Portable Network Graphics

Portable Network Graphics (**PNG** /ˈpɪŋ/[2][3]) is a raster graphics file format that supports lossless data compression. PNG was created as an improved, non-patented replacement for Graphics Interchange Format (GIF), and is the most used lossless image compression format on the Internet.[4]

PNG supports palette-based images (with palettes of 24-bit RGB or 32-bit RGBA colors), grayscale images (with or without alpha channel), and full-color non-palette-based RGB[A] images (with or without alpha channel). PNG was designed for transferring images on the Internet, not for professional-quality print graphics, and therefore does not support non-RGB color spaces such as CMYK.

PNG files nearly always use file extension PNG or png and are assigned MIME media type image/png. PNG was approved for this use by the Internet Engineering Steering Group on 14 October 1996,[5] and was published as an ISO/IEC standard in 2004.[1]

1 History and development

See also: Graphics Interchange Format § Unisys and LZW patent enforcement

The motivation for creating the PNG format was in early 1995, after it became known that the Lempel–Ziv–Welch (LZW) data compression algorithm used in the Graphics Interchange Format (GIF) format was patented by Unisys. There were also other problems with the GIF format that made a replacement desirable, notably its limit of 256 colors at a time when computers able to display far more than 256 colors were becoming common.

A January 1995 precursory discussion thread, on the usenet newsgroup "comp.graphics" with the subject *Thoughts on a GIF-replacement file format*, had many propositions, which would later be part of the PNG file format. In this thread, Oliver Fromme, author of the popular DOS JPEG viewer QPEG, proposed the PING name, meaning *PING is not GIF*, and also the PNG extension.[6]

Although GIF allows for animation, it was decided that PNG should be a single-image format.[7] In 2001, the developers of PNG published the Multiple-image Network Graphics (MNG) format, with support for animation. MNG achieved moderate application support, but not enough among mainstream web browsers and no usage among web site designers or publishers. In 2008, certain Mozilla developers published the Animated Portable Network Graphics (APNG) format with similar goals. APNG is a format that is natively supported by Gecko- and Presto-based web browsers and is also commonly used for thumbnails on Sony's Playstation Portable system (using the normal PNG file extension), but as of 2013, usage of APNG remains very minimal.

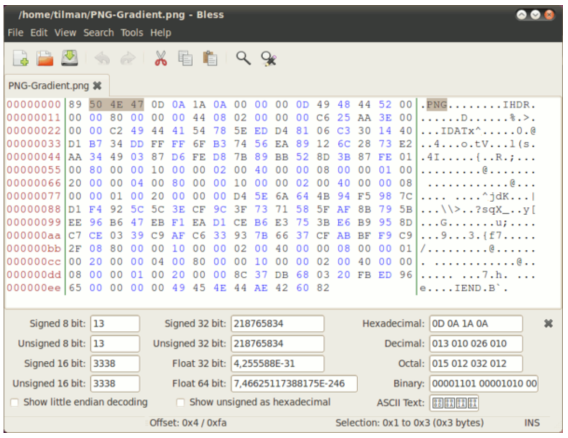
- 1 October 1996: Version 1.0 of the PNG specification was released, and later appeared as RFC 2083. It became a W3C Recommendation on 1 October 1996.
- 31 December 1998: Version 1.1, with some small changes and the addition of three new chunks, was released.
- 11 August 1999: Version 1.2, adding one extra chunk, was released.
- 10 November 2003: PNG became an International Standard (ISO/IEC 15948:2003). This version of PNG differs only slightly from version 1.2 and adds no new chunks.
- 3 March 2004: ISO/IEC 15948:2004.[1]

2 PNG Working Group

The original PNG specification was authored by an ad-hoc group of computer graphics experts and enthusiasts. Discussions and decisions about the format were done exclusively via email. The original authors listed on RFC 2083 are:[8]

- Editor: Thomas Boutell
- Contributing Editor: Tom Lane
- Authors (in alphabetical order): Mark Adler, Thomas Boutell, Christian Brunschen, Adam M. Costello, Lee Daniel Crocker, Andreas Dilger, Oliver Fromme, Jean-Loup Gailly, Chris Herborth, Aleks Jakulin, Neal Kettler, Tom Lane, Alexander Lehmann, Chris Lilley, Dave Martindale, Owen Mortensen, Keith S. Pickens, Robert P. Poole, Glenn Randers-Pehrson, Greg Roelofs, Willem van Schaik, Guy Schalnat, Paul Schmidt, Tim Wegner, Jeremy Wohl

3 Technical details


/home/tilman/PNG-Gradient.png - Bless
File Edit View Search Tools Help
PNG-Gradient.png

```
00000000 89 50 4E 47 0D 0A 1A 0A 00 00 00 0D 49 48 44 52 00 .PNG........IHDR.
00000011 00 00 80 00 00 00 44 08 02 00 00 00 C6 25 AA 3E 00 ......D......%.>.
00000022 00 00 C2 49 44 41 54 78 5E ED D4 81 06 C3 30 14 40 ...IDATx^.....0.@
00000033 D1 B7 34 DD FF FF 6F B3 74 56 EA 89 12 6C 28 73 E2 ..4...o.tV...l(s.
00000044 AA 34 49 03 87 D6 FE D8 7B 89 BB 52 8D 3B 87 FE 01 .4I.....{..R.;...
00000055 00 80 00 00 10 00 00 02 00 40 00 00 08 00 00 01 00 .........@.......
00000066 20 00 00 04 00 80 00 00 10 00 00 02 00 40 00 00 08  ............@...
00000077 00 00 01 00 20 00 00 00 D4 5E 6A 64 4B 94 F5 98 7C .... ....^jdK...|
00000088 D1 F4 92 5C 5C 3E CF 9C 3F 73 71 58 5F AF 8B 79 5B ...\\>..?sqX_..y[
00000099 EE 96 B6 47 EB F1 EA D1 CE B6 E3 75 3B E6 B9 95 8D ...G.......u;....
000000aa C7 CE 03 39 C9 AF C6 33 93 7B 66 37 CF AB BF F9 C9 ...9...3.{f7.....
000000bb 2F 08 80 00 00 10 00 00 02 00 40 00 00 08 00 00 01 /.........@......
000000cc 00 20 00 00 04 00 80 00 00 10 00 00 02 00 40 00 00 . ............@..
000000dd 08 00 00 01 00 20 00 00 8C 37 DB 68 03 20 FB ED 96 ..... ...7.h. ...
000000ee 65 00 00 00 00 49 45 4E 44 AE 42 60 82             e....IEND.B`.
```

