JNG (JPEG Network Graphics) Format Version 0.998b

For list of authors, see Credits (Chapter 6).

Status of this Memo

This is a **DRAFT proposal**. Some version of this document will become version 1.00.

Comments on this document can be sent to the MNG specification maintainers at one of the following addresses:

- mng-list@ccrc.wustl.edu
- png-group@w3.org
- png-info@uunet.uu.net

Distribution of this memo is unlimited.

At present, the latest version of this document is available on the World Wide Web from

```
ftp://swrinde.nde.swri.edu/pub/mng/documents/.
```

In the case of any discrepancy between this extract and the full MNG specification, the full MNG specification shall take precedence.

Changes from Version 0.998

- Added hexadecimal and C notation versions of the signature.

Abstract

This document presents the format of a JNG (JPEG Network Graphics) datastream. JNG is a lossy single-image member of the PNG (Portable Network Graphics) format family. It encapsulates a JPEG datastream in PNG-style chunks, along with an optional alpha channel and ancillary chunks that carry color-space information and comments. While JNG is primarily intended as a subformat of the MNG (Multiple-image Network Graphics) format, standalone JNG files are also possible.

1 The JPEG Network Graphics (JNG) Format

JNG (JPEG Network Graphics) is the lossy sub-format for MNG objects.

MNG-LC and MNG-VLC applications can choose to support JNG or not. Those that do not can check bit 4 (JNG is present/absent) of the MHDR simplicity profile to decide whether they can process the datastream.

Note: This specification depends on the PNG Portable Network Graphics specification [PNG]. The PNG specification is available at the PNG home page,

```
http://www.libpng.org/pub/png/
```

A JNG datastream consists of a header chunk (JHDR), JDAT chunks that contain a complete JPEG datastream, optional IDAT chunks that contain a PNG-encoded grayscale image that is to be used as an alpha mask, and an IEND chunk. The alpha mask, if present, must have the same dimensions as the image itself. The JDAT and IDAT chunks can be interleaved. Some of the PNG ancillary chunks are also recognized in JNG datastreams.

While JNG is primarily intended for use as a sub-format within MNG, a single-image JNG datastream can be written in a standalone file. If so, the JNG datastream begins with an 8-byte signature containing

```
 139  74  78  71  13  10  26  10  (decimal)
  91  4a  4e  47  0d  0a  1a  0a  (hexadecimal)
\213   J   N   G  \r  \n \32  \n  (ASCII C notation)
```

which is similar to the PNG signature with “\213 J N G” instead of “\211 P N G” in bytes 0–3.

We may at some future time register an Internet Media Type for JNG files. Until then, the interim media type `image/x-jng` can be used. It is recommended that the file extension “.jng” (lower case preferred) be used.

JNG is pronounced “Jing.”

1.1 Critical JNG chunks

This section specifies the critical chunks that are defined in the JNG format.

1.1.1 JHDR JNG header

The format of the JHDR chunk introduces a JNG datastream. It contains:

```
Width:      4 bytes (unsigned integer, range 0..65535).
Height:     4 bytes (unsigned integer, range 0..65535).
Color_type: 1 byte
              8: Gray (Y).
             10: Color (YCbCr).
             12: Gray-alpha (Y-alpha).
             14: Color-alpha (YCbCr-alpha).

Image_sample_depth:
            1 byte
              8: 8-bit samples and quantization tables.
             12: 12-bit samples and quantization tables.
             20: 8-bit image followed by a 12-bit image.

Image_compression_method:
            1 byte
              8: ISO-10918-1 Huffman-coded baseline JPEG.

Image_interlace_method:
            1 byte.
              0: Sequential JPEG, single scan.
              8: Progressive JPEG.

Alpha_sample_depth:
            1 byte.
              0, 1, 2, 4, 8, or 16, if the Alpha compression method is 0
                (PNG)
              8, if the Alpha compression method is 8 (JNG).

Alpha_compression_method:
            1 byte.
              0: PNG grayscale IDAT format.
              8: JNG 8-bit grayscale JDAA format.

Alpha_filter_method:
            1 byte.
              0: Adaptive PNG (see PNG spec) or not applicable (JPEG).

Alpha_interlace_method:
            1 byte.
              0: Noninterlaced PNG or sequential single-scan JPEG.
```

The width, height, image_sample_depth, image_compression_method, and image_interlace_method fields are redundant because equivalent information is also embedded in the JDAT datastream. They appear in the JHDR chunk for convenience. Their values must be identical to their equivalents embedded in the JDAT chunk. We use four bytes in the width and height fields for similarity to MNG and PNG, and to leave room for future expansion, even though two bytes would have been sufficient.

