INTERNATIONAL STANDARD

**ISO/IEC 21122-5**

Second edition
2022-09

# Information technology — JPEG XS low-latency lightweight image coding system —

Part 5:
**Reference software**

*Technologies de l'information — Système de codage d'images léger à faible latence JPEG XS —*

*Partie 5: Logiciel de référence*

Reference number
ISO/IEC 21122-5:2022(E)

# Contents

Page

# Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives or www.iec.ch/members_experts/refdocs).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents) or the IEC list of patent declarations received (see https://patents.iec.ch).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see www.iso.org/iso/foreword.html. In the IEC, see www.iec.ch/understanding-standards.

This document was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*.

This second edition cancels and replaces the first edition (ISO/IEC 21122-5:2020), which has been technically revised.

The main changes are as follows:

— Updated reference software.

A list of all parts in the ISO/IEC 21122 series can be found on the ISO and IEC websites.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html and www.iec.ch/national-committees.

.

# Introduction

This document is part of a series of standards for a low-latency lightweight image coding system, denoted JPEG XS.

It provides the software reference implementation of the ISO/IEC 21122-1 and ISO/IEC 21122-2 standards. It has been successfully compiled and tested on various operating systems at the time of writing. It is important to note that this reference software implementation represents just one way of implementing JPEG XS. This implementation can serve as a validation anchor to other implementations.

No guarantee of the quality that will be achieved by an encoder is provided by its conformance to ISO/IEC 21122-1 and ISO/IEC 21122-2, as the conformance is only defined in terms of specific constraints imposed on the syntax of the generated codestream. In particular, while sample encoder software implementations may suffice to provide some illustrative examples of which quality can be achieved within the ISO/IEC 21122 series, they provide neither an assurance of minimum guaranteed image encoding quality nor maximum achievable image encoding quality.

Similarly, the computation resource characteristics in terms of program or data memory usage, execution speed, etc. of sample software encoder or decoder implementations should not be construed as a representative of the typical, minimal or maximal computational resource characteristics to be exhibited by implementations of the ISO/IEC 21122 series.

The International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC) draw attention to the fact that it is claimed that compliance with this document may involve the use of a patent.

ISO and IEC take no position concerning the evidence, validity and scope of this patent right.

The holder of this patent right has assured ISO and IEC that he/she is willing to negotiate licences under reasonable and non-discriminatory terms and conditions with applicants throughout the world. In this respect, the statement of the holder of this patent right is registered with ISO and IEC. Information may be obtained from the patent database available at www.iso.org/patents or https://patents.iec.ch.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights other than those in the patent database. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

# Information technology — JPEG XS low-latency lightweight image coding system —

# Part 5: Reference software

## 1 Scope

This document contains the reference software of the ISO/IEC 21122 series. It acts as a guideline for implementation and as a reference for conformance testing.

## 2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 21122-1, *Information technology — JPEG XS low-latency lightweight image coding system — Part 1: Core coding system*

ISO/IEC 21122-2, *Information technology — JPEG XS low-latency lightweight image coding system — Part 2: Profiles and buffer models*

## 3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC 21122-1 and ISO/IEC 21122-2 apply.

ISO and IEC maintain terminology databases for use in standardization at the following addresses:

— ISO Online browsing platform: available at https://www.iso.org/obp

— IEC Electropedia: available at https://www.electropedia.org/

## 4 Reference software

### 4.1 General

In addition to the main text, including Annexes A to C, the reference software is available at https://standards.iso.org/iso-iec/21122/-5/ed-2/en/.

### 4.2 Purpose

The purpose of this document is to provide the following:

— Provide a reference decoder software implementation, capable of decoding codestreams that conform to ISO/IEC 21122-1, or to ISO/IEC 21122-1 and ISO/IEC 21122-2.

— Provide a sample encoder software implementation, capable of producing codestreams that conform to ISO/IEC 21122-1, or to ISO/IEC 21122-1 and ISO/IEC 21122-2.

The use of this reference software is not required for making an implementation of an encoder or decoder in conformance to any of the parts of the ISO/IEC 21122 series. Requirements established in ISO/IEC 21122-1 and ISO/IEC 21122-2 take precedence over the behaviour of the reference software.