Signed 8 bit: 13 Signed 32 bit: 218765834 Hexadecimal: 0D 0A 1A 0A
Unsigned 8 bit: 13 Unsigned 32 bit: 218765834 Decimal: 013 010 026 010
Signed 16 bit: 3338 Float 32 bit: 4,255588E-31 Octal: 015 012 032 012
Unsigned 16 bit: 3338 Float 64 bit: 7,46625117388175E-246 Binary: 00001101 00001010 00
Show little endian decoding Show unsigned as hexadecimal ASCII Text:
Offset: 0x4 / 0xfa Selection: 0x1 to 0x3 (0x3 bytes) INS

The PNG image viewed with a hex editor

3.1 File header

A PNG file starts with an 8-byte signature:[9] (see hex editor image at right)

3.2 "Chunks" within the file

After the header comes a series of chunks, each of which conveys certain information about the image. Chunks declare themselves as *critical* or *ancillary*, and a program encountering an ancillary chunk that it does not understand can safely ignore it. This chunk-based storage layer structure, similar in concept to a container format, is designed to allow the PNG format to be extended while maintaining compatibility with older versions—it provides forward compatibility, and this same file structure (with different signature and chunks) is used in the associated MNG, JNG, and APNG formats.

A chunk consists of four parts: length (4 bytes), chunk type/name (4 bytes), chunk data (length bytes) and CRC (cyclic redundancy code/checksum; 4 bytes). The CRC is a network-byte-order CRC-32 computed over the chunk type and chunk data, but not the length.

Chunks are given a four-letter case sensitive ASCII type/name; compare FourCC. The case of the different letters in the name (bit 5 of the numeric value of the character) is a bit field that provides the decoder with some information on the nature of chunks it does not recognize.

The case of the first letter indicates whether the chunk is critical or not. If the first letter is uppercase, the chunk is critical; if not, the chunk is ancillary. Critical chunks contain information that is necessary to read the file. If a decoder encounters a critical chunk it does not recognize, it must abort reading the file or supply the user with an appropriate warning.

The case of the second letter indicates whether the chunk is "public" (either in the specification or the registry of special-purpose public chunks) or "private" (not standardised). Uppercase is public and lowercase is private. This ensures that public and private chunk names can never conflict with each other (although two private chunk names could conflict).

The third letter must be uppercase to conform to the PNG specification. It is reserved for future expansion. Decoders should treat a chunk with a lower case third letter the same as any other unrecognised chunk.

The case of the fourth letter indicates whether the chunk is safe to copy by editors that do not recognize it. If lowercase, the chunk may be safely copied regardless of the extent of modifications to the file. If uppercase, it may only be copied if the modifications have not touched any critical chunks.

3.2.1 Critical chunks

A decoder must be able to interpret critical chunks to read and render a PNG file.

- IHDR must be the first chunk; it contains (in this order) the image's width, height, bit depth and color type.[10]
- PLTE contains the palette; list of colors.
- IDAT contains the image, which may be split among multiple IDAT chunks. Such splitting increases filesize slightly, but makes it possible to generate a PNG in a streaming manner. The IDAT chunk contains the actual image data, which is the output stream of the compression algorithm.[11]
- IEND marks the image end.

The PLTE chunk is essential for color type 3 (indexed color). It is optional for color types 2 and 6 (truecolor and truecolor with alpha) and it must not appear for color types 0 and 4 (grayscale and grayscale with alpha).

3.2.2 Ancillary chunks

Other image attributes that can be stored in PNG files include gamma values, background color, and textual metadata information. PNG also supports color management through the inclusion of ICC color space profiles.[12]

- bKGD gives the default background color. It is intended for use when there is no better choice available, such as in standalone image viewers (but not web browsers; see below for more details).
- cHRM gives the chromaticity coordinates of the display primaries and white point.

- gAMA specifies gamma.
- hIST can store the histogram, or total amount of each color in the image.
- iCCP is an ICC color profile.
- iTXt contains UTF-8 text, compressed or not, with an optional language tag. iTXt chunk with the keyword 'XML:com.adobe.xmp' can contain Extensible Metadata Platform (XMP).
- pHYs holds the intended pixel size and/or aspect ratio of the image.
- sBIT (significant bits) indicates the color-accuracy of the source data.
- sPLT suggests a palette to use if the full range of colors is unavailable.
- sRGB indicates that the standard sRGB color space is used.
- sTER stereo-image indicator chunk for stereoscopic images.[13]
- tEXt can store text that can be represented in ISO/IEC 8859-1, with one name=value pair for each chunk.
- tIME stores the time that the image was last changed.
- tRNS contains transparency information. For indexed images, it stores alpha channel values for one or more palette entries. For truecolor and grayscale images, it stores a single pixel value that is to be regarded as fully transparent.
- zTXt contains compressed text with the same limits as tEXt.

The lowercase first letter in these chunks indicates that they are not needed for the PNG specification. The lowercase last letter in some chunks indicates that they are safe to copy, even if the application concerned does not understand them.

3.3 Pixel format

Pixels in PNG images may contain either a number indexing sample data in the separate table, the palette, contained in the PLTE chunk or the sample data itself, encoded as between one and four numbers. In both cases the numbers are referred to as channels and every number in the image is encoded with an identical format.

The permitted formats encode each number as an unsigned integral value using a fixed number of bits, referred to in the PNG specification as the *bit depth*. Notice that this is not the same as color depth, which is commonly used to refer to the total number of bits in each pixel, not each channel. The permitted bit depths are summarized in the table along with the total number of bits used for each pixel.

The number of channels will depend on whether the image is grayscale or color and whether it has an alpha channel. PNG allows the following combinations of channels, called the *color type*.

- 0: grayscale
- 2: red, green and blue: rgb/truecolor
- 3: indexed: channel containing indices into a palette of colors
- 4: grayscale and alpha: level of opacity for each pixel
- 6: red, green, blue and alpha

The color type is specified as an 8-bit value however only the low 3 bits are used and, even then, only the five combinations listed above are permitted. So long as the color type is valid it can be considered as a bit field as summarized in the table to the right:

- bit value 1: the image data stores palette indices. This is only valid in combination with **bit value 2**;
- bit value 2: the image samples contain three channels of data encoding trichromatic colors, otherwise the image samples contain one channel of data encoding relative luminance,
- bit value 4: the image samples also contain an alpha channel expressed as a linear measure of the opacity of the pixel. This is not valid in combination with **bit value 1**.

