

Графический файл формат BMP

Файлы BMP

Формат хранения растрового (bitmap) изображения разработан и поддерживается компанией Microsoft и используется в операционной системе Windows и создаваемых под неё приложениях.

Файл формата "BMP" имеет следующую достаточно простую структуру, содержащую в своей основе используемый Windows в формат DIB (Device Independent Bitmap) с разделенной информацией о цвете и изображении.

Файл BMP состоит из следующих четырех частей:

- ☐ Заголовок растрового файла (BITMAPFILEHEADER)
- ☐ Заголовок растровой информации (BITMAPINFOHEADER)
- ☐ Палитра т.е. таблица цветов (RGBQUAD [])
- ☐ Массив пикселей (Pixels [] [])

Заголовок растрового файла BMP

- Структура **BITMAPFILEHEADER** содержит сведения о типе, размере и макете файла, содержащего DIB:

```
typedef struct tagBITMAPFILEHEADER {  
    WORD bfType;           // Тип файла растрового изображения  
                           // должен быть BM (2 байта)  
    DWORD bfSize;          // Размер файла растрового изображения  
                           // в байтах (4 байта)  
    WORD bfReserved1;      // Зарезервированное слово растрового  
                           // файла, должно быть 0 (2 байта)  
    WORD bfReserved2;      // Зарезервированное слово растрового  
                           // файла, должно быть 0 (2 байта)  
    DWORD bfOffBits;       // Начальная позиция данных растрового  
изображения относительно растрового изображения (4 байта)  
} BITMAPFILEHEADER;
```

Заголовок файла растрового изображения имеет в общей сложности **14 байт**.

Заголовок растровой информации

```
typedef struct tagBITMAPINFOHEADER {  
    DWORD biSize;    // Количество байтов, занимаемых структурой (4 байта)  
    LONG biWidth;    // Ширина изображения в пикселях (4 байта)  
    LONG biHeight;   // Высота изображения в пикселях (4 байта)  
    WORD biPlanes;   // Число цветовых плоскостей 1 (2 байта)  
    WORD biBitCount; // Глубина цвета, бит на пиксел:1 (2 цвета),  
                    // 4 (16 цветов), 8 (256 цветов), 24 (истинный цвет)  
                    // или 32 (улучшенный истинный цвет) (2 байта)  
    DWORD biCompression; // Тип сжатия растрового изображения должен  
                        // быть 0 (без сжатия), (4 байта)  
    DWORD biSizeImage;    // Размер изображения в байтах (4 байта)  
    LONG biXPelsPerMeter; // Разрешение изображения по горизонтали,  
                        // количество пикселей на метр (4 байта)  
    LONG biYPelsPerMeter; // Вертикальное разрешение растрового  
    изображения, пикселей на метр (4 байта)  
    DWORD biClrUsed; // Количество цветов в таблице цветов, фактически  
    используемых растровым изображением (4 байта)  
    DWORD biClrImportant; // Количество важных цветов в процессе  
    отображения растрового изображения (4 байта)  
} BITMAPINFOHEADER;
```

Общий заголовок информации о растровом изображении **40 байт.**

Палитра т.е. таблица цветов

В таблице цветов длина изменяется от **8** до **1024** байт

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

Особенности таблицы цветов. В структуре **BITMAPINFOHEADER** значение поля **biClrUsed** определяет, будут ли использоваться растровым изображением индексированные цвета.

Таблица цветов обязательно если изображение состоит из 1-, 4-, 8-битных пикселей, при большей битности она не обязательна.

Обращение к данным массива пикселей

Первый пиксель последней строки изображения **BMP** представлен первым байтом данных пикселей. Затем сканирование продвигается вперед, сопоставляя следующий пиксель со следующим байтом в данных пикселей до последнего пикселя в этой строке.

Строка сканирования аналогична ряду пикселей. Поскольку BMP следует сканированию снизу вверх, первая строка сканирования является последней строкой изображения BMP. После того, как строка сканирования построила пиксель для этой строки, она перемещается вверх на одну строку и начинает разрешать пиксели из данных пикселей для этой строки.

Массив пикселей (Pixels [y][x])

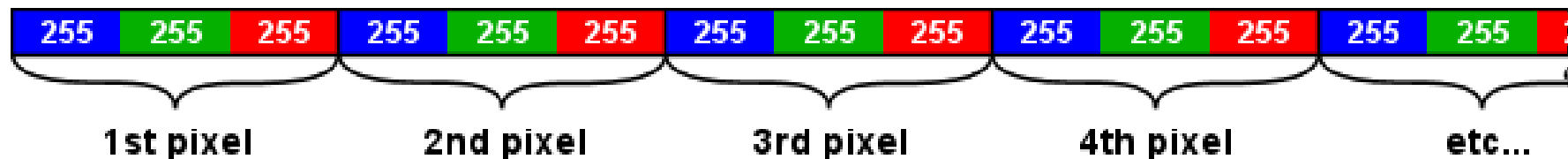
Pixel[h-1][0]	Pixel[h-1][1]	Pixel[h-1][2]	...	Pixel[h-1][w-1]	Padding
Pixel[h-2][0]	Pixel[h-2][1]	Pixel[h-2][2]	...	Pixel[h-2][w-1]	Padding
•					
•					
•					
Pixel[1][0]	Pixel[1][1]	Pixel[1][2]	...	Pixel[1][w-1]	Padding
Pixel[0][0]	Pixel[0][1]	Pixel[0][2]	...	Pixel[0][w-1]	Padding

Padding в строках BMP данных

В реальном мире пиксель может занимать от нескольких бит до нескольких байтов, чтобы определить свой цвет. Это зависит исключительно от значения **BitsPerPixel**. Но в любом случае каждый пиксель BMP определяется одинаковым размером.

В случае 24-битных изображений используется структура:

```
typedef struct tagRGBTRIPLE {  
    BYTE    rgbtBlue;  
    BYTE    rgbtGreen;  
    BYTE    rgbtRed;  
} RGBTRIPLE
```



Для согласованности и простоты каждая строка сканирования дополняется **0** до ближайшей 4-байтовой границы. Это означает, что когда BMP сканирует строку изображения, он рассматривает блок пикселей, который делится на 4 байта.