The reference decoder software generates the normative output of the decoding process specified by ISO/IEC 21122-1. The output of the reference decoder satisfies the requirements of the strict conformance point of ISO/IEC 21122-4. It therefore allows conformance testing by means of comparing the output of a candidate implementation by that of the reference software specified in this document. Conformance testing is specified in ISO/IEC 21122-4. The reference decoder software can also be used to validate the syntactic correctness of a codestream by attempting to decode a candidate codestream with the reference software.

### 4.3 Examples of use

Some examples of use for the reference decoder software implementations are as follows:

— As an illustration of how to perform the decoding processes specified in ISO/IEC 21122-1.

— As the starting basis for the implementation of a decoder that conforms to ISO/IEC 21122-1.

— For testing the conformance of a decoder implementation to ISO/IEC 21122-1 with the procedures specified in ISO/IEC 21122-4. Details on reference testing can be found in ISO/IEC 21122-4.

— For (non-exhaustive) testing of the conformance of a codestream (or file) to the constraints specified in ISO/IEC 21122-1.

The lack of detection of any conformance violation by any reference software implementation should not be considered as a definite proof that the codestream under testing conforms to all constraints required for it be conforming to ISO/IEC 21122-1 or ISO/IEC 21122-2. In fact, ISO/IEC 21122-4 defines additional testing procedures that can be used to test a codestream for conformance to ISO/IEC 21122-1 and ISO/IEC 21122-2, such as for example testing the conformance to the buffer models specified there.

Some examples of use for a sample encoder software are as follows:

— As an illustration of how to implement an encoding process that produces codestreams that are conforming to ISO/IEC 21122-1 and ISO/IEC 21122-2.

— As starting point for an implementation of an encoder that conforms to ISO/IEC 21122-1 and ISO/IEC 21122-2.

— As a means of generating codestreams conforming to ISO/IEC 21122-1 and ISO/IEC 21122-2 for testing purposes.

— As a means of demonstrating and evaluating examples of the quality that can be achieved by an encoding process that conforms to ISO/IEC 21122-1 and ISO/IEC 21122-2.

It is not recommended to use the reference software in a production environment, as it lacks proper protection mechanisms against memory corruption and code execution exploits.

## 5 Copyright, licensing and intellectual property

The reference software was originally developed by the parties indicated in the file LICENCE.md within the package forming a part of this document, in the course of development of ISO/IEC 21122-5. ISO/IEC draws the attention of the users of this software to the license terms and conditions specified in the same LICENSE.md file.

# Annex A
(informative)

# Building the reference software

## A.1 Unpacking the software

The reference software is provided in the form of a ZIP archive named `xs_ref_sw_ed2.zip`. Unpacking the archive is operating system specific, but on most POSIX compliant operating systems this can be done by opening a command line terminal and issuing the following:

```
unzip xs_ref_sw_ed2.zip
```

This will unpack all files in the archive into the current directory.

## A.2 Building the software

The source code is written in C11 and is designed to run on most modern operating systems. The exact tools and versions depend on the operating system and the compiler to be used. A README.md file is included with the software describing the steps to build the software.

# Annex B
(informative)

# Image file formats read and written by the reference software

## B.1 General

The reference software supports a wide variety of image file formats, used as input for the encoder and as output for the decoder, that are well established and easy to use, understand, and implement. This annex describes the three most important image file formats.

## B.2 PGX

### B.2.1 General

This clause describes the PGX image file format, supported by the reference software and used by ISO/IEC 21122-4 for the decoded reference images. This file format is capable of representing any input or output image that is supported by JPEG XS (1 or more components and 1 to 16 bits per component value).

The format consists of a directory file, one header file per component, and one raw data file per component.

NOTE The `difftest_ng` tool (see ISO/IEC 21122-4) can convert between PGX and many other image file formats.

### B.2.2 Directory file

The directory file is the file that is provided as input file to the comparison tool (PGX file extension). It consists, for each component, of the file name of the raw data file encoded in ASCII (see Reference [1]), relative to the path where the directory file is located. Each raw data file name is terminated by a single line feed character (ASCII 10=0x0a).

NOTE The PGX directory file does not separate lines by CR/LF pairs, i.e. the ASCII 13=0x0d, 10=0x0a code sequence.

### B.2.3 Header file

The header file is derived from the file name of the raw data file listed in the directory file by removing the postfix `.raw` and replacing it by the postfix `.h`. It describes the format in which the raw format is encoded. There is one separate header file per component.