With indexed color images, the palette always stores trichromatic colors at a depth of 8 bits per channel (24 bits per palette entry). Additionally, an optional list of 8-bit alpha values for the palette entries may be included; if not included, or if shorter than the palette, the remaining palette entries are assumed to be opaque. The palette must not have more entries than the image bit depth allows for, but it may have fewer (for example, if an image with 8-bit pixels only uses 90 colors then it does not need palette entries for all 256 colors). The palette must contain entries for all the pixel values present in the image.

The standard allows indexed color PNGs to have 1, 2, 4 or 8 bits per pixel; grayscale images with no alpha channel may have 1, 2, 4, 8 or 16 bits per pixel. Everything else uses a bit depth per channel of either 8 or 16. The combinations this allows are given in the table above. The standard requires that decoders can read all supported color formats, but many image editors can only produce a small subset of them.

3.4 Transparency of image

PNG offers a variety of transparency options. With truecolor and grayscale images either a single pixel value can be declared as transparent or an alpha channel can be added (enabling any percentage of partial transparency to be used). For paletted images, alpha values can be added to palette entries. The number of such values stored may be less than the total number of palette entries, in which case the remaining entries are considered fully opaque.

The scanning of pixel values for binary transparency is supposed to be performed before any color reduction to avoid pixels' becoming unintentionally transparent. This is most likely to pose an issue for systems that can decode 16-bits-per-channel images (as they must to be compliant with the specification) but only output at 8 bits per channel (the norm for all but the highest end systems).

Alpha storage can be "associated" ("premultiplied") or "unassociated", but PNG standardized[15] on "unassociated" ("non-premultiplied") alpha so that images with separate transparency masks can be stored losslessly.

3.5 Compression

PNG uses a 2-stage compression process:

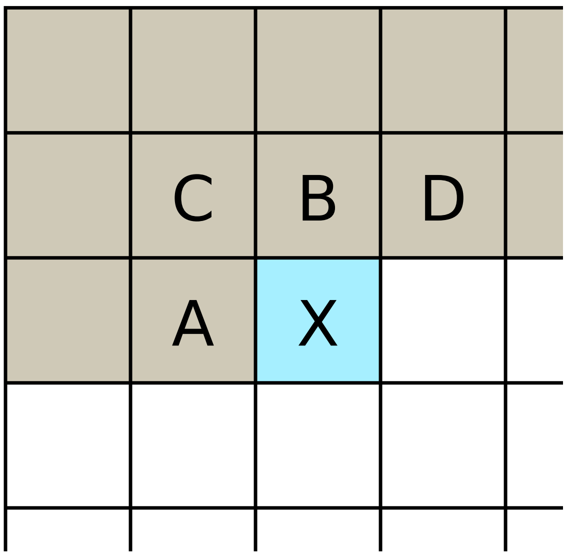
- pre-compression: filtering (prediction)
- compression: DEFLATE

PNG uses a non-patented lossless data compression method known as DEFLATE, which is the same algorithm used in the zlib compression library.

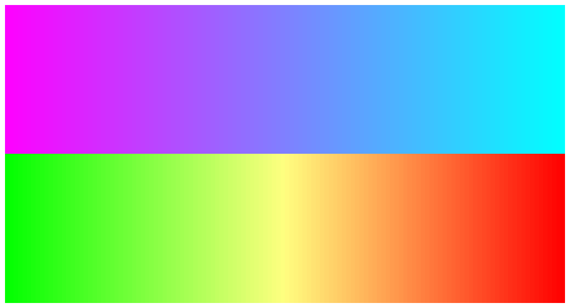
3.5.1 Filtering

Before DEFLATE is applied, the data is precompressed, via a prediction method: a single *filter method* is used for the entire image, while for each image line, a *filter type* is chosen that transforms the data so that it is hopefully more easily compressed.[16]

There is only one filter method in the current PNG specification (denoted method 0), and thus in practice the only choice is which filter type to apply to each line. For this method, the filter predicts the value of each pixel based on the values of previous neighboring pixels, and subtracts the predicted color of the pixel from the actual value, as in DPCM. An image line filtered in this way is often more compressible than the raw image line would be, especially if it is similar to the line above, since the differences from prediction will generally be clustered around 0, rather than spread over all possible image values. This is particularly important in relating separate rows, since DEFLATE has no understanding that an image is a 2D entity, and instead just sees the image data as a stream of bytes.



PNG's filter method 0 can use the data in pixels A, B, and C to predict the value for X.


A PNG with 256 colors, which is only 251 bytes large with pre-filter. The same image as a GIF would be more than thirteen times larger.

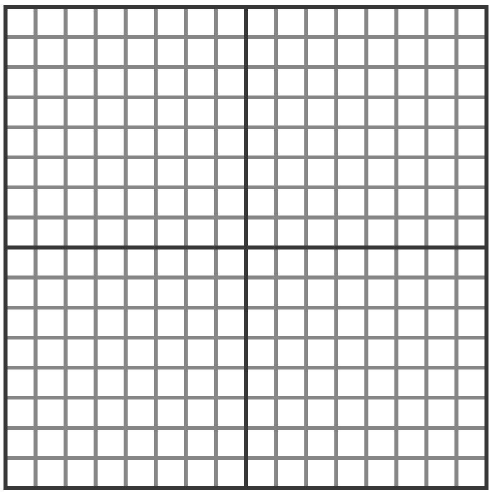
There are five filter types for filter method 0; each type predicts the value of each byte (of the image data before filtering) based on the corresponding byte of the pixel to the left (A), the pixel above (B), and the pixel above and to the left (C) or some combination thereof, and encodes the *difference* between the predicted value and the actual value. Filters are applied to byte values, not pixels; pixel values may be one or two bytes, or several values per byte, but never cross byte boundaries. The filter types are:[17]

The Paeth filter is based on an algorithm by Alan W. Paeth.[18] Compare to the version of DPCM used in lossless JPEG, and to the discrete wavelet transform using 1×2, 2×1, or (for the Paeth predictor) 2×2 windows and Haar wavelets.

Compression is further improved by choosing filter types adaptively on a line-by-line basis. This improvement, and a heuristic method of implementing it commonly used by PNG-writing software, were created by Lee Daniel Crocker, who tested the methods on many images during the creation of the format;[19] the choice of filter is a component of file size optimization, as discussed below.

