Example of how information may appear

* Description: ??
  + Size = ??
  + Location = ??

PNG files consist of multiple blocks of data. The blocks have an identifier that is 4 bytes in size.

The blocks are known as chunks at consist of 4 sections. They will be referred to blocks for ease of understanding.

* Description: Length of the block
  + Size = 4
  + Location = 0
    - Assume start of block
* Description: Type
  + Size = 4
  + Location = 4
* Description: Data
  + Size = LengthOfTheBlock
  + Location = 8
* Description: CRC
  + Size = 4
  + Location = 8+LengthOfTheBlock
  + Checksum value. Should not be skipped but it will be for now.

PNG files can contain many blocks however, only 4 are required. The others can be skipped.

IHDR, PLTE, IDAT, and IEND are the bare minimum that a decoder must be able to interpret.

PNG files start with an 8-byte header. It is as follows.

* Description: Header
  + Size = 8
  + Location = 0
  + Can be skipped
  + Will always look like this
    - 89 50 4e 47 0d 0a 1a 0a

IHDR block - It contains important information about the image such as the size.

* Description: Width
  + Size = 4
  + Location = 0
    - Assume that this is the start of the data in the block
* Description: Height
  + Size = 4
  + Location = 4
* Description: Bit Depth
  + Size = 1
  + Location = 8
  + Valid values are
    - 1, 2, 4, 8, 16
    - Different for color types though
* Description: Color Type
  + Size = 1
  + Location = 9
  + Valid values are
    - 0, 2, 3, 4, 6
    - Greyscale, Truecolor, Indexed-color, Greyscale with alpha, Truecolor with alpha
* Description: Compression Method
  + Size = 1
  + Location = 10
  + Must be 0
* Description: Filter method
  + Size = 1
  + Location = 11
* Description: Interlace method
  + Size = 1
  + Location = 12
  + Valid values are
    - 0 – No interlacing, 1 – Adam7 interlacing

PLTE block - It contains information about the palette if an image has one

* Description: RGB Triplet
  + Size = 3
  + Location = 0
    - Occurs n times where n is the length of the data.
    - The length must be divisible by 3

IDAT block - It contains compressed image data. Compressed using deflate and zlib

* Description: Compressed Data
  + Size = n
    - N is the length of the block
  + Location = 0

IEND block - It is the end of the PNG file.

Nothing is inside the chunk’s data field

tRNS block - It contains information about transparency in a palette.

(Not required, however it is good to have.)

(Only valid for color type 0, 2, and 3)

* Description: Grey sample value (Color type 0)
  + Size = 2
  + Location = 0
  + The grey value that is transparent
* Description: RGB sample value (Color type 2)
  + Size = 6
  + Location = 0
  + The RGB value that is transparent
    - R – 2 bytes
    - G – 2 bytes
    - B – 2 bytes
* Description: Palette sample value (Color type 3)
  + Size = N
    - N is the length of the block
  + Location = 0
  + How transparent is an index.

CRC checksum value:

TODO

Notes about filtering

Filtering is done on bytes and not pixels.

The first byte on the scanline determines which type of filtering it is. Filter method 0 is the only one defined in the standard.

Filt -> The filtered value

Orig -> The original value

Recon -> The reconstructed value. Also, the original value.

If there is no byte to the left, above, or above and left, it is considered 0.

* Type 0
  + No filtering
    - Filt (x, y) = Orig(x, y)
      * Recon (x, y) = Filt (x, y)
* Type 1
  + Subtractive filtering
    - Filt (x, y) = Orig(x, y) – Orig(x-1, y)
      * Recon (x, y) = Filt (x, y) + Recon(x-1, y)
* Type 2
  + Upward filtering
    - Filt (x, y) = Orig(x, y) – Orig(x, y-1)
      * Recon (x, y) = Filt (x, y) + Recon(x, y-1)
* Type 3
  + Average filtering
    - Filt (x, y) = Orig(x, y) – floor( (Orig(x-1, y) + Orig(x, y-1)) / 2)
      * Recon (x, y) = Filt (x, y) + floor( (Recon(x-1, y) + Recon(x, y-1)) / 2)
* Type 4
  + Paeth filtering
    - Filt (x, y) = Orig(x, y) – PaethPredictor( Orig(x-1, y), Orig(x, y-1), Orig(x-1, y-1)
      * Recon (x, y) = Filt(x, y) + PaethPredictor( Recon(x-1, y), Recon(x, y-1), Recon(x-1, y-1)
    - PaethPredictor(A, B, C):
      * The value which is closest to p = A+B-C