap data) 之间的偏移量
	000Eh	Bitmap Header Size	1 dword	位图信息头 (Bitmap Info Header) 的长度，用来描述位图的颜色、压缩方法等。下面的长度表示： 28h - Windows 3.1x, 95, NT, ... 0Ch - OS/2 1.x F0h - OS/2 2.x
	0012h	Width	1 dword	位图的宽度，以像素为单位
	0016h	Height	1 dword	位图的高度，以像素为单位
	001Ah	Planes	1 word	位图的位面数
图 像 信 息 头	001Ch	Bits Per Pixel	1 word	每个像素的位数 1 - Monochrome bitmap 4 - 16 color bitmap 8 - 256 color bitmap 16 - 16bit (high color) bitmap 24 - 24bit (true color) bitmap 32 - 32bit (true color) bitmap
	001Eh	Compression	1 dword	压缩说明： 0 - none (也使用 BI_RGB 表示) 1 - RLE 8-bit / pixel (也使用 BI_RLE4 表示)

				2 - RLE 4-bit / pixel (也使用 BI_RLE8 表示) 3 - Bitfields (也使用 BI_BITFIELDS 表示)
	0022h	Bitmap Data Size	1 dword	用字节数表示的位图数据的大小。该数必须是 4 的倍数
	0026h	HResolution	1 dword	用像素/米表示的水平分辨率
	002Ah	VResolution	1 dword	用像素/米表示的垂直分辨率
	002Eh	Colors	1 dword	位图使用的颜色数。如 8-位/像素表示为 100h 或者 256.
	0032h	Important Colors	1 dword	指定重要的颜色数。当该域的值等于颜色数时, 表示所有颜色都一样重要
调色板数据	0036h	Palette	N * 4 byte	调色板规范。对于调色板中的每个表项, 这 4 个字节用下述方法来描述 RGB 的值: <ul style="list-style-type: none"> • 1 字节用于蓝色分量 • 1 字节用于绿色分量 • 1 字节用于红色分量 • 1 字节用于填充符(设置为 0)
图像数据	0436h	Bitmap Data	x bytes	该域的大小取决于压缩方法, 它包含所有的位图数据字节, 这些数据实际就是彩色调色板的索引号

3 构件详解

1. 位图文件头

位图文件头包含有关于文件类型、文件大小、存放位置等信息, 在 Windows 3.0 以上版本的位图文件中用 BITMAPFILEHEADER 结构来定义:

```
typedef struct tagBITMAPFILEHEADER { /* bmfh */
```

```
    UINT bfType;
```

```
    DWORD bfSize;
```

```

UINT bfReserved1;

UINT bfReserved2;

DWORD bfOffBits;

} BITMAPFILEHEADER;

```

其中:

bfType	说明文件的类型.
bfSize	说明文件的大小, 用字节为单位
bfReserved1	保留, 设置为 0
bfReserved2	保留, 设置为 0
bfOffBits	说明从 BITMAPFILEHEADER 结构开始到实际的图像数据之间的 字 节偏移量

2. 位图信息头

位图信息用 BITMAPINFO 结构来定义, 它由位图信息头(bitmap-information header)和彩色表(color table)组成, 前者用 BITMAPINFOHEADER 结构定义, 后者用 RGBQUAD 结构定义。BITMAPINFO 结构具有如下形式:

```

typedef struct tagBITMAPINFO { /* bmi */

    BITMAPINFOHEADER bmiHeader;

    RGBQUAD bmiColors[1];

} BITMAPINFO;

```

其中:

bmiHeader	说明 BITMAPINFOHEADER 结构
bmiColors	说明彩色表 RGBQUAD 结构的阵列

BITMAPINFOHEADER 结构包含有位图文件的大小、压缩类型和颜色格式, 其结构定义为:

```

typedef struct tagBITMAPINFOHEADER { /* bmih */

    DWORD biSize;

    LONG biWidth;

    LONG biHeight;

```

```

WORD biPlanes;

WORD biBitCount;

DWORD biCompression;

DWORD biSizeImage;

LONG biXPelsPerMeter;

LONG biYPelsPerMeter;

DWORD biClrUsed;

DWORD biClrImportant;

} BITMAPINFOHEADER;

```

其中：

biSize	说明 BITMAPINFOHEADER 结构所需要的字节数
biWidth	说明图像的宽度，以像素为单位
biHeight	说明图像的高度，以像素为单位
biPlanes	为目标设备说明位面数，其值设置为 1
biBitCount	说明位数/像素，其值为 1、2、4 或者 24
biCompression	<ul style="list-style-type: none"> 说明图像数据压缩的类型。其值可以是下述值之一： BI_RGB：没有压缩； BI_RLE8：每个像素 8 位的 RLE 压缩编码，压缩格式由 2 字节组成(重复像素计数和颜色索引)； BI_RLE4：每个像素 4 位的 RLE 压缩编码，压缩格式由 2 字节组成
biSizeImage	说明图像的大小，以字节为单位。当用 BI_RGB 格式时，可设置为 0
biXPelsPerMeter	说明水平分辨率，用像素/米表示
biYPelsPerMeter	说明垂直分辨率，用像素/米表示
biClrUsed	说明位图实际使用的彩色表中的颜色索引数
biClrImportant	说明对图像显示有重要影响的颜色索引的数目，如果是 0，表示都重要。

现就 BITMAPINFOHEADER 结构作如下说明：

(1) 彩色表的定位

应用程序可使用存储在 biSize 成员中的信息来查找在 BITMAPINFO 结构中的彩色表，如下所示：

```
pColor = ((LPSTR) pBitmapInfo + (WORD) (pBitmapInfo->bmiHeader.biSize))
```