When the `color_type` is 8 or 10 (no alpha channel), the last four bytes, which describe the IDAT or JDAA data, must be set to zero. The `alpha_sample_depth` must be nonzero when the alpha channel is present.

1.1.2 JDAT JNG image data

A JNG datastream must contain one or more JDAT chunks, whose data, when concatenated, forms a complete JNG JPEG datastream. JNG decoders are required to read all baseline JNG JPEG and eight-bit progressive JNG JPEG datastreams. Twelve-bit capability is not required.

JDAT chunks are like PNG IDAT chunks in that there may be multiple JDAT chunks, the data from which are concatenated to form a single datastream that can be sent to the decompressor. No chunks are permitted among the sequence of JDAT chunks, except for interleaved IDAT chunks. The ordering requirements of other ancillary chunks are the same with respect to JDAT as they are in PNG with respect to the IDAT chunk.

A JNG JPEG is a baseline, extended-sequential, or progressive JPEG as defined by JPEG Part 1 [ISO/IEC-10918-1]. JNG uses only JFIF-compatible [JFIF] component interpretations, and imposes a few additional restrictions that reflect limitations of many existing JPEG implementations. In particular, only Huffman entropy coding is permitted.

Actually, a JNG may contain two separate JNG JPEG datastreams (one eight-bit and one twelve-bit), each contained in a series of JDAT chunks, and separated by a JSEP chunk (see the JSEP chunk specification below, Paragraph 1.1.6). Decoders that are unable to (or do not wish to) handle twelve-bit datastreams are allowed to display the eight-bit datastream instead, if one is present.

The core of the JNG JPEG definition is baseline JNG JPEG, which is JPEG Part 1's definition of baseline JPEG further restricted by JFIF restrictions and JNG-specific restrictions. JNG JPEG also includes progressive JPEG, which is also defined in JPEG Part 1 and has JNG-specific restrictions.

- Baseline JNG JPEG restrictions

  A baseline JPEG according to JPEG Part 1 is DCT-based (lossy) sequential JPEG, using 8-bit sample precision and Huffman entropy coding, with the following further restrictions:

  - Quantization table precision must be 8 bits for baseline JPEG.
  - Huffman code tables can have table numbers 0 and 1 only.

  The SOF marker type for baseline JPEG is SOF0.

  JDAT datastreams must always follow "interchange JPEG" rules: all necessary quantization and Huffman tables must be included in the datastream; no tables can be omitted.

- JFIF-compatible restrictions

  The image data is always stored left-to-right, top-to-bottom.

  The encoded data shall have one of the two color space interpretations allowed by the JFIF specification:

  - Grayscale: a single component representing luminance, ranging from 0 for black to 255 for white (or 0 to 4095 when dealing with twelve-bit data). This component shall have JPEG component identifier 1.
  - YCbCr: three components representing luminance, chroma blue, and chroma red, in that order. The components shall be assigned JPEG component identifiers 1, 2, 3 respectively. YCbCr is defined as a linear transformation from RGB color space:

    ```
    Y = Luma_red*R + Luma_green*G + Luma_blue*B
    Cb = (B - Y) / (2 - 2*Luma_blue) + Half_scale
    Cr = (R - Y) / (2 - 2*Luma_red)  + Half_scale
    ```

    By convention, the luminance coefficients are always those defined by CCIR Recommendation 601-1:

    ```
    Luma_red   = 0.299
    Luma_green = 0.587
    Luma_blue  = 0.114
    ```

    The constant Half_scale is 128 when dealing with eight-bit data, 2048 for twelve-bit data. With these equations, Y, Cb, and Cr all have the same range as R, G, and B: 0 to 255 for eight-bit data, 0 to 4095 for twelve-bit data.