If interlacing is used, each stage of the interlacing is filtered separately, meaning that the image can be progressively rendered as each stage is received; however, interlacing generally makes compression less effective.

3.6 Interlacing



An illustration of Adam7 interlacing over a 16×16 image.

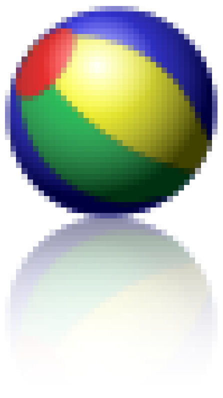
PNG offers an optional 2-dimensional, 7-pass interlacing scheme—the Adam7 algorithm. This is more sophisticated than GIF's 1-dimensional, 4-pass scheme, and allows a clearer low-resolution image to be visible earlier in the transfer, particularly if interpolation algorithms such as bicubic interpolation are used.[20]

However, the 7-pass scheme tends to reduce the data's compressibility more than simpler schemes.

3.7 Animation

PNG itself does not support animation at all. MNG is an extension to PNG that does; it was designed by members of the PNG Group. MNG shares PNG's basic structure and chunks, but it is significantly more complex and has a different file signature, which automatically renders it incompatible with standard PNG decoders.

The complexity of MNG led to the proposal of APNG by developers of the Mozilla Foundation. It is based on PNG, supports animation and is simpler than MNG. APNG offers fallback to single-image display for PNG decoders that do not support APNG. However, neither of these formats is currently widely supported. APNG is supported in Firefox 3.0 and Opera 9.5,[21] but since Opera changed its layout engine to Blink, support was dropped. The latest version of Safari on iOS 8 and Safari 8 for OS X Yosemite support APNG.[22] The PNG Group decided in April 2007 not to embrace APNG.[23] Several



An animated PNG (displays as static image in some web browsers)

alternatives were under discussion, ANG, aNIM/mPNG, "PNG in GIF" and its subset "RGBA in GIF".[24]

4 Comparison with other file formats

Main article: Comparison of graphics file formats

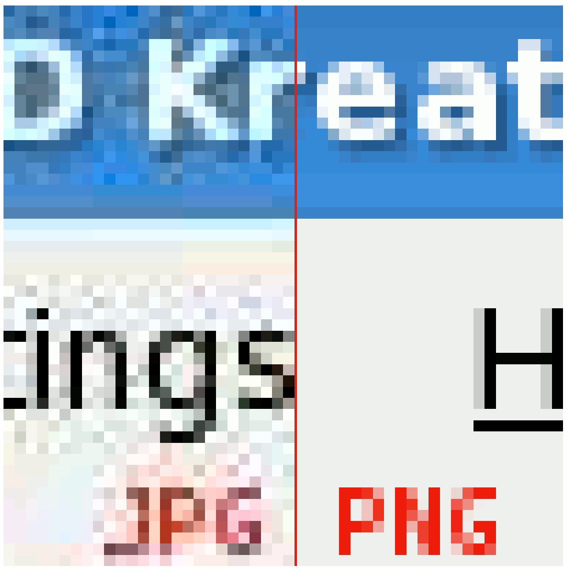
4.1 Graphics Interchange Format (GIF)

- On small images, GIF can achieve greater compression than PNG (see the section on filesize, below).
- On most images, except for the above case, a GIF will have a larger file-size than an indexed PNG.
- PNG gives a much wider range of transparency options than GIF, including alpha channel transparency.
- Whereas GIF is limited to 8-bit indexed color, PNG gives a much wider range of color depths, including 24-bit (8 bits per channel) and 48-bit (16 bits per channel) truecolor, allowing for greater color precision, smoother fades, etc.[25] When an alpha channel is added, up to 64 bits per pixel (before compression) are possible.
- When converting an image from the PNG format to GIF, the image quality may suffer due to posterization if the PNG image has more than 256 colors.

- GIF intrinsically supports animated images. PNG supports animation only via unofficial extensions (see the section on animation, above).

PNG images are less widely supported by older browsers. In particular, IE6 has limited support for PNG.[26]

4.2 JPEG



Composite image comparing lossy compression in JPEG with lossless compression in PNG: the JPEG artifacts are easily visible in the background, where the PNG image has solid color.

JPEG (Joint Photographic Experts Group) format can produce a smaller file than PNG for photographic (and photo-like) images, since JPEG uses a lossy encoding method specifically designed for photographic image data, which is typically dominated by soft, low-contrast transitions, and an amount of noise or similar irregular structures. Using PNG instead of a high-quality JPEG for such images would result in a large increase in filesize with negligible gain in quality. In contrast, when storing images that contain text, line art, or graphics – images with sharp transitions and large areas of solid color – the PNG format can compress image data more than JPEG can. Additionally, PNG is lossless, while JPEG produces noticeable visual artifacts around high-contrast areas. Where an image contains both sharp transitions and photographic parts, a choice must be made between the two effects. JPEG does not support transparency.

Because JPEG uses lossy compression, it also suffers from generation loss, where repeatedly decoding and re-encoding an image to save it again causes a loss of information each time, degrading the image. This does not happen with repeated viewing or copying, but only if the file is edited and saved over again. Because PNG is lossless, it is suitable for storing images to be edited. While PNG is reasonably efficient when compressing photographic images, there are lossless compression formats designed specifically for photographic images, lossless WebP and Adobe DNG (digital negative) for example. However these formats are either not widely supported, or are proprietary. An image can be stored losslessly and converted to JPEG format only for distribution, so that there is no generation loss.

The PNG specification does not include a standard for embedding Exif image data from sources such as digital cameras. Instead, PNG has various dedicated ancillary chunks for storing the metadata that other file formats (such as JPEG) would typically store in Exif format.

Early web browsers did not support PNG images; JPEG and GIF were the main image formats. JPEG was commonly used when exporting images containing gradients for web pages, because of GIF's limited color depth. However, JPEG compression causes a gradient to blur slightly. A PNG file will reproduce a gradient as accurately as possible for a given bit depth, while keeping the file size small. PNG became the optimal choice for small gradient images as web browser support for the format improved. No images at all are needed to display gradients in modern browsers, as gradients can be created using CSS.

