

# Format BMP

## Table des matières

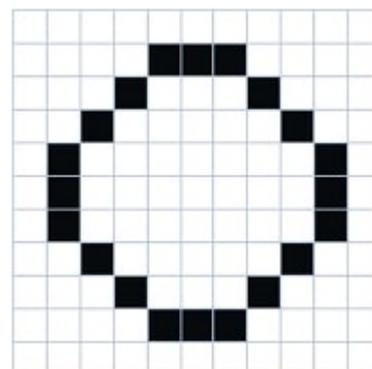
1. Introduction.....	2
2. Entête du fichier.....	2
3. Entête de l'image.....	3
4. Palette de l'image.....	3
5. Codage de l'image.....	4
6. Résolution.....	5
7. Définition.....	5
8. Taille du fichier.....	6
9. RLE.....	6

Bitmap, connu aussi sous son abréviation BMP, est un format d'image matricielle ouvert développé par Microsoft et IBM. C'est un des formats d'images les plus simples à développer et à utiliser pour programmer. Il est lisible par quasiment tous les visualiseurs et éditeurs d'images.



## 1. Introduction

Le format BMP est un des formats les plus simples développé conjointement par Microsoft et IBM, ce qui explique qu'il soit particulièrement répandu sur les plates formes Windows et OS/2<sup>1</sup>. Un fichier .BMP est un fichier bitmap, c'est-à-dire un fichier d'image graphique stockant les pixels<sup>2</sup> sous forme de tableau de points et gérant les couleurs soit en couleur vraie soit grâce à une palette indexée. Le format BMP a été étudié de telle manière à obtenir un bitmap indépendant du périphérique d'affichage (DIB, Device independent bitmap).



La structure d'un fichier bitmap est la suivante :

- En-tête du fichier (en anglais file header)
- En-tête du bitmap (en anglais bitmap information header, appelé aussi information Header)
- Palette (optionnellement)
- Corps de l'image

## 2. Entête du fichier

L'entête du fichier fournit des informations sur le type de fichier (Bitmap), sa taille et indique où commencent les informations concernant l'image à proprement parler.

L'entête du fichier est composé de quatre champs :

1. La signature (sur 2 octets), indiquant qu'il s'agit d'un fichier BMP à l'aide des deux caractères :
  - BM indique qu'il s'agit d'un Bitmap Windows.
  - BA indique qu'il s'agit d'un Bitmap OS/2.
  - CI indique qu'il s'agit d'une icône couleur OS/2.
  - CP indique qu'il s'agit d'un pointeur de couleur OS/2.
  - IC indique qu'il s'agit d'une icône OS/2.
  - PT indique qu'il s'agit d'un pointeur OS/2.
2. La taille totale du fichier en octets (codée sur 4 octets)
3. Un champ réservé (sur 4 octets)
4. L'offset<sup>3</sup> de l'image (sur 4 octets) : l'adresse relative du début des informations concernant l'image par rapport au début du fichier

1 Operating System/2

2 picture element

3 décalage

### 3. Entête de l'image

L'entête de l'image fournit des informations sur l'image, notamment ses dimensions et ses couleurs.

L'entête de l'image est composé de plusieurs champs :

1. La signature de l'entête de l'image en octets (codée sur 4 octets). Les valeurs hexadécimales suivantes sont possibles suivant le type de format BMP :
  - 0x28 pour Windows 3.1x, 95, NT, ...
  - 0x0C pour OS/2 1.x
  - 0xF0 pour OS/2 2.x
2. La largeur de l'image (sur 4 octets), c'est-à-dire le nombre de pixels horizontalement (en anglais width)
3. La hauteur de l'image (sur 4 octets), c'est-à-dire le nombre de pixels verticalement (en anglais height)
4. Le nombre de plans (sur 2 octets). Cette valeur vaut toujours 1
5. La profondeur de codage de la couleur (sur 2 octets), c'est-à-dire le nombre de bits utilisés pour coder la couleur. Cette valeur peut-être égale à 1, 4, 8, 16, 24 ou 32
6. La méthode de compression (sur 4 octets). Cette valeur vaut 0 lorsque l'image n'est pas compressée, ou bien 1, 2 ou 3 suivant le type de compression utilisé :
  - 1 pour un codage RLE de 8 bits par pixel
  - 2 pour un codage RLE de 4 bits par pixel
  - 3 pour un codage bitfields, signifiant que la couleur est codé par un triple masque représenté par la palette
7. La taille totale de l'image en octets (sur 4 octets).
8. La résolution horizontale (sur 4 octets)
9. La résolution verticale (sur 4 octets)
10. Le nombre de couleurs de la palette (sur 4 octets)
11. Le nombre de couleurs importantes de la palette (sur 4 octets). Ce champ peut être égal à 0 lorsque chaque couleur a son importance.