Each header file consists of a single line, terminated by a single Line Feed character (ASCII 10=0x0a) describing the format. It consists of the following fields, where angle brackets `< >` indicate parameters described below and `SPC` indicates a blank space (ASCII 32=0x20):

```
P<dataformat>SPC<endian>SPC<signed><precision>SPC<width>SPC<height>
```

where:

| | |
|---|---|
| `P` | identifies the header file and shall be present. It has no particular meaning beyond format identification. |

| | |
|---|---|
| `<dataformat>` | identifies the sample format. ISO/IEC 21122 stores integer samples only, which are indicated by the single character `G` as `dataformat`. |
| `<endian>` | identifies the endianness of the encoded data. The character sequence `ML` indicates big endian encoding, i.e. most significant byte first, the character sequence `LM` little endian encoding, i.e. least significant byte first. |
| `<signed>` | indicates whether sample values are signed or unsigned. ISO/IEC 21122 covers only unsigned samples, indicated by the character `+` in this field. |
| `<precision>` | indicates the bit depth of the sample values in the component described by this header file. The bit depth is represented as ASCII encoded decimal number. |
| `<width>` | is the number of samples per line for this component, represented as ASCII encoded decimal number. |
| `<height>` | is the number of lines of the image, represented as ASCII encoded decimal number. |

### B.2.4 Data file

Each data file contains the sample values themselves in raster scan order, left to right, top to bottom. If the precision of the component is 8 bits or below, each sample is represented in 8 bits, right aligned to the entire byte, i.e. unused bits remain 0 and make up the most-significant bits of the byte. If the precision of the component is larger than 8 bits, each sample is represented by two bytes, encoded in the order indicated by the header, i.e. either most significant byte first if the endianness field is `ML`, or with the least significant byte first if the endianness field is `LM`. The data bits are right-aligned into the two bytes, most significant bits remain 0 if necessary.

## B.3 NetPBM (PPM/PGM)

### B.3.1 General

The NetPBM format[2] is capable of carrying images of bit depths *P* between 1 and 16 bits per component, with either 1 or 3 components per sample. In the case of single component (i.e. grayscale) images, the file extension PGM (Portable Gray Map) is used. In the case of 3 component images, the file extension PPM (Portable Pixel Map) is used.

While 3 component PPM files are specified to describe images in the sRGB colour space, the reference software is agnostic to any colour space specification and will only encode and decode raw sample values.

A PGM/PPM file consist of a header that describes the dimensions and bit depth of the image, concatenated with the binary sample values contained in the image in raster scan order, left to right, top to bottom, with 1, 2, 3 or 6 bytes per sample (depending on the number of components and the bit depth *P*).

Bit depths of $P$>16 cannot be represented by the NetPBM format.

### B.3.2 Header

The header is constructed as follows:

- Two ASCII (see Reference [1]) characters identifying this file as a NetPBM file:
  - For single component images (PGM): `P5`
  - For 3 component images (PPM): `P6`
- An arbitrary number of white space characters, i.e. blank (ASCII 32=0x20), TAB (ASCII 9=0x09) Line Feed (ASCII 10=0x0a) or Carriage Return (ASCII 13=0x0d);

— An integral image width in the number of sample positions, formatted as ASCII decimals;

— White space;

— An integral image height in the number of sample positions, formatted as ASCII decimals;

— White space;

— A maximum sample value formatted as ASCII decimals. This maximum sample value is a number of the form $2^P-1$ where $P$ is the bit depth of the samples.

— A **single** white space character.

### B.3.3 Data

#### B.3.3.1 General

Following the header (immediately after the single white space character), the $W \times H$ sample values follow, in raster scan order, left to right, top to bottom. Each sample value is represented by 1, 2, 3 or 6 bytes, depending on the bit depth $P$ and the number of components (i.e. 1 or 3 components).

#### B.3.3.2 Bit depth P≤8 and 1 component

In this case, each sample value is encoded in binary, in a single byte, right aligned with the unused most significant bits set to zero.

#### B.3.3.3 Bit depth P>8 and 1 component

In this case, each sample value is encoded in binary, in 16 bits (two bytes). The data bits are right-aligned into the two bytes, with the unused most significant bits set to zero. The first byte contains the 8 most significant bits, while the second byte contains the least 8 significant bits. This format is also denoted as big endian.