4.3 JPEG-LS

JPEG-LS is a “near-lossless” image format by the Joint Photographic Experts Group, though far less widely known and supported than the other lossy JPEG format discussed above. It is directly comparable with PNG, and has a standard set of test images.[27] On the Waterloo Repertoire ColorSet, a standard set of test images (unrelated to the JPEG-LS conformance test set), JPEG-LS generally performs better than PNG, by 10–15%, but on some images PNG performs substantially better, on the order of 50–75%.[28] Thus, if both of these formats are options and file size is an important criterion, they should both be considered, depending on the image.

4.4 TIFF

Tagged Image File Format (TIFF) is a format that incorporates an extremely wide range of options. While this makes TIFF useful as a generic format for interchange between professional image editing applications, it makes adding support for it to applications a much bigger task and so it has little support in applications not concerned with image manipulation (such as web browsers). The high level of extensibility also means that most applications provide only a subset of possible features, potentially creating user confusion and compatibility issues.

The most common general-purpose, lossless compression algorithm used with TIFF is Lempel–Ziv–Welch (LZW). This compression technique, also used in GIF, was covered by patents until 2003. TIFF also supports the compression algorithm PNG uses (i.e. Compression Tag 0008_{16} 'Adobe-style') with medium usage and support by applications. TIFF also offers special-purpose lossless compression algorithms like CCITT Group IV, which can compress bilevel images (e.g., faxes or black-and-white text) better than PNG's compression algorithm.

PNG supports non-premultiplied alpha only[15] whereas TIFF also supports "associated" (premultiplied) alpha.

5 Software support

The official reference implementation of the PNG format is the programming library *libpng*.[29] It is published as free software under the terms of a permissive free software licence. Therefore, it is usually found as an important system library in free operating systems.

5.1 Bitmap graphics editor support for PNG

Main article: Comparison of raster graphics editors

The PNG format is widely supported by graphics programs, including Adobe Photoshop, Corel's Photo-Paint and Paint Shop Pro, the GIMP, GraphicConverter, Helicon Filter, ImageMagick, Inkscape, IrfanView, Pixel image editor, Paint.NET and Xara Photo & Graphic Designer and many others. Some programs bundled with popular operating systems which support PNG include Microsoft's Paint and Apple's iPhoto and Preview, with the GIMP also often being bundled with popular Linux distributions.

Adobe Fireworks (formerly by Macromedia) uses PNG as its native file format, allowing other image editors and preview utilities to view the flattened image. However, Fireworks by default also stores meta data for layers, animation, vector data, text and effects. Such files should not be distributed directly. Fireworks can instead export the image as an optimized PNG without the extra meta data for use on web pages, etc.

5.2 Web browser support for PNG

Main article: Web browser image format support

PNG support first appeared in Internet Explorer 4.0b1 and in Netscape 4.04.[30]

Despite calls by the Free Software Foundation[31] and the World Wide Web Consortium (W3C),[32] tools such as gif2png,[33] and campaigns such as Burn All GIFs,[34] PNG adoption on websites has been fairly slow due to late and buggy support in Internet Explorer, particularly regarding transparency.[35]

PNG compatible browsers include: Apple Safari, Google Chrome, Mozilla Firefox, Opera, Camino, Internet Explorer 7 (still numerous issues),[36] Internet Explorer 8 (still some issues), Internet Explorer 9 and many others. For the complete comparison, see Comparison of web browsers (Image format support).

Especially versions of Internet Explorer (Windows) below 9.0 have numerous problems which prevent it from correctly rendering PNG images.[36]

- 4.0 crashes on large PNG chunks.[37]
- 4.0 does not include the functionality to view .png files,[38] but there is a registry fix.[36]
- 5.0 and 5.01 have broken OBJECT support.[39]
- 5.01 prints palette images with black (or dark gray) backgrounds under Windows 98, sometimes with radically altered colors.[40]
- 6.0 fails to display PNG images of 4097 or 4098 bytes in size.[41]
- 6.0 cannot open a PNG file that contains one or more zero-length IDAT chunks. This issue was first fixed in security update 947864 (MS08-024). For more information, see this article in the Microsoft Knowledge Base: 947864 MS08-024: Cumulative Security Update for Internet Explorer [42]
- 6.0 sometimes completely loses ability to display PNGs, but there are various fixes.[43]
- 6.0 and below have broken alpha-channel transparency support (will display the default background color instead).[44][45][46] However there are various fixes:
  - Degradable PNG Transparency for IE6
  - webfx - PNG Behavior (IE behavior/.htc)
  - The PNG problem in Windows Internet Explorer (IE behavior/.htc) (unmaintained)
  - TwinHelix - Near-native PNG support with alpha opacity to IE 5.5 and 6 (IE behavior/.htc)
  - A Better IE 5.5 and 6 PNG Fix (supports CSS background-position, background-repeat) (IE behavior/.htc)
  - 24ways.org - Transparent PNGs in Internet Explorer 6 by Drew McLellan (JavaScript)
  - PNG-24 Alpha Transparency With Microsoft Internet Explorer or better (MSIE 5.5+) (PHP)
  - PNGPong, an open source solution to display transparent PNGs in IE, Firefox, and Safari without the use of filters, PHP, or complicated Javascript and CSS (JavaScript+Flash)
  - Cross Browser PNG Transparency (CSS)

  - CSS PNG fix (with background call none fix) (CSS)
  - SitePoint - Use 8-bit PNGs with Fireworks
  - Use 8-bit PNGs with Photoshop and pngquant
  - dillerdesign belatedPNG (JavaScript+VML)
  - Dean Edwards's IE7.js and IE8.js fixes this issue (for specially-named .PNG files, for performance reasons), and other IE 5.5, 6, and 7 CSS incompatibilities as well.

- 7.0 and below cannot combine 8-bit alpha transparency AND element opacity (CSS - filter: Alpha (opacity=xx)) without filling partially transparent sections with black.[47]
- 8.0 and below have inconsistent/broken gamma support.[36]
- 8.0 and below don't have color-correction support.[36]

5.3 Operating system support for PNG icons

PNG icons have been supported in most distributions of Linux since at least 1999, in desktop environments such as GNOME.[48] In 2006, Microsoft Windows support for PNG icons was introduced in Windows Vista.[49] PNG icons are supported in AROS, Mac OS X, iOS and MorphOS as well. In addition, Android makes a large use of PNGs.