    The JFIF convention for YCbCr differs from typical digital television practice in that no headroom/footroom is reserved: the coefficient values range over the full available 8 or 12 bits.

    Intercomponent sample alignment shall be such that the first (upper leftmost) samples of each component share a common upper left corner position. This again differs from common digital TV practice, in which the first samples share a common center position. The JFIF convention is simpler to visualize: subsampled chroma samples always cover an integral number of luminance sample positions, whereas with co-centered alignment, chroma samples only partially overlap some luminance samples.

- Additional JNG restrictions

  JNG imposes three additional restrictions not found in the text of either JPEG Part 1 or the JFIF specification:

  - The sampling factors for YCbCr images must be one of these sets:
    - 1h1v,1h1v,1h1v (also called 4:4:4 or 1x1 sampling)
    - 2h1v,1h1v,1h1v (also called 4:2:2 or 2x1 sampling)
    - 2h2v,1h1v,1h1v (also called 4:2:0 or 2x2 sampling)
    - 1h2v,1h1v,1h1v (also called 1x2 sampling)

    In other words, the chroma components may be downsampled 2:1 or 1:2 horizontally or vertically relative to luminance, or they may be left full size. These four sampling ratios are the only ones supported by a wide spectrum of implementations (1x2 is relatively uncommon, and is usually the result of a lossless rotation of a 2x1 sampling).

    For grayscale images, the sampling factors are irrelevant according to a strict reading of JPEG Part 1. Hence decoder authors should accept any sampling factors for grayscale. However, we recommend that encoders always emit sampling factors 1h1v for grayscale, since some decoders have been observed to malfunction when presented with other sampling factors.

- There must be only one scan in an image: that is, YCbCr images must be fully interleaved. There is little advantage to be gained by encoding a baseline image in multiple scans, and many baseline decoders do not support multiple scans at all.
- The DNL (Define Number of Lines) marker is prohibited. The image height must always be specified accurately in the SOFn marker and in the JHDR chunk.

- Recommended progressive JPEG subset

  For JNG progressive JPEG datastreams, the JPEG process is progressive Huffman coding (SOF marker type SOF2) rather than baseline (SOF0). All JNG-compliant decoders must support full progression, including both spectral-selection and successive-approximation modes, with any sequence of scan progression parameters allowed by the JPEG Part 1 standard.

  Otherwise, all the restrictions listed above apply, except these:

  - Multiple-scan support is obviously required for progressive JPEG.
  - Huffman table numbers up to 3 (the full JPEG limit) may be used, since the baseline two-table limit is unlikely to be needed by any decoder that can handle progressive JPEG.

  We require full progression support since relatively little code savings can be achieved by subsetting the JPEG progression features. In particular, successive approximation offers significant gains in the visual quality of early scans. Omitting successive-approximation support from a decoder does not save nearly enough code to justify restricting JNG progressive encoders to spectral selection only.

  No particular progressive scan sequence is specified or recommended by this specification. Not enough experience has been gained with progressive JPEG to warrant making such a recommendation. To allow for future experimentation with scan sequences, decoders are expected to handle any JPEG-legal sequence. Again, the code savings that might be had by making restrictive assumptions are too small to justify a limitation.

  When the JSEP chunk is present, both images must be progressive if one of them is progressive.

- Recommended 12-bit JPEG subset

  JNG JPEGs may optionally use 12-bit sample precision as defined in JPEG Part 1.

  For a sequential image, the SOF marker type must be SOF1 (extended sequential) not SOF0, and the baseline restriction of two Huffman tables is removed. Also, the encoder may use either 8-bit or 16-bit quantization tables. All other JNG baseline restrictions still apply. It is recommended that JNG encoders not use extended-sequential mode except to encode 12-bit data.

  For a progressive image, the only difference between 8-bit and 12-bit modes is that the sample precision is 12 bits and the encoder may use either 8-bit or 16-bit quantization tables. All other JNG restrictions still apply.

