

# The 1-Bit JControl Image File Format (JCIF)

<image> = <header><pixeldata>

<header> = "JCIF"<magicnumber><colordepth><width><height>

<magicnumber> = 0x01 (16-Bit)

<colordepth> = 0x01 (16-Bit)

<width> = image width in pixels (16-Bit)

<height> = image height in pixels (16-Bit)

<pixeldata> = { <eight\_pixel\_row> } +

<eight\_pixel\_row> = { <eight\_pixels> } \* width

<eight\_pixels> = pixels are stored in a byte, one upon the other, LSB first (8-Bit each)

Thus, image pixeldata is always stored as a multiple of eight\*<width>.

Example for an image: (8x10 pixels)

```

      . .XXXX. .
      .X. . . .X.
      X. . . . .X
      X.X. .X.X
      X. . . . .X
      X. . . . .X
      X.XXXX.X
      X. .XX. .X
      .X. . . .X.
      . .XXXX. .

LSB   .   .   X   X   X   X   .   .
      .   X   .   .   .   .   X   .
      X   .   .   .   .   .   .   X
      X   .   X   .   .   X   .   X
      X   .   .   .   .   .   .   X
      X   .   .   .   .   .   .   X
      X   .   X   X   X   X   .   X
MSB   X   .   .   X   X   .   .   X

      FC 02 49 C1 C1 49 02 FC

LSB   .   X   .   .   .   .   X   .
      .   .   X   X   X   X   .   .

      00 01 02 02 02 02 01 00
```

The pixel data will be represented as FC0249C1C14902FC0001020202020100

The whole image will be store as

```
4A 43 49 46 00 01 00 01 00 08 00 0A FC 02 49 C1
C1 49 02 FC 00 01 02 02 02 02 01 00
```

## The 4-Bit JControl Image File Format (JCIF)

**<image>** = **<header>**[**<palette>**]**<pixeldata>**

**<header>** = "JCIF"**<magicnumber>****<colordepth>****<width>****<height>****<foreground>****<background>****<mode>**

**<magicnumber>** = 0x00 (16-Bit)

**<colordepth>** = 0x04 (16-Bit)

**<width>** = image width in pixels (16-Bit)

**<height>** = image height in pixels (16-Bit)

**<foreground>** = the foreground color index in the palette (4-Bit)

**<background>** = the background color index in the palette (4-Bit)

**<mode>** = a set of flags as follows (8-Bit)

COMPRESS 0x01: image is compressed

NOPALETTE 0x02: if set, no palette is included, standard windows palette must be used

TRANSPARENT 0x08: background color that was specified above may not be painted

FOREGROUND 0x10: if not set, the image has no special foreground color, the index at **<foreground>** will be ignored.

BACKGROUND 0x20: if not set, the image has no special background color, the index at **<background>** will be ignored.

The palette is an array of 48 bytes, i.e. three bytes for every color

**<palette>** = { **<color>** } \* 16

**<color>** = **<red>****<green>****<blue>**

**<red>** = the red part of the color (8-Bit)

**<green>** = the green part of the color (8-Bit)

**<blue>** = the blue part of the color (8-Bit)

Note that a palette may only be defined if bit 2 in **<mode>** is not set.

**<pixeldata>** = **<uncompressed\_data>**|**<compressed\_data>**

**<uncompressed\_data>** = { **<pixel\_pair>** } +

**<pixel\_pair>** = **<pixel>****<pixel>**

**<pixel>** = the palette index for the pixel color, 0-15 (4-Bit)

If the image's width is an odd number of pixels, each row will be filled up by 4 bits to complete the last byte, though they will be ignored at painting.

**<compressed\_data>** = { **<pixel\_count>****<pixel>** } +

**<pixel\_count>** = number of following pixels with the same color 1=1, 2=2, ..., 15=15, 0=16 (4-Bit)

**<pixel>** = the palette index for the pixel color, 0-15 (4-Bit)

*NOTE:* This is not row-overlapping, i.e. each row starts with a new **<pixel\_count>****<pixel>**!