6 File size and optimization software

PNG file size can vary significantly depending on how it is encoded and compressed; this is discussed and a number of tips are given in *PNG: The Definitive Guide.*[28]

6.1 Compared to GIF

Compared to GIF files, a PNG file with the same information (256 colors, no ancillary chunks/metadata), compressed by an effective compressor will normally be smaller than GIF. Depending on the file and the compressor, PNG may range from somewhat smaller (10%) to significantly smaller (50%) to somewhat larger (5%), but is rarely significantly larger[28] for large images. This is attributed to the performance of PNG's DEFLATE compared to GIF's LZW, and because the added precompression layer of PNG's predictive filters take account of the 2-dimensional image structure to further compress files; as filtered data encodes differences between pixels, they will tend to cluster closer to 0, rather than being spread across all possible values, and thus be more easily compressed by DEFLATE. However, some versions of Adobe Photoshop, CorelDRAW and MS Paint provide poor PNG compression, creating the impression that GIF is more efficient.[28]

6.2 File size factors

PNG files vary in size due to a number of factors:

color depth Color depth can range from 1 to 64 bits per pixel.

ancillary chunks PNG supports metadata—this may be useful for editing, but unnecessary for viewing, as on websites.

interlacing As each pass of the Adam7 algorithm is separately filtered, this can increase file size.[28]

filter As a precompression stage, each line is filtered by a predictive filter, which can change from line to line. As the ultimate DEFLATE step operates on the whole image's filtered data, one cannot optimize this row-by-row; the choice of filter for each row is thus potentially very variable, though heuristics exist.[note 1]

compression With additional computation, DEFLATE compressors can produce smaller files.

There is thus a filesize trade-off between high color depth, maximal metadata (including color space information, together with information that does not affect display), interlacing, and speed of compression, which all yield large files, with lower color depth, fewer or no ancillary chunks, no interlacing, and tuned but computationally intensive filtering and compression. For different purposes one will choose different trade-offs: a maximal file may be best for archiving and editing, while a stripped down file may be best for use on a website, and similarly fast but poor compression is preferred when repeatedly editing and saving a file, while slow but high compression is preferred when a file is stable: when archiving or posting. Interlacing is a trade-off: it dramatically speeds up early rendering of large files (improves latency), but may increase file size (decrease throughput) for little gain, particularly for small files.[28]

6.2.1 Lossy PNG compression

Even though PNG has been designed as a lossless format, PNG encoders can pre-process image data in a lossy fashion (so as to reduce colors used) to improve PNG compression.[50]

6.3 Image editing software

Some programs are more efficient than others when saving PNG files, this relates to implementation of the PNG compression used by the program.

Many graphics programs (such as Apple's Preview software) save PNGs with large amounts of metadata and color-correction data that are generally unnecessary for Web viewing. Unoptimized PNG files from Adobe Fireworks are also notorious for this since they contain options to make the image editable in supported editors. Also CorelDRAW (at least version 11) sometimes produces PNGs which cannot be opened by Internet Explorer (versions 6–8).

Adobe Photoshop's performance on PNG files has improved in the CS Suite when using the Save For Web feature (which also allows explicit PNG/8 use).

Adobe's Fireworks saves larger PNG files than many programs by default. This stems from the mechanics of its *Save* format: the images produced by Fireworks' save function include large, private chunks, containing complete layer and vector information. This allows further lossless editing. When saved with the *Export* option, Fireworks' PNGs are competitive with those produced by other image editors, but are no longer editable as anything but flattened bitmaps. Fireworks is unable to save size-optimized vector-editable PNGs.

Other notable examples of poor PNG compressors include:

- Microsoft's Paint for Windows XP
- Microsoft Picture It! Photo Premium 9

Poor compression increases the PNG file size but does not affect the image quality or compatibility of the file with other programs.

When the color depth of a truecolor image is reduced to an 8-bit palette (as in GIF), the resulting image data is typically much smaller. Thus a truecolor PNG will typically be larger than a color-reduced GIF, although PNG could store the color-reduced version as a palettized file of comparable size. Conversely, some tools, when saving images as PNGs, automatically save them as truecolor, even if the original data use only 8-bit color, thus bloating the file unnecessarily.[28] Both factors can lead to the misconception that PNG files are larger than equivalent GIF files.

6.4 Optimizing tools

Various tools are available for optimizing PNG files; they do this by:

- (optionally) removing ancillary chunks,
- reducing color depth, either:
  - use a palette (instead of RGB) if the image has 256 or fewer colors,
  - use a smaller palette, if the image has 2, 4, or 16 colors, or
  - (optionally) lossily discard some of the data in the original image,
- optimizing line-by-line filter choice, and
- optimizing DEFLATE compression.

6.4.1 Tool list

- pngcrush is the oldest of the popular PNG optimizers. It allows for multiple trials on filter selection and compression arguments, and finally choose the smallest one. This working model is used in almost every png optimizer.

- OptiPNG was inspired by pngcrush, but iterates over a wider range of compression parameters and performs trials in-memory for faster execution.[51] The main purpose of OptiPNG is to reduce the size of the PNG IDAT data stream by trying various filtering and compression methods. It also performs automatic bit depth, color type and color palette reduction where possible, and can correct some data integrity errors in input files. (pngcrush has the ability to do color reduction in a later version.)

- Advpng from package AdvanceCOMP was made to use 7-zip's deflater (which is slower but have smaller output than zlib), to optimize png files. However, since PNG is filtered before deflate compression, while advpng uses filter 0 globally (in other words it only uses unfiltered data), it's not a good consideration for png optimization. (In most scenarios, filtering helps more than a good deflater.)
  Advdef from the same package, however, is able to recompress the zlib stream, acting as a re-deflater.

- pngout was made with the author's own deflater (same to the author's zip utility, kzip), while keeps all facilities of color reduction / filtering. However, pngout doesn't allow for using several trials on filters in a single run, so it's suggested to use its commercial GUI version, pngoutwin, or used with a wrapper to automates the trials or to recompress using its own deflater while keep the filter line by line.[note 2]

- zopflipng was also made with a self-own deflater, zopfli. It has all the optimizing features optipng/pngcrush have (including automating trials) while providing a good deflater.

A simple comparison of their features is listed below.

Before zopflipng was available, a good way in practice to perform a png optimization is to use a combination of 2 tools in sequence for optimal compression: one which optimizes filters (and removes ancillary chunks), and one which optimizes DEFLATE. Although pngout offers both, only one type of filter can be specified in a single run, therefore it can be used with a wrapper tool or in combination with optipng or pngcrush,[note 2] acting as a re-deflater, like advdef.

