

STRUCTURE D'UN FICHIER BITMAP

Objectifs :

Ce document décrit la façon dont est enregistré un fichier BitMap (*.BMP) au format Windows ou OS/2.

Sommaire :

Le header
La zone BitMapInfo
La Palette
L'Image
Remerciements

I. LE HEADER :

Il indique les caractéristiques principales du **fichier** : son type (BitMap), sa taille, et l'emplacement de l'image dans le fichier.

Postition	Type	Description
00h	DB "BM"	Signature : BM indique que l'on se trouve en présence d'un fichier bitmap
02h	DD ?	Taille du fichier en octets
06h	DW 0	Réservé
08h	DW 0	Réservé
0Ah	DD ?	Offset de l'image

[*Retour au début*](#)

II. LA ZONE BITMAP INFO :

Elle définit les caractéristiques essentielles de **l'image**, c'est pourquoi on la distingue du header, mais elle suit directement celui-ci.

Position	Type	Description
0Eh	DD ?	Taille de la zone BitmapInfo
12h	DD ?	Largeur de l'image en pixels
16h	DD ?	Hauteur de l'image en pixels
1Ah	DW 1	Nombre de plans (1)
1Ch	DW 1 4 8 24	Bits par pixel (1=monochrome, 4=16 couleurs, 8=256 couleurs, 24=16 M de couleurs)
1Eh	DD ?	Format de compression de l'image : <ul style="list-style-type: none"> • 0 : non compressée • 1 : compressée à 8 bits par pixel (RLE8) • 2 : compressée à 4 bits par pixel (RLE4)
22h	DD ?	Taille de l'image en octets
26h	DD ?	Résolution horizontale en pixels par mètre
2Ah	DD ?	Résolution verticale en pixels par mètre
2Eh	DD ?	Nombre de couleurs utilisées. 0 = maximum possible. Si l'image est 24 bits, ce champ spécifie la taille de la palette de couleurs de référence utilisée pour optimiser les palettes Windows
32h	DD ?	Nombre d'index de couleurs importants pour l'affichage. 0 <=> toutes les couleurs importantes

Commentaires :

- ATTENTION : la **largeur** X doit être paire. Si la résolution est par exemple de 179 avec un mode 8 bits, alors chaque ligne aura une taille de 180 octets, avec un octet vide à chaque fin de ligne pour combler le vide. Cela peut paraître handicapant, mais ça vous permet en fait d'accéder au fichier mot par mot, et non plus octet par octet.
- La **résolution** en pixels par mètre peut paraître surprenante, mais en fait elle sert pour l'impression.
- Formats de **compression** :

La compression est du type RLE, c'est-à-dire Run Length Encoding. Le principe consiste à regrouper des suites de points de même couleur en un seul point en on définit combien de fois il doit être répété (on écrit "05 FE" au lieu de "FE FE FE FE FE" : ça prend quand même moins de place).

- Format 8 bits (RLE-4):

L'image est compressée selon deux modes, qui peuvent se trouver tous les deux dans la même image, c'est-à-dire que certaines portions sont compressées selon le mode encodé, et d'autres selon le mode absolu (qui équivaut à une compression zéro). Mais étudions le mode de compression :

Mode encodé : le format consiste à compresser les données en blocs de deux octets. Le premier octet définit le nombre de pixels à afficher et le second leur couleur. Si le premier octet vaut 0, c'est un code "escape" et il faut lire le second pour en traduire la signification :

- 2e octet = 0 : Fin de ligne

- 2e octet = 1 : Fin du bitmap
- 2e octet = 2 : Delta : les deux octets suivants désignent les positions horizontale et verticale du point suivant, relativement au point courant (valeur non signée)
- 2e octet = 03h à 0FFh : passage en mode absolu (c.a.d non compressé)

Mode absolu : c'est un simple retour au format non compressé. Le second octet (celui entre 03 et FF) indique le nombre de pixels à afficher normalement, c'est-à-dire sans décompression.

- En format 4 bits (RLE-4), c'est la même chose, sauf que la couleur est codée différemment : le second octet est séparé en deux valeurs de 4 bits : la partie base est la couleur #1 et la partie haute la couleur #2. On dessine ensuite au lieu d'une couleur deux couleurs tour à tour (en commençant par la #1). Cela vaut pour les deux modes, encodé et absolu. Apparemment, cela est surtout utilisé pour faire des images avec du tramage.

Quoi qu'il en soit, avec l'avènement des modes haute-résolution et des mémoires de masse à grande capacité, cette compression est devenue totalement obsolète puisqu'elle ne s'applique qu'au format 8 bits, et la plupart des logiciels de dessin, même s'ils gèrent cette fonctionnalité en option, ne l'utilisent pas. Autre inconvénient : cette méthode n'est efficace que pour les images de synthèse (créées artificiellement), car elles contiennent beaucoup de pixels de même couleurs, alors que les images naturelles, comme les photos, n'en contiennent quasiment pas, ce qui peut amener alors à une augmentation de la taille du fichier. Vous n'êtes donc pas du tout obligé de gérer la compression RLE.

[*Retour au début*](#)

III. LA PALETTE :

C'est un tableau contenant la liste des couleurs pour le mode 8 bits. Il suit immédiatement la zone Bitmap Info, et n'est présent qu'en mode 8 bits. Une couleur est codée sur 4 octets : R:G:B:Réservé. La taille de la palette est donc de 1024 octets.

[*Retour au début*](#)

IV. L'IMAGE :

Enfin la voilà ! Elle suit normalement directement la zone BitMap Info (ou la palette, s'il y en a une), mais il est préférable d'y accéder via l'offset défini dans le header. Elle est stockée normalement, suivant le format défini dans 1Ch et 1Eh (en 24 bits, c'est R:G:B:X) mais **attention** : l'image est stockée de à l'envers : il faut la lire en BGR et le premier pixel dans le fichier correspond au la dernier dans l'image ! Prenez-le en compte !

V. REMERCIEMENTS :

Document écrit par Antoche – Antoche@altern.org - #ICQ : 38 08 99 43

Libre reproduction et diffusion autorisée - modifications interdites sans autorisation de l'auteur.

[[Retour au début](#)] - [Retour à la page [graphisme](#)] - [Retour à la [page d'accueil](#)]