#### B.3.3.4 Bit depth P≤8 and 3 components

In this case, each sample value is encoded in binary, in 3 bytes, first the red, then the green, then the blue colour coordinate relative to colour primaries that are of no relevance for the reference software. The data bits are right-aligned with the unused most significant bits set to zero. This format stores the colour coordinates interleaved.

#### B.3.3.5 Bit depth P>8 and 3 components

In this case, each sample value is encoded in binary, in 6 bytes. The data bits are right-aligned into the two bytes, with the unused most significant bits set to zero. First the 8 most significant bits of the red colour coordinate, followed by the least 8 significant bits of the red colour coordinate, followed by the 8 most significant bits of the green colour coordinate, followed by the 8 least significant bits of the green colour coordinate, followed by the 8 most significant bits of the blue colour coordinate, then the 8 least significant bits of the blue colour coordinate. This format is also denoted as big endian, and stores the colour coordinates interleaved.

## B.4 Planar YUV

### B.4.1 General

The YUV image file format is an unframed format, i.e. the data does not include any indication of the dimensions and bit depths of the image. Such side information need to be given to the sample encoder as command line parameters, and is required for converting YUV into other formats for visual inspection of such images. YUV files always consist of an even width of (luminance) samples.

Sample precisions *P>16* cannot be represented by ISO/IEC 21122-1 and are thus irrelevant for the purpose of this document.

A YUV file of an image of width $W$ and height $H$ consists of the following (in order):

— $N_y$ sample values of the luminance channel Y, in raster scan order left to right, top to bottom;

— $N_u$ sample values of the U chrominance channel, in raster scan order, left to right, top to bottom;

— $N_v$ sample values of the V chrominance channel, in raster scan order, left to right, top to bottom.

Each individual sample value is encoded in either one or two bytes, depending on the bit depth $P$, as follows:

— If $P \leq 8$, one byte per sample is used to encode a value, and the data bits are right-aligned, with the unused most significant bits set to zero.

— If $8 \leq P \leq 16$, two bytes per sample are used to encode a value, and the data bits are right-aligned into the two bytes, with the unused most significant bits set to zero.

In addition, values $N_{y,}$ $N_u$, and $N_v$ represent the number of sample values for respectively the Y luminance, the U chrominance, and the V chrominance channels. Their specific value depends on the YUV subsampling format.

For YUV 4:4:4, the following numbers apply: $N_y = W \times H$, $N_u = W \times H$, and $N_v = W \times H$.

For YUV 4:2:2, the following numbers apply: $N_y = W \times H$, $N_u = W \times H/2$, and $N_v = W \times H/2$.

For YUV 4:2:0, the following numbers apply: $N_y = W \times H$, $N_u = W \times H/4$, and $N_v = W \times H/4$.

# Annex C
(informative)

# Using the reference software

## C.1 General

This annex provides guidelines on how to use the reference decoder and encoder. For more information, please refer to the packaged `README.md` and the source code.

## C.2 Decoding

The decoder takes a JXS file and generates from that a decoded image (many output image formats are supported, like PPM, PGM, YUV, and PGX). To run it, use:

```
    jxs_decoder [options] <codestream.jxs> <output>
```

The decoder accepts the following options:

| | |
|---|---|
| `-D` | This option will print to the console the actual JPEG XS configuration used in the JXS file as a string. The string represents XS encoding parameters that were used to generate the codestream. Applying the option twice (i.e. as `-DD` or as `-D -D`) will also show the gain and priority values. See also Clause C.4 for the in-depth definition of the syntax and the parameters. |
| `-F <file>` | This option will generate a text output file containing the extraction of codestream fragment sizes and code group counts, as described in Reference [3]. |
| `-f <number>` | Tells the decoder to interpret the input and output as a sequence of files (for frame-based video sequencing). The number specifies the first frame index to process. To be used in combination with `-n`. |
| `-n <number>` | Specifies the total number number of frames to process. |
| `-v` | Output verbose. Use multiple times to increase the verbosity. |

NOTE If the `-D` option is provided, but the output file name is not given, the decoder will only output the XS encoding parameters of the codestream.

Example

```
    jxs_decoder -v woman.jxs out.ppm
```

## C.3 Encoding

The encoder is capable of reading various input file formats, such as PPM, PGM, PGX, planar YUV, and raw RGB, and encodes them with JPEG XS into JXS codestream(s). To run it, use:

```
jxs_encoder [options] <input> <codestream.jxs>
```

NOTE The reference software encoder is provided only as an example implementation. It might not be capable of delivering optimal quality performance due to limitations of the implementation.