6.4.2 Ancillary chunk removal

For removing ancillary chunks, most PNG optimization tools have the ability to remove all color correction data from PNG files (gamma, white balance, ICC color profile, standard RGB color profile). This often results in much smaller file sizes. For example, the following command line options achieve this with pngcrush:

pngcrush -rem gAMA -rem cHRM -rem iCCP -rem sRGB *InputFile.png OutputFile.png*

Ancillary chunks can also be losslessly removed using the free Win32 program PNGExtra.

6.4.3 Filter optimization

OptiPNG, pngcrush, pngout, and zopflipng all offer options applying one of the filter types 0–4 globally (using the same filter type for all lines) or with a “pseudo filter” (numbered 5), which for each line chooses one of the filter types 0–4 using an adaptive algorithm. Zopflipng offers 3 different adaptive method, including a brute-force search that attempts to optimize the filtering.[note 8]

pngout and zopflipng provide an option to preserve/reuse[note 2][note 9] the line-by-line filter set present in the input image.

OptiPNG, pngcrush and zopflipng provide options to try different filter strategies in a single run and choose the best. The freeware command line version of pngout doesn't offer this, but the commercial version, pngoutwin, does.[note 10]

6.4.4 DEFLATE optimization

AdvanceCOMP advdef, advpng, Ken Silverman's PNGOUT and zopflipng employ DEFLATE compression algorithms that are more exhaustive and produce smaller files than the zlib reference implementation used by the other compressors.

advpng doesn't have an option to apply filters and always uses filter 0 globally (leaving the image data unfiltered); therefore it should not be used where the image benefits significantly from filtering. By contrast, advdef from the same package doesn't deal with PNG structure and acts only as a re-deflater, retaining any existing filter settings.

6.4.5 Wrapper tools

Most wrapper tools take several passes using different optimizers and then select the smallest file.

Wrapper tools that simplify this workflow include: ImageOptim, a GUI front-end for Mac OS X; Kashmir Web Optimizer- GUI front-end for Windows; pngoptim a cmd batch script for Windows; imgopt, a command-line shell script that also losslessly optimizes JPEG images, Smush.it, an image-optimizing web service; TinyPNG, which provides compression by reducing the number of colors in the image automatically, but preserving alpha transparency; and Compress PNG that allows users to pick the number of colors that should be used.

The littleutils is another open-source package, containing a wrapper script called opt-png that uses pngcrush and a variant of pngrewrite to reduce bit-depth when possible. Perl scripts might wish to employ Image-Pngslimmer which allows some dynamic optimization.

The current version of IrfanView can use PNGOUT as an external plug-in, obviating the need for a separate compressor.

An open source Windows program called FileOptimizer losslessly optimizes many filetypes, including PNG. It runs multiple PNG optimization programs: advpng, apngopt, optipng, PngOptimizer, pngout, pngrewrite, and pngwolf.

Another open source Windows tool, pngoptim, uses a more opinionated approach, utilizing only pngout and zopflipng instead of taking the brute force approach of trying multiple optimizers.

6.5 Icon optimization

Since icons intended for Windows Vista and later versions may contain PNG subimages, the optimizations can be applied to them as well. At least one icon editor, Pixelformer, is able to perform a special optimization pass while saving ICO files, thereby reducing their sizes. FileOptimizer (mentioned above) can also handle ICO files.

Icons for Mac OS X may also contain PNG subimages, yet there isn't such tool available.

7 See also

- Computer graphics, including:
    - Comparison of layout engines (graphics)
- Image editing
- Image file formats
- Related graphics file formats

- APNG Animated PNG
- JPEG Network Graphics (JNG)
- Multiple-image Network Graphics (MNG)

- Similar file formats
  - X PixMap for portable icons
- Scalable Vector Graphics
- WebP

8 Notes

[1] The filtering is used to increase the similarity to the data, hence increasing the compression ratio. However, there is theoretically no formula for similarity, nor absolute relationship between the similarity and compressor, thus unless the compression is done, one can't tell one filter set is better than another.

[2] Use pngout -f6 to reuse previous filter set

[3] The tools offering such feature could act as a pure re-deflater to PNG files.

[4] The reference deflater implementation, zlib, is not good enough. See Page Zopfli, zip format in 7-zip and pngout

[5] Not only advpng doesn't support color reduction, it also fails with the images with a reduced colorspace

[6] Advpng can only apply filter 0 globally, thus it's neither yes or no, but N/A.

[7] Advdef only works to inflate the deflated data, and re-deflate it

[8] [optipng|pngcrush|pngout] -f OR zopflipng --filters

[9] zopflipng --filters=p

[10] Pngoutwin's setting dialog for optimization offers the user a selection of filter strategies.

9 References

[1] "ISO/IEC 15948:2004 - Information technology -- Computer graphics and image processing -- Portable Network Graphics (PNG): Functional specification". Retrieved 2011-02-19.

[2] "History of PNG". Libpng.org. 29 May 2010. Retrieved 2010-10-20.

[3] "IEC standard (scope)". 10 November 2003.

[4] Matthias Gelbmann (January 31, 2013). "The PNG image file format is now more popular than GIF". *W3Techs*. Q-Success. Retrieved March 22, 2013. PNG is now used on 62.4% of all websites, just ahead of GIF with 62.3%.

[5] IANA.org

[6] TBH View profile More options (6 January 1995). "Thoughts on a GIF-replacement file format". Groups.google.com. Retrieved 2010-10-20.

[7] "PNG standard, section 8.4". PNG itself is strictly a single-image format. (...) In the future, a multiple-image format based on PNG may be defined. Such a format will be considered a separate file format

[8] Thomas Boutell (1 March 1997). "PNG (Portable Network Graphics) Specification 1.0".

[9] "PNG (Portable Network Graphics) Specification, Version 1.1–12. Appendix: Rationale". Libpng.org. Retrieved 2010-10-20.

[10] Glenn Randers-Pehrson & Thomas Boutell (editors) (1999). "Chunk Specifications". *PNG (Portable Network Graphics) Specification, Version 1.2*. Massachusetts Institute of Technology (MIT). Retrieved 30 Jan 2011.

[11] "Portable Network Graphics (PNG) Specification (Second Edition)". W3.org. Retrieved 2013-05-01.

[12] "Portable Network Graphics (PNG) Specification (Second Edition) Information technology — Computer graphics and image processing — Portable Network Graphics (PNG): Functional specification. ISO/IEC 15948:2003 (E) W3C Recommendation 10 November 2003".

