SMPTE 268M-2003
Revision of
ANSI/SMPTE 268M-1994

SMPTE STANDARD

for File Format for Digital Moving-Picture Exchange (DPX), Version 2.0

1 Scope

1.1 This standard defines a file format for the exchange of digital moving pictures on a variety of media between computer-based systems. It does not define the characteristics of input or output devices or displays. This format will be known as the SMPTE digital picture exchange format version 2.0, or DPX in short form. The file extension will be .dpx.

1.2 This flexible, resolution-independent file format describes pixel-based (raster) images with attributes defined in the binary file header. Each file represents a single image with up to eight image elements. Image elements are defined as a single component (e.g. luma) or multiple components (e.g. red, green, and blue) as defined by table 1.

1.3 Image data is packed for efficient storage with the option to pad to 32-bit word boundaries (two alternative padding formats are allowed: see table 3B and annex B). Multibyte quantities may be stored with either the most significant byte first or the least significant byte first, where first means in the location with the lowest address, or the first byte in sequence from a byte-serial data channel. Both byte-order conventions are supported. The ‘‘magic number’’ in field 1 of the file information section is used to distinguish the byte order (annex A provides an historical perspective for the existence of the two byte-order conventions).

2 Normative references

The following standards contain provisions which, through reference in this text, constitute provisions of this standard. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this standard are encouraged to investigate the possibility of applying the most recent edition of the standards indicated below.

ANSI/IEEE 754-1985 (R1991), Binary Floating-Point Arithmetic

ANSI/SMPTE 125M-1995, Television — Component Video Signal 4:2:2 — Bit-Parallel Digital Interface

SMPTE 12M-1999, Television, Audio and Film — Time and Control Code

SMPTE 240M-1999, Television — 1125-Line High-Definition Production Systems — Signal Parameters

SMPTE 254-2002, Motion-Picture Film (35-mm) — Manufacturer-Printed, Latent Image Identification Information

SMPTE 274M-2003, Television — 1920 x 1080 Image Sample Structure, Digital Representation and Digital Timing Reference Sequences for Multiple Picture Rates

SMPTE 296M-2001, Television — 1280x720 Progressive Image Sample Structure — Analog and Digital Representation and Analog Interface

Approved
May 5, 2003

ISO 8601:2000, Data Elements and Interchange Formats — Information Interchange — Representation of Time and Dates

ISO/IEC 8859-1:1998, Information Processing — 8-Bit Single-Byte Coded Graphics Character Sets — Latin Alphabets 1

ITU-R BT.601-5 (10/95), Studio Encoding Parameters of Digital Television for Standard 4:3 and Wide-Screen 16:9 Aspect Ratios

ITU-R BT.709-5 (04/02), Parameter Values for the HDTV Standards for Production and International Programme Exchange

3 Definitions

The reference field number from clauses 5 and 6 is indicated in brackets at the end of each definition:

3.1 magic number: Indicates the start of the image file and is used to determine byte order. The file format allows machines to create files in either of the two most common byte orders, whichever is easier for that machine. Byte-order translation is only required for machines reading files that were created on a machine with reverse byte order. Programs creating DPX files should write the magic number with the ASCII value of ‘‘SDPX’’ (0x53445058 hex). Programs reading DPX files should use the first four bytes to determine the byte order of the file. The first four bytes will be S, D, P, X if the byte order is most significant byte first, or X, P, D, S if the byte order is least significant byte first. [1]

3.2 image file size: Indicates the size of the entire file , i.e. containing both header and image data. [4]

3.3 ditto key: Indicates that all fields are the same as the previous frame in the sequence except for fields related to the frame number (48, 50, 58, 61). Also, the offsets to the image data (21.12) will change if run-length encoding is used. The ditto key is a read-time shortcut only, and the other fields in the header must still be filled in when the file is created. [5]

3.4 creation date/time: Defined as yyyy:mm:dd:hh:mm:ssLTZ, formatted according to ISO 8601. [10]

“LTZ” means “Local Time Zone;” format is:

LTZ = Z *(time zone = UTC)*, or LTZ = +/–hh, or LTZ = +/–hhmm *(local time is offset from UTC)*

3.5 encryption key: Indicates that the image data is encrypted to prevent unauthorized use. The default is **FFFFFFFF** for no encryption. Any other value indicates that the image data is encrypted and this value can be used as the encryption key. Note that the header data is not encrypted. [15]

3.6 image orientation: Indicates the orientation of the image data required for display. The possible orientations are listed in table 2. The standard orientation for core set images (code 0) is left to right (line direction) and top to bottom (frame direction). [17]

3.7 image element data structure: A data structure (group of fields) is repeated for each image element. An image element can contain a single component or multiple components, as defined in table 1. The presentation order of both whole elements and components of multi-component image elements is also defined in Table 1. All components in an image element must have the same number of bits, transfer function, and colorimetric specification. [21]

3.8 reference low data code value: Defines the minimum expected code value for image data. For printing density, the default value is 0. For ITU-R 601-5 luma, the default value is 16. [21.2]