The encoder provides the following options:

| | |
|---|---|
| `-c <XS config>` | XS encoding parameters configuring the encoder. This option can be used multiple times, combining the provided settings. See also Clause C.4 for the in-depth definition of the syntax and the parameters. |
| `-D` | This option will print the actual JPEG XS encoding parameters used in the JXS file as a string (see also -c option). Applying the option twice (i.e. as `-DD` or as `-D -D`) will also show the gain and priority values. See also Clause C.4 for the in-depth definition of the syntax and the parameters. |
| `-f <number>` | Tells the encoder to interpret the input and output as a sequence of files (for frame-based video sequencing). The number specifies the first frame index to process. To be used in combination with `-n`. |
| `-n <number>` | Specifies the total number number of frames to process. |
| `-v` | Output verbose. Use multiple times to increase the verbosity. |
| `-w <width>` | Specifies the width of the input file (required for unframed formats like yuv16). |
| `-h <height>` | Specifies the height of the input file (required for unframed formats like yuv16). |
| `-d <bit depth>` | Specifies the bit depth of the input file (required for unframed formats like yuv16). |

NOTE For unframed input file formats, the number of components and often the bit depth are implicitly derived from the file extension.

Examples

```
jxs_encoder -c profile=Main444.12 -c rate=2 woman.ppm woman.jxs
```

```
jxs_encoder -c "profile=Main444.12;rate=2" woman.ppm woman.jxs
```

These two examples are equivalent and encode the `woman.ppm` image using the Main 444.12 profile at a target bit rate of 2 bits per pixel.

## C.4 XS encoding settings

### C.4.1 String syntax

The encoder -c option and the encoder/decoder `-D` options all use the same syntax to describe JPEG XS encoding parameters. These parameters are mostly one-to-one representations of codestream configuration options as described in ISO/IEC 21122-1 and ISO/IEC 21122-2 (a few represent implicit properties, such as the target bitrate, target codestream size, or the rate-allocation line buffer budget).

The profiles, levels and sublevels that can be used are all defined in ISO/IEC 21122-2. Setting either one of these will impose specific XS encoding parameters and limitations. For example, selecting the Main420.12 profile will trigger the encoder to require subsampled 4:2:0 input. The encoder validates the provided XS encoding parameters under the given restrictions and limitations and will produce an error in the case of conflicting settings.

The generic syntax and rules of the XS encoding parameters string are as follows:

— Each encoding parameter is specified as a `key=value` pair. The key is an alpha-numeric string, with all possible keys defined in C.4.2;

— Encoding parameters are separated by semi-colons (";" ASCII 59=0x3b). Multiple `-c` arguments are internally concatenated to produce a single XS encoding parameters string comprised of key/value pairs;

— Values can be either textual or numeric (integral or floating point), but a value is not allowed to be an empty string;

— The XS encoding parameters string is case insensitive (everything is converted to lower case);

— White spaces are ignored;

— By not specifying an XS encoding parameter, the encoder will resolve to either a default value (as defined by the profile) or an automated best-effort guess;

— Not all encoding parameter configurations are valid, and the encoder will produce an error in case a configuration is invalid.

### C.4.2 Keys

The following XS encoding parameter keys are defined (for more information, consult ISO/IEC 21122-1 and ISO/IEC 21122-2):

| | |
|---|---|
| `p`<br>`profile` | The profile to use (by name). See ISO/IEC 21122-2 or the help output of the encoder if run without options. |
| `l`<br>`lev`<br>`level` | The level to use (by name). See ISO/IEC 21122-2 or the help output of the encoder if run without options. |
| `s`<br>`sublev`<br>`sublevel` | The sublevel to use (by name). See ISO/IEC 21122-2 or the help output of the encoder if run without options. |
| `rate`<br>`bpp` | Specify a target bit-rate (in bpp). Cannot be combined with `size`. |
| `size` | Specify a target codestream size (in bytes). Cannot be combined with `bpp` or `rate`. |
| `budget_lines`<br>`budget_report_lines` | The size of the buffer model (in lines) for the rate allocation of the encoder. See ISO/IEC 21122-2. |
| `cpih`<br>`mct` | Select the multiple component transform (MCT). Valid options are:<br>— `none` or 0 for no transform;<br>— `rct` or 1 for RCT (RGB to YUV);<br>— `tetrix,<Cf>,<e1>,<e2> or 3,<Cf>,<e1>,<e2>` for Star-Tetrix (note that additional settings are required, comma separated). |
| `cfa` | In case of 1-component input with one of the Bayer profiles, a CFA pattern needs to be specified. Supported patterns are:<br>— RGGB;<br>— BGGR;<br>— GRBG;<br>— GBGR. |