### 4. Palette de l'image

La palette est optionnelle. Lorsqu'une palette est définie, elle contient successivement 4 octets pour chacune de ses entrées représentant :

1. La composante rouge (sur un octet)
2. La composante verte (sur un octet)
3. La composante bleue (sur un octet)
4. Un champ réservé (sur un octet)

Remarque :

- en BMP 24 bits (1 octet bleu, 1 octet vert, 1 octet rouge par pixel), la palette n'est pas nécessaire (on peut représenter toutes les couleurs). Dans ce cas, la palette n'existe pas dans le fichier BMP ;
- en BMP 8 bits, on ne peut représenter que 256 couleurs, il faut définir une correspondance entre la couleur du pixel et les trois composantes. Il a donc été ajouté une table juste après l'en-tête (octet 56), qui donne pour chaque valeur (de 0 à 255) les trois composantes RVB qui y correspondent).

La palette est donc une table de correspondance de type

Valeur à afficher	RVB
0	R=r0, B=b0, V=v0 + octet 00
...	
255	R=r255, B=b255, V=v255 + octet 00

Par exemple, pour une image en niveaux de gris (256 nuances), les trois valeurs R, V et B seront égales et varieront entre 0 et 255.

La structure du fichier BMP devient donc :

- en-tête
- palette à l'octet 54 (36 en hexadécimal)
- image à l'octet 1 078 (0436 en hexadécimal)

il faut tenir compte de cela lors du calcul de l'en-tête.

## 5. Codage de l'image

Le codage de l'image se fait en écrivant successivement les bits correspondant à chaque pixel, ligne par ligne **de gauche à droite puis de bas en haut**.

- Les images en 2 couleurs utilisent 1 bit par pixel, ce qui signifie qu'un octet permet de coder 8 pixels
- Les images en 16 couleurs utilisent 4 bits par pixel, ce qui signifie qu'un octet permet de coder 2 pixels
- Les images en 256 couleurs utilisent 8 bits par pixel, ce qui signifie qu'un octet code chaque pixel
- Les images en couleurs réelles utilisent 24 bits par pixel, ce qui signifie qu'il faut 3 octets pour coder chaque pixel, en prenant soin de respecter l'ordre de l'alternance bleu, vert et rouge.

Chaque ligne de l'image doit comporter un nombre total d'octets qui soit un **multiple de 4** ; si ce n'est pas le cas, la ligne doit être complétée par des 0 de telle manière à respecter ce critère.

Dans la palette on écrit successivement dans l'ordre les couleurs en RGB.

## 6. Résolution

Les images matricielles, de type bitmap, sont constituées d'une matrice de points (les pixels) qui

sont si proches les uns des autres qu'il est impossible de les distinguer individuellement. Ce n'est qu'en les agrandissant fortement que ces points deviennent visibles.

Plus les différents pixels sont petits, plus les détails de l'image sont précis et plus la résolution de cette image est élevée. La contrepartie est que le poids du fichier augmente considérablement, dans la mesure où pour une image bitmap, chaque pixel existe dans sa couleur spécifique.

Outre le volume considérable requis, ces images ont un autre défaut, connu sous le nom de crénelage (ou aliasing). Particulièrement visible pour un œil attentif, ce défaut affecte les figures géométriques. Ainsi, une image qui semblait bien lisse vue de loin présente un aspect inesthétique si elle est examinée de plus près. Le crénelage limite les possibilités d'agrandissement des images bitmap : il n'est pas possible d'agrandir simplement une portion d'une image sans atteindre les limites de la résolution.

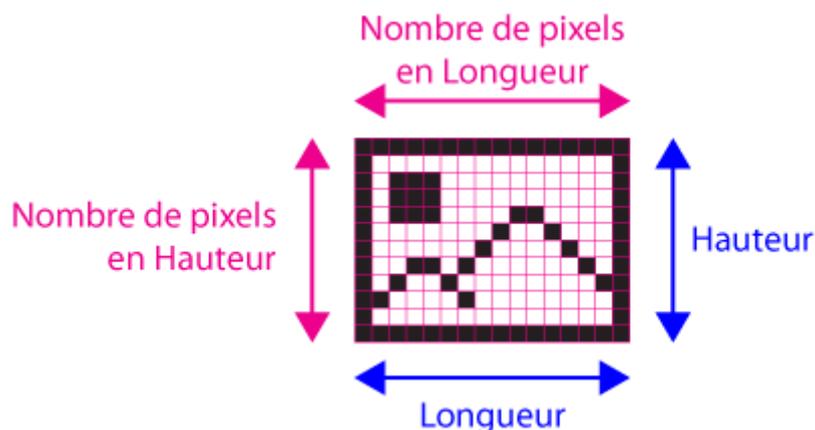
La résolution d'une image matricielle est donc le **nombre de pixels qui composent l'image par unité de longueur**. Elle correspond au nombre de pixels contenu sur l'image dans sa Longueur divisé par sa taille en Longueur ou correspond au nombre de pixels contenu sur l'image dans la Hauteur divisé par sa taille en Hauteur.

$$\text{Résolution} = \frac{\text{Nombre de pixels composant l'image dans sa Longueur}}{\text{Taille de l'image en Longueur}}$$

ou

$$\text{Résolution} = \frac{\text{Nombre de pixels composant l'image dans sa Hauteur}}{\text{Taille de l'image en Hauteur}}$$

L'unité de longueur étant très souvent le pouce<sup>4</sup>, elle s'exprime en : **ppp** (pixel par pouce) ou **ppi** (pixel per inch).