1.1.3 IDAT JNG PNG-encoded alpha data

This chunk is exactly like the IDAT chunk in a PNG grayscale image, except that it is interpreted as an alpha mask to be applied to the image data from the JDAT chunks, when `alpha_compression_method=0`. The alpha channel, if present, can have sample depths 1, 2, 4, 8, or 16.

The filter method can be any filter method that is defined for PNG datastreams that are embedded in MNG datastreams.

The IDAT chunks can be interleaved with the JDAT chunks (see Recommendations for Encoders: JNG interleaving below). No other chunk type can appear among the sequence of IDAT and JDAT chunks. No other chunk type can appear between the sequences of IDAT and JDAT chunks when they are not interleaved. The samples in the IDAT must be presented in noninterlaced order, left to right, top to bottom. As in PNG, zero means fully transparent and $2^{\mathrm{alpha\_sample\_depth}} - 1$ means fully opaque.

The IDAT chunks must precede the JSEP chunk, if the JSEP chunk is present. Minimal viewers that ignore the twelve-bit JDAT chunks must read the IDAT chunks and apply the alpha samples to the eight-bit image that is contained in the JDAT chunks that precede the JSEP chunk. Viewers that skip the eight-bit JDAT chunks must decode the IDAT chunks that precede the JSEP chunk and apply the alpha samples to the twelve-bit image that is contained in the JDAT chunks that follow the JSEP chunk.

1.1.4 JDAA JNG JPEG-encoded alpha data

This chunk is exactly like the JDAT chunk in a non-progressive JNG 8-bit grayscale image, except that it is interpreted as an alpha mask to be applied to the image data from the JDAT chunks, when `alpha_compression_method=8`. The alpha channel, if present, can have only sample depth 8. The JDAA chunks can be interleaved with the JDAT chunks (see Recommendations for Encoders: JNG interleaving below).

Like IDAT chunks, the JDAA chunks must precede the JSEP chunk, if the JSEP chunk is present, and are handled similarly.

1.1.5 IEND End of JNG datastream

The JNG IEND chunk is identical to its counterpart in PNG. Its data length is zero, and it serves to mark the end of the JNG datastream.

1.1.6 JSEP 8-bit/12-bit image separator

JNG permits storage of both an 8-bit and a 12-bit JPEG datastream in a single JNG file. This feature allows an 8-bit image to be provided for non-12-bit-capable decoders. The JSEP chunk is used to separate the two datastreams.

The JSEP chunk is empty.

A JSEP chunk must appear between the JDAT chunks of an eight-bit datastream and those of a twelve-bit datastream, when `image_sample_depth=20` in the JHDR chunk. When `image_sample_depth != 20`, the JSEP chunk must not be present. The eight-bit datastream must appear first. Both images must have the same width, height, color type, compression method, and interlace method. Viewers can choose to display one or the other image, but not both.

1.2 Ancillary JNG chunks

Some PNG ancillary chunks can also appear in JNG datastreams, and are used for the same purposes as described in the PNG specification [PNG] and the Extensions to the PNG Specification document [PNG-EXT].

If the bKGD chunk is present, it must be written as if it were written for a PNG datastream with sample_depth=8. It has one 2-byte entry for grayscale JNGs and three 2-byte entries for color JNGs. The first (most significant) byte of each entry must be 0.

The following chunks have exactly the same meaning and have the same syntax as given in the PNG specification: cHRM, gAMA, iCCP, sRGB, pHYs, oFFs, sCAL, iTXt, tEXt, tIME, and zTXt.

The PNG PLTE, hIST, pCAL, sBIT, sPLT, tRNS, fRAc, and gIF* chunks are not defined in JNG.

When cHRM, gAMA, iCCP, or sRGB are present, they provide information about the color space of the decoded JDAT image, and they have no effect on the decoded alpha samples from the IDAT or JDAA chunks. Any viewer that processes the gAMA chunk must also recognize and process the sRGB chunk. It can treat it as if it were a gAMA chunk containing the value .45455 and it can ignore its “intent” field.