| | |
|---|---|
| `nlt` | Non-linear transform. Additional settings are required, depending on the chosen NLT. Valid NLTs are:<br>— `none`;<br>— `quadratic,<sigma>,<alpha>`;<br>(sigma represents the sign and alpha the absolute value of the DCO offset)<br>— `extended,<theta1>,<theta2>`;<br>(theta1 and theta2 are the encoder-side thresholds, i.e. the black level and the toe region, with the linear exponent E set to 3, see also ISO/IEC 21122-1)<br>— `extended,<E>,<T1>,<T2>`<br>(E, T1, and T2 are the parameters of the NLT marker, see also ISO/IEC 21122.1). |
| `cw` | Precinct column width in multiples of 8 LL-band samples. Cw=0 represents full-width (one column). |
| `slh` | Slice height (in lines). Must be a multiple of the intrinsic precinct height. |
| `bw` | Nominal bit precision of the wavelet coefficients (restricted to input-bit-depth, 18, or 20). |
| `fq` | Number of fractional bits of the wavelet coefficients (restricted to 8, 6, or 0). |
| `nlx` | Number of horizontal wavelet decompositions. |
| `nly` | Number of vertical wavelet decompositions (must be smaller than or equal to `nlx`). |
| `lh` | Long precinct header enforcement flag (0 to disable or 1 to enable). |
| `rl` | Raw-mode selection per packet flag (0 to disable or 1 to enable). |
| `qpih`<br>`quant` | Quantization type (use 0 for dead-zone quantization, and 1 for uniform quantization). |
| `fs` | Sign handling strategy (use 0 for jointly, and 1 for separate packets). |
| `rm` | Run mode (use 0 so runs indicate zero prediction residuals, and 1 so runs indicate zero coefficients). |
| `sd` | Number of components for which the wavelet decomposition is suppressed. |
| `gains` | Comma separated list of gain values (one value per band, which depends on number of components, subsampling and `nlx/nly`). Alternatively, the encoder has some built-in tables for PSNR and Visual optimized compression. Specify either `psnr` or `visual` keywords to select built-in gains/priorities. |
| `priorities` | Comma separated list of priority values (same amount as gains). |

Example 1

```
jxs_encoder -vv -DD -c "rate=4;lh=0;rl=1;gains=visual" sintel.ppm sintel.jxs
```

Encodes sintel.ppm at 4 bits per pixel. Profile, level and sublevel are automatically selected. Long precinct header flag is disabled, raw-mode per packet flag is enabled, and matching built-in gains/

priorities are applied for visual optimized quality. Verbosity is set to level 2. The double `-D` option will also print out the full XS encoding parameters string as extra information.

Example 2

```
jxs_encoder -c "p=Main422.10" -c "rate=2" -w 1920 -h 1080 -d 10 sintel.yuv16_422p sintel.
jxs
```

Encodes a frame of sintel, given in YUV 422 10-bit format to 2 bits per pixel, using the Main422.10 profile.

NOTE 1 One particular useful way to learn and understand the XS encoding parameters string syntax and possibilities is to use the decoder to print (using the `-D` option) configurations of the provided reference codestreams of ISO/IEC 21122-4.

NOTE 2 Some implicit encoding parameters cannot be retrieved from a codestream using the `-D` option (i.e. the budget_lines parameter).

## Bibliography

[1] ISO/IEC 646, *Information technology — ISO 7-bit coded character set for information interchange*

[2] Poskanzer J., PPM Format Specification", online at http://netpbm.sourceforge.net/doc/ppm.html (retrieved July 2019)

[3] ISO/IEC 21122-4, *Information Technology — JPEG XS Low-latency lightweight image coding system — Part 4: Conformance testing*

**ICS 35.040.30**

Price based on 13 pages