La résolution d'une image bitmap définit le degré de détails de l'image.

Ainsi, plus la résolution est élevée, meilleure est la restitution de l'image.

## 7. Définition

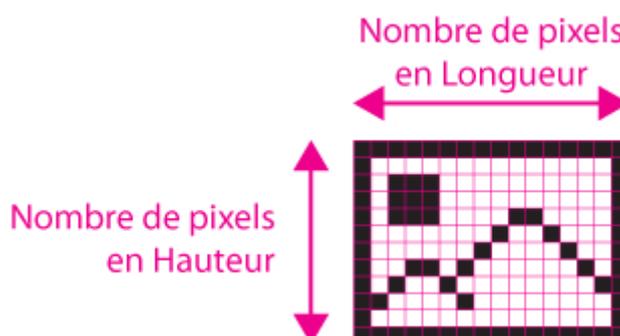
La Définition d'une image matricielle correspond à la **quantité de pixels composant l'image**.

La Définition d'une image bitmap correspond donc au produit du nombre de pixels qui compose l'image en Longueur (axe horizontal) par celui de sa Hauteur (axe vertical).

---

<sup>4</sup> inch

Définition = ( Nombre de pixel en Longueur ) x ( Nombre de pixel en Hauteur )



Plus une image est définie, plus est "lourde", c'est à dire composée d'un grand nombre de pixels.

On peut aussi, depuis les appareils photo numériques, exprimer la définition en chiffre brut : 3 millions de pixels par exemple, signifie 2048 x 1536 pixels en image au ratio 4/3, et 8 millions de pixels 3264 x 2448 pixels.

## 8. Taille du fichier

Le format BMP est quasiment inexistant sur le Web : il ne dégrade pas l'image et n'utilise généralement pas de compression, aussi est-il très lourd. Il existe néanmoins la compression RLE<sup>5</sup> pour le format BMP.

Pour calculer la taille approximative du fichier BMP final, on multiplie la profondeur couleurs (1, 4, 8, 16 ou 24 bits) par la hauteur et par la largeur en pixels : (Profondeur de couleurs en bits) × (Hauteur en pixels) × (Largeur en pixels) bits.

Par exemple, une image en 800 × 600 (800 pixels de large par 600 pixels de haut) en 24 bits (16,8 millions de couleurs) aura une taille de (800 × 600 × 24) bits soit 1,44 Mo (1,37 Mio).

## 9. RLE

Le système RLE s'applique essentiellement à des documents scannés en noir et blanc : au lieu de coder un bit par point, on dispose d'un compteur — en général sur un octet — indiquant combien de points blancs ou noirs se suivent. Comme il est fréquent d'avoir au moins 8 pixels noirs ou 8 pixels blancs de suite, et que 256 ne sont pas rares sur les endroits vierges ou les à-plats noirs, le système a bien pour effet une compression. S'il y a plus de 256 bits de la même couleur, on peut placer ensuite un octet spécifiant 0 bit de la couleur opposée, puis coder le nombre de bits qui restent.

Les formats d'images utilisent cette compression en considérant que toutes les lignes de pixels sont jointes pour former une unique séquence de couleurs.

- Le format BMP de Windows et OS/2 permet d'utiliser la compression RLE pour les images en 1, 4 et 8 bits/pixel (respectivement noir & blanc, 16 couleurs et 256 couleurs).
- Le format PCX utilise également le principe de la compression RLE pour les images en 8 et 24 bits/pixel. Dans le cas des images en 24 bits/pixel, l'image est en fait découpée en trois plans de couleur (rouge, vert et bleu) où chaque plan est encodé comme une image en 8 bits/pixel.

Exemple : considérons un écran de texte noir sur fond blanc constitué de longues séquences de

<sup>5</sup> run-length encoding : codage par plages, algorithme de compression de données sans perte

pixels blancs pour le fond, et de courtes séquences de pixels noirs pour le texte. Représentons une ligne d'un tel écran, avec B (comme black) pour les pixels noirs et W (comme white) pour les pixels blancs :

WWWWWWWWWWWWBWWWWWWWWWWWWWWWWBBBWWWWWWWWWWWWWW  
WWWWWWWWWWWWBWWWWWWWWWWWWWW

Un encodage RLE consiste à indiquer pour chaque suite de pixels d'une même couleur le nombre de pixels de cette séquence. Le résultat comporte en général moins de caractères, bien que ça ne soit pas une obligation. On obtient, par exemple, pour la ligne précédente : 12W1B14W3B23W1B11W

Tandis que :

WBWBWBWBWB

donnerait : 1W1B1W1B1W1B1W1B1W1B

Ce qui est deux fois plus long.