

BitFLip (BFL) Image Format

Specification **v1.0**, 1980-01-01, Slendi <slendi@socopon.com>

1. Overview

BFL is a compact, monochrome bitmap image format with optional 1-bit transparency. Each image has two bitplanes that are encoded:

1. **Image bitplane** (required): 1 = white pixel, 0 = black pixel
2. **Alpha bitplane** (optional): 1 = transparent pixel, 0 = opaque pixel

Each bitplane is independently encoded using a run-length encoding or stored raw bit-packed. Each resulting byte stream can then be optionally LZSS-compressed. All multi-byte integers are little-endian.

2. File Structure

```
struct Header {
    char magic[3]; // Magic bytes "BFL"
    u16 width;     // Width of the image in pixels
    u16 height;    // Height of the image in pixels
    u8  flags;     // Various flags
    u32 img_len;   // Image stream length
    u32 al_len;    // Alpha stream length, 0 if none
    u8  img[];     // Image stream bytes
    u8  al[];     // Alpha stream bytes
}
```

2.1. Dimensions

- Width and height are 16-bit unsigned integers.
- Pixel order is row-major, top-to-bottom, left-to-right.
- Total pixel count $N = \text{Width} * \text{Height}$. All bitplane encoders/decoders operate on exactly N bits.

2.2. Flags

- 0×01 **FLAG_HAS_ALPHA** - Alpha bitplane is present.
- 0×02 **FLAG_IMG_RAW** - Image stream is raw bit-packed.
- 0×04 **FLAG_TRA_RAW** - Alpha stream is raw bit-packed.
- 0×08 **FLAG_IMG_NOLZ** - Image stream is **not** LZSS-compressed.
- 0×10 **FLAG_TRA_NOLZ** - Alpha stream is **not** LZSS-compressed.

It is up to the encoder to determine which combination of those flags results in a smaller file.

All remaining bits are reserved and should be set to 0.

3. Bit-packing RAW streams

When a stream is marked RAW in flags, bits are packed LSB-first within each byte.

- Bit for pixel index i is stored at byte $i / 8$, bit position $i \% 8$.
- Unused high bits of the final byte (if N is not a multiple of 8) **must** be zero when encoding and **must** be ignored when decoding.

4. Coinflip RLE

Unless a stream is marked RAW, each bitplane is encoded using “coinflip” run-length coding:

```
Byte 0: Initial state (0 = start with 0-runs, 1 = start with 1-runs)
Byte 1..k: Repeated groups of:
    Count      (u8, number of pixels to emit of the current state)
    [Optional 0] (u8, do not toggle state if present)
```

5. LZSS

After coinflip RLE or RAW bit-packing, each resulting byte stream may be compressed independently using LZSS with the following parameters:

- **Window size:** 4096
- **Lookahead:** 18
- **Minimum match:** 3

5.1. Block format

Streams are encoded as a sequence of 8-item groups preceded by a flag byte:

- For each bit b (0..7) in the flag (LSB first):
 - If $(\text{flag} \gg b) \ \& \ 1 == 1$: Literal — copy next byte to output.
 - Else: Match — read two bytes b_0 , b_1 and emit:
 - $\text{length} = (\text{b0} \gg 4) + 3$ (range 3..18)
 - $\text{offset} = ((\text{b0} \ \& \ 0 \times 0 \text{F}) \ll 8) \mid \text{b1}$ (range 1..4095)
 - Copy length bytes from $\text{out_size} - \text{offset}$.