[13] "PNG News from 2006". Libpng.org.

[14] Portable Network Graphics (PNG) Specification (Second Edition): 11.2.2 IHDR Image header.

[15] "PNG Specification: Rationale". *w3.org*.

[16] "Portable Network Graphics (PNG) Specification (Second Edition): 9 Filtering". W3.org. Retrieved 2010-10-20.

[17] "Filter Algorithms". *PNG Specification*.

[18] Paeth, A.W., "Image File Compression Made Easy", in Graphics Gems II, James Arvo, editor. Academic Press, San Diego, 1991. ISBN 0-12-064480-0.

[19] Crocker, Lee Daniel (July 1995). "PNG: The Portable Network Graphic Format". *Dr. Dobb's Journal* **20** (232): 36–44.

[20] "Introduction to PNG". nuwen.net. Retrieved 2010-10-20.

[21] "Opera Desktop Team: Post-Alpha Opera 9.5 Release". My.opera.com. Retrieved 2010-10-20.

[22] "iOS 8 and iPhone 6 for web developers and designers: next evolution for Safari and native webapps". mobilexweb.com. 2014-09-17. Retrieved 2014-09-24.

[23] "Vote failed: APNG 20070405a". 20 April 2007.

[24] "PNG Group animation proposal comparison + test-software". *xs4all.nl*. Archived from the original on 24 January 2009.

[25] "A Basic Introduction to PNG Features". Libpng.org. Retrieved 2010-10-20.

[26] “GIF, PNG, JPG. Which One To Use?". Sitepoint.com. 3 August 2009. Retrieved 2010-10-20.

[27] “T.87 : Lossless and near-lossless compression of continuous-tone still images - Baseline”. International Telecommunication Union. Retrieved 20 March 2011.

[28] Chapter 9. Compression and Filtering, in *PNG: The Definitive Guide* by Greg Roelofs.

[29] “libpng”. Retrieved 2013-07-13.

[30] “Use of PNG Images to Display Data”. Oregon Water Science Center. 16 February 2006.

[31] “Why There Are No GIF files on GNU Web Pages”. *GNU Operating System*. 16 December 2008.

[32] “PNG Fact Sheet”. World Wide Web Consortium. 7 October 1996.

[33] “Resource page for gif2png 2.5.11”. *catb.org*.

[34] “Burn All GIFs”.

[35] “PNG Transparency in Internet Explorer”. PC Magazine. 5 October 2004.

[36] “Browsers with PNG Support”. 14 March 2009.

[37] “Windows Explorer Crashes When I Click on a Fireworks PNG File to View It”. Adobe Systems. 5 June 2007.

[38] “Unable to view .png images with Internet Explorer 4.0”. *Microsoft Knowledge Base*.

[39] “PNG Graphics That Are Inside of an Object Tag Print as a Negative Image”. *Microsoft Knowledge Base*.

[40] “PNG Images Are Printed Improperly in Internet Explorer 5.01”. *Microsoft Knowledge Base*.

[41] “You cannot view some PNG images in Internet Explorer 6”. *Microsoft Knowledge Base*.

[42] “You cannot use Internet Explorer 6 to open a PNG file that contains one or more zero-length IDAT chunks”. *Microsoft Knowledge Base*.

[43] “PNG Frequently Asked Questions”.

[44] “PhD: Portable Network Graphics Lose Transparency in Web Browser”. *Microsoft Knowledge Base*.

[45] “PNG Files Do Not Show Transparency in Internet Explorer”. *Microsoft Knowledge Base*.

[46] Lovitt, Michael (21 December 2002). “Cross-Browser Variable Opacity with PNG: A Real Solution”. *A List Apart*.

[47] “IE7 alpha transparent PNG + opacity”. *Channel 9*.

[48] Fulbright, Michael (1999). “GNOME 1.0 Library Roadmap”.

[49] “Windows Vista - Icons”. *OOne*. 2007. Retrieved 2007-11-12.

[50] “PNG can be a lossy format”. Pngmini.com. Retrieved 2014-02-01.

[51] Truţa, Cosmin. “A guide to PNG optimization”.

10 Further reading

- Roelofs, Greg (April 1997). “Linux Gazette: History of the Portable Network Graphics (PNG) Format”. *Linux Journal* (Specialized Systems Consultants, Inc.) **1997** (36es). ISSN 1075-3583.
- Roelofs, Greg (2003). *PNG: The Definitive Guide* (2nd ed.). O'Reilly Media. ISBN 1-56592-542-4.

11 External links

11.1 libpng.org

- PNG Home Site
- libpng Home Page
- *The Story of PNG* by Greg Roelofs

11.2 W3C

- W3 PNG Specification
- Test inline PNG images

11.3 Others

- RFC 2083
- More information about PNG color correction
- The GD-library to generate dynamic PNG-files with PHP
- A guide to PNG optimization
- PNG Adam7 interlacing
- Encoding Web Shells in PNG files: Encoding human readable data inside an IDAT block.

12 Text and image sources, contributors, and licenses

12.1 Text

12.2 Images

- **File:PNG-Gradient.png** *Source:* https://upload.wikimedia.org/wikipedia/commons/8/89/PNG-Gradient.png *License:* Public domain *Contributors:* ? *Original artist:* ?
- **File:PNG-Gradient_hex.png** *Source:* https://upload.wikimedia.org/wikipedia/commons/1/17/PNG-Gradient_hex.png *License:* GPL *Contributors:* Own work *Original artist:* Authors of the Bless Hex Editor
- **File:PNG_transparency_demonstration_1.png** *Source:* https://upload.wikimedia.org/wikipedia/commons/4/47/PNG_transparency_demonstration_1.png *License:* CC BY-SA 3.0 *Contributors:* Own work: Rendered in POV-Ray by user:ed_g2s. *Original artist:* POV-Ray source code
- **File:Pixel-prediction.svg** *Source:* https://upload.wikimedia.org/wikipedia/commons/3/39/Pixel-prediction.svg *License:* Public domain *Contributors:* Own work *Original artist:* ZooFari
- **File:Symbol_neutral_vote.svg** *Source:* https://upload.wikimedia.org/wikipedia/en/8/89/Symbol_neutral_vote.svg *License:* Public domain *Contributors:* ? *Original artist:* ?

12.3 Content license

- Creative Commons Attribution-Share Alike 3.0