

| PIF File Header             |         |                |         |
|-----------------------------|---------|----------------|---------|
| Signature                   |         |                |         |
| File Size                   |         |                |         |
| File Offset to PixelArray   |         |                |         |
|                             |         |                |         |
| Image Information Header    |         |                |         |
| Image Type                  |         | Bits per Pixel |         |
| Image Width                 |         | Image Height   |         |
| Image Size                  |         |                |         |
| Color Table Size            |         | Compression    |         |
|                             |         |                |         |
| Color Table (semi-optional) |         |                |         |
| Red 0                       | Green 0 | Blue 0         | Red 1   |
| Green 1                     | Blue 1  | Red 2          | Green 2 |
| Blue 2                      | Red 3   | Green 3        | Blue 3  |
| •                           |         |                |         |
| •                           |         |                |         |
| •                           |         |                |         |
| Red n                       | Green n | Blue n         | { PAD } |
|                             |         |                |         |
| Byte offset:                |         |                |         |
| 0                           | 1       | 2              | 3       |

### The Structure of Portable Image File Format (PIF)

**Note:** The size of the File and Image header are fixed

**Note:** The presence of the Color Table is mandatory when Bits per Pixel  $\leq 8$ , unless Image Type states RGB332, RGB16C or B/W

**Note:** The size of the Color Table Entries is always 3 Bytes. Extra padding is added at the end to keep the Table a multiple of 4 Bytes

**Note:** Pixel size depending on [Bits per Pixel] field. If Bits per Pixel is  $\leq 4$ , multiple Pixels are grouped in one Byte but are not allowed to overlap the byte / 8-bit boundary.

| Image Data PixelArray [x,y] |              |              |              |     |                |
|-----------------------------|--------------|--------------|--------------|-----|----------------|
| Pixel[0,0]                  | Pixel[1,0]   | Pixel[2,0]   | Pixel[3,0]   | ... | Pixel[w-1,0]   |
| Pixel[0,1]                  | Pixel[1,1]   | Pixel[2,1]   | Pixel[3,1]   | ... | Pixel[w-1,1]   |
| Pixel[0,2]                  | Pixel[1,2]   | Pixel[2,2]   | Pixel[3,2]   | ... | Pixel[w-1,2]   |
| •                           |              |              |              |     |                |
| •                           |              |              |              |     |                |
| •                           |              |              |              |     |                |
| Pixel[0,h-2]                | Pixel[1,h-2] | Pixel[2,h-2] | Pixel[3,h-2] | ... | Pixel[w-1,h-2] |
| Pixel[0,h-1]                | Pixel[1,h-1] | Pixel[2,h-1] | Pixel[3,h-1] | ... | Pixel[w-1,h-1] |

**Note:** Little-Endian is used

**Note:** If multiple Pixels are packed within a Byte, handle Pixels from LSB to MSB

**Note:** Inofficial name of PIF: Pазzy's Image File

| PIF File Header           |   |
|---------------------------|---|
| Signature:                | To identify a valid .PIF file. The signature is <PIL> as string, including null character: {'P','I','L','\0'} |
| File Size:                | Total size of the file, from the Signature to the last Pixel  |
| FileOffset to PixelArray: | Offset to the start of the Pixel Array, to directly seek to the image data                                    |

| Image Information Header |   |
|--------------------------|---|
| Image Type:              | Defines the Image Data Type together with Bits per Pixel<br>Depending on the code, the image data might be indexed <ul style="list-style-type: none"> <li>• 0x433C = RGB888 Uncompressed / -processed image data</li> <li>• 0xE5C5 = RGB565 16-bit image data with reduced color set</li> <li>• 0x1E53 = RGB332 8-bit image data, further reduced colors</li> <li>• 0xB895 = RGB16C 16 color mode with fixed Windows/IBM Colors</li> <li>• 0x7DAA = B/W Black and White color mode</li> <li>• 0xC091 = Indexed Indexed Colors according to the Color Table</li> </ul> |
| Bits per Pixel:          | Bit size that each Pixel occupies. Bit size foran Indexed Image cannot go beyond 8 bits.  |
| Image Width:             | Width of the image in Pixel   |
| Image Height:            | Height of the image in Pixel  |
| Image Size:              | Size of the image in Bytes  |
| Color Table Size:        | Index size (R+G+B) of the color table, only used in Indexed mode, otherwise zero.   |
| Compression:             | If 0x7DDE, then RLE compression is used on the Image Data. If 0x0000, no compression is applied.  |

| Color Table (semi-optional)   |
|---|
| The color data in the table is always 24bit large RGB888. The amount of Indexes has to be same or less than the [Bits per Pixel] allow, otherwise the is invalid. If data refers to a higher index number than the Color Table holds, the image is invalid. |

| Image Data PixelArray   |
|---|
| Raw (uncompressed) Pixel data should be processed as defined by the Image Type, read one by one.  |
| If RLE compression is enabled, the data format looks as following:<br>A negative value defines that the next x-amount of Pixels are individual pixels. A positive value defines that the next Pixel repeats x-times. Zero is a illegal RLE value. |
| Example: RLE (-2) Pixel (4) Pixel (2) RLE (15) Pixel (7)<br>First two Pixels are individual Pixels, the next Pixel to be drawn 15 times.  |

**Note:** Formula used to generate RGB16C mode:

red =  $255 \times [2/3 \times (\text{colorNumber} \& 4) / 4 + 1/3 \times (\text{colorNumber} \& 8) / 8]$   
green =  $255 \times [2/3 \times (\text{colorNumber} \& 2) / 2 + 1/3 \times (\text{colorNumber} \& 8) / 8]$   
blue =  $255 \times [2/3 \times (\text{colorNumber} \& 1) / 1 + 1/3 \times (\text{colorNumber} \& 8) / 8]$