The chunk copying and ordering rules for JNG are the same as those in PNG, except for the fact that the JDAT chunks and IDAT or JDAA chunks can be interleaved.

2 Recommendations for Encoders

The following recommendations do not form a part of the specification.

2.1 Interleaving JDAT, JDAA, and IDAT chunks

When a JNG datastream contains an alpha channel, and the file is intended for transmission over a network, it is useful to interleave the IDAT or JDAA and the JDAT chunks. In the case of sequential JPEG, the interleaving should be arranged so that the alpha data arrives more or less in sync with the color data for the scanlines. In the case of progressive JPEG, the alpha data should be interleaved with the first JPEG pass, so that *all* of the alpha data has arrived before the beginning of the second JPEG pass.

2.2 Use of the JDAA chunk

It is recommended that the JDAA chunk be used only to convey smoothly varying alpha channels and not to convey binary transparency which is more precisely and efficiently conveyed in IDAT chunks.

3 Revision History

3.1 Version 0.998b

Proposed 20 January 2001

- None.

3.2 Version 0.998a

Proposed 19 January 2001

- None.

3.3 Version 0.998

Released 18 January 2001

3.4 Version 0.997

Released 10 January 2001

3.5 Version 0.995a

Proposed 23 December 2000

Proposes some changes to the simplicity profile.

3.6 Version 0.99

Released 10 December 2000

3.7 Version 0.98

Released 01 October 2000

- Disallowed the JSEP chunk when `image_sample_depth != 20`

3.8 Version 0.97

Released 28 February 2000.

- Minor editorial changes only.

3.9 Version 0.96

Released 18 July 1999.

- Changed "`JDAT_sample_depth`" to "`image_sample_depth`" and "`IDAT_sample_depth`" to "`alpha_sample_depth`", etc.
- Added Revision History, References, and Security Considerations sections.

3.10 Version 0.95

- Initial public release, approved by the PNG Development Group on 11 May 1999.

4 References

[ISO/IEC-10918-1]
International Organization for Standardization and International Electrotechnical Commission, "Digital Compression and Coding of Continuous-tone Still Images, Part 1: Requirements and guidelines" ISO/IEC IS 10918-1, ITU-T T.81.
See also Pennebaker, William B., and Joan L. Mitchell, "JPEG : Still Image Data Compression Standard" Van Nostrand Reinhold, ISBN:0442012721, September 1992

[JFIF]
C-Cube Microsystems, "JPEG File Interchange Format, Version 1.02", September 1992.

[PNG]
Boutell, T., et. al., "PNG (Portable Network Graphics Format) Version 1.0", RFC 2083,
ftp://ftp.isi.edu/in-notes/rfc2083.txt also available at
ftp://swrinde.nde.swri.edu/pub/png/documents/. This specification has also been published as a W3C

Recommendation, which is available at
http://www.w3.org/TR/REC-png.html.

See also the PNG-1.2 specification:
Randers-Pehrson, G., et. al., “PNG (Portable Network Graphics Format) Version 1.2”, which is available at
ftp://swrinde.nde.swri.edu/pub/png/documents/.

[PNG-EXT]
Randers-Pehrson, G., et al, “Extensions to the PNG 1.2 Specification”,
ftp://swrinde.nde.swri.edu/pub/png/documents/pngext-*.

5 Security Considerations

Security considerations are addressed in the PNG specification.

No known additional security concerns are raised by this format.

6 Credits

Editor

- Glenn Randers-Pehrson, randeg @ alum.rpi.edu

Contributors

Contributors’ names are presented in alphabetical order:

- Gerard Juyn, gjuyn @ xs4all.nl
- Tom Lane, tgl @ sss.pgh.pa.us
- Glenn Randers-Pehrson, randeg @ alum.rpi.edu
- Nancy M. Randers-Pehrson, randeg @ alum.rpi.edu
- Greg Roelofs, newt @ pobox.com

Document source

This document was built from the file `mng-master-20010120` on 20 January 2001.

Copyright Notice

End of JNG Specification.