(2) biBitCount

biBitCount=1 表示位图最多有两种颜色，黑色和白色。图像数据阵列中的每一位表示一个像素。

biBitCount=4 表示位图最多有 16 种颜色。每个像素用 4 位表示，并用这 4 位作为彩色表的表项来查找该像素的颜色。例如，如果位图中的第一个字节为 0x1F，它表示有两个像素，第一像素的颜色就在彩色表的第 2 表项中查找，而第二个像素的颜色就在彩色表的第 16 表项中查找。

biBitCount=8 表示位图最多有 256 种颜色。每个像素用 8 位表示，并用这 8 位作为彩色表的表项来查找该像素的颜色。例如，如果位图中的第一个字节为 0x1F，这个像素的颜色就在彩色表的第 32 表项中查找。

biBitCount=24 表示位图最多有 $2^{24}=16\ 777\ 216$ 种颜色。bmiColors (或者bmciColors)成员就为NULL。每 3 个字节代表一个像素，其颜色有R、G、B字节的相对强度决定。

(3) ClrUsed

BITMAPINFOHEADER 结构中的成员 ClrUsed 指定实际使用的颜色数目。如果 ClrUsed 设置成 0，位图使用的颜色数目就等于 biBitCount 成员中的数目。

(4) 图像数据压缩

① **BI_RLE8**：每个像素为 8 位的 RLE 压缩编码，可使用编码方式和绝对方式中的任何一种进行压缩，这两种方式可在同一幅图中的任何地方使用。

编码方式：由 2 个字节组成，第一个字节指定使用相同颜色的像素数目，第二个字节指定使用的颜色索引。此外，这个字节对中的第一个字节可设置为 0，联合使用第二个字节的值表示：

- 第二个字节的值为 0：行的结束。
- 第二个字节的值为 1：图像结束。
- 第二个字节的值为 2：其后的两个字节表示下一个像素从当前开始的水平和垂直位置的偏移量。

绝对方式：第一个字节设置为 0，而第二个字节设置为 0x03~0xFF 之间的一个值。在这种方式中，第二个字节表示跟在这个字节后面的字节数，每个字节包含单个像素的颜色索引。压缩数据格式需要字边界(word boundary)对齐。

[例 6.1] 用十六进制表示的 8 位压缩图像数据如下：

03 04 05 06 00 03 45 56 67 00 02 78 00 02 05 01 02 78 00 00 09 1E 00 01

这些压缩数据可解释为：

压缩数据	扩展数据
03 04	04 04 04
05 06	06 06 06 06 06
00 03 45 56 67 00	45 56 67
02 78	78 78
00 02 05 01	从当前位置右移 5 个位置后向下移一

	行
02 78	78 78
00 00	行结束
09 1E	1E 1E 1E 1E 1E 1E 1E 1E 1E
00 01	RLE 编码图像结束

② **BI_RLE4**: 每个像素为 4 位的 RLE 压缩编码, 同样也可使用编码方式和绝对方式中的任何一种进行压缩, 这两种方式也可在同一幅图中的任何地方使用。这两种方式是:

编码方式: 由 2 个字节组成, 第一个字节指定像素数目, 第二个字节包含两种颜色索引, 一个在高 4 位, 另一个在低 4 位。第一个像素使用高 4 位的颜色索引, 第二个使用低 4 位的颜色索引, 第 3 个使用高 4 位的颜色索引, 依此类推。

绝对方式: 这个字节对中的第一个字节设置为 0, 第二个字节包含有颜色索引数, 其后续字节包含有颜色索引, 颜色索引存放在该字节的高、低 4 位中, 一个颜色索引对应一个像素。此外, BI_RLE4 也同样联合使用第二个字节中的值表示:

- 第二个字节的值为 0: 行的结束。
- 第二个字节的值为 1: 图像结束。
- 第二个字节的值为 2: 其后的两个字节表示下一个像素从当前开始的水平和垂直位置的偏移量。

[例 6.2] 用十六进制数表示的 4 位压缩图像数据:

03 04 05 06 00 06 45 56 67 00 04 78 00 02 05 01 04 78 00 00 09 1E 00 01

这些压缩数据可解释为 :

压缩数据	扩展数据
03 04	0 4 0
05 06	0 6 0 6 0
00 06 45 56 67 00	4 5 5 6 6 7
04 78	7 8 7 8
00 02 05 01	从当前位置右移 5 个位置后向下移一行
04 78	7 8 7 8
00 00	行结束
09 1E	1 E 1 E 1 E 1 E 1
00 01	RLE 图像结束

3. 彩色表

彩色表包含的元素与位图所具有的颜色数相同，像素的颜色用 RGBQUAD 结构来定义。对于 24-位真彩色图像就不使用彩色表，因为位图中的 RGB 值就代表了每个像素的颜色。彩色表中的颜色按颜色的重要性排序，这可以辅助显示驱动程序为不能显示足够多颜色数的显示设备显示彩色图像。RGBQUAD 结构描述由 R、G、B 相对强度组成的颜色，定义如下：

```
typedef struct tagRGBQUAD { /* rgbq */  
  
    BYTE rgbBlue;  
  
    BYTE rgbGreen;  
  
    BYTE rgbRed;  
  
    BYTE rgbReserved;  
  
} RGBQUAD;
```

其中：

rgbBlue	指定蓝色强度
rgbGreen	指定绿色强度
rgbRed	指定红色强度
rgbReserved	保留，设置为 0

4. 位图数据

紧跟在彩色表之后的是图像数据字节阵列。图像的每一扫描行由表示图像像素的连续的字节组成，每一行的字节数取决于图像的颜色数目和用像素表示的图像宽度。扫描行是由底向上存储的，这就是说，阵列中的第一个字节表示位图左下角的像素，而最后一个字节表示位图右上角的像素。