3.9 reference low quantity represented: Defines the corresponding signal level or measured value to the reference low data code value. For printing density, the default is a density of 0.00. For ITU-R 601-5, the luma default is 0 mV. [21.3]

3.10 reference high data code value: Defines the maximum expected code value for image data. For 10-bit printing density, the default code value is 1023. For ITU-R 601-5 luma, the default value is 235. [21.4]

3.11 reference high quantity represented: Defines the corresponding signal level or measured value to the reference high data code value. For printing density, the default is a density of 2.048. For ITU-R 601-5 luma, the default is 700 mV. [21.5]

3.12 descriptor for image element n: Defines the components that make up an image element and their pixel packing order. The valid components are listed in table 1. [21.6]

3.13 transfer characteristic: Defines the amplitude transfer function used to transform the data from a linear original. The inverse of the transfer function is needed to recreate a linear image element (see table 5A). [21.7]

3.14 colorimetric specification: Defines the appropriate color reference primaries (for additive color systems like television) or color responses (for printing density) (see table 5B). [21.8]

3.15 bit depth: Defines the number of bits for each component in the image element. All components must have the same bit depth. Valid bit depths are 1-, 8-, 10-, 12-, and 16-bit integer, and 32- and 64-bit IEEE floating point (see table 3A). [21.9]

3.16 packing: For image element n, defines the data packing mode. The valid options are listed in table 3B. [21.10]

3.17 encoding: For image element n, defines whether or not the element is run-length encoded. The valid options are listed in table 3C. [21.11]

3.18 offset: To data for image element n, defines the offset in bytes to the image data for element n from the beginning of the file. [21.12]

3.19 end-of-line padding: Specifies the number of padded bytes at the end of each line. The default is 0 (no padding). [21.13]

3.20 end-of-image padding: Specifies the number of padded bytes at the end of each image element. The default is 0 (no padding). [21.14]

3.21 X offset: Defines the line offset (in pixels) from the first pixel in the original image. The default is 0. This is useful if an image is cropped and the user wishes to specify its location with respect to the original contiguous image.[30]

3.22 Y offset: Defines the frame offset (in lines) from the first line in the original contiguous image. The default is 0. [31]

3.23 X center: Defines the X image center in pixel units (floating point). [32]

3.24 Y center: Defines the Y image center in line units (floating point). [33]

3.25 X pixel count: Defines the number of pixels per line in the original image. [34]

3.26 Y pixel count: Defines the number of lines per image in the original image. [35]

3.27 source image filename: Defines the source image from which this image was extracted or processed. [36]

3.28 source image date/time: Defines the creation time of the source image from which the image was extracted or processed. Formatting is as per clause 3.4 [37]

3.29 border validity: Defines the region of an image that is eroded due to edge-sensitive filtering operations. The X-left, X-right, Y-top, and Y-bottom value defines the width of the eroded border. The default is 0,0,0,0 in pixel units (no erosion). [40]

3.30 pixel aspect ratio: Specified as the ratio of a horizontal integer and a vertical integer. For example, a SMPTE 274M signal has a pixel aspect ratio of 1:1, which is 1920 active pixels and 1080 active lines in a 16:9 frame. [41]

3.31 X scanned size: Defines the horizontal size of the original scanned optical image in millimeters. [42.1]

3.32 Y scanned size: Defines the vertical size of the original scanned optical image in millimeters. [42.2]

3.33 film edge code information: Encodes data from machine readable portion of film edge code, according to SMPTE 254. [43, 44, 45, 46, 47, 48]

3.34 frame position in sequence: Defines the frame number in the image sequence. [50]

3.35 sequence length: Defines the total number of frames in the image sequence. [51]

3.36 held count: Specifies how many sequential frames for which to hold the current frame. In animation, it is often desirable to hold identical frames. [52]

3.37 shutter angle: Defines the shutter angle in degrees of the motion-picture camera. This specifies the temporal sampling aperture. [54]

3.38 frame identification: A user-defined field that labels select frames as key frames or wedge frames, etc. [55]

3.39 slate information: A user-defined ASCII field for recording production information from the camera slates. [56]

3.40 SMPTE time code: The characters are encoded into the 32-bit word according to table 6 [58].

3.41 SMPTE user bits: These are encoded according to table 6 [59]

3.42 field number: Of the first field in the file, may be 1 or 2 for component video, 1 to 4 for NTSC or component video decoded from NTSC, or 1 to 12 for PAL or component video decoded from PAL. Color frame sequence information is useful when decoding and subsequently re-encoding component video. The field number is set to 0 where field designation is inappropriate. [61]

3.43 video signal standard: Defines the video source. Video signal standards are listed in table 4. [62]

3.44 horizontal sampling rate (Hz): The clock rate at which samples were acquired. This is an inverse function of the total number of samples per scan line, rather than the active number of pixels per line indicated in field 19. Thus, for SMPTE 274M at 24.00 Hz frame rate, for example, it would be 74.25 MHz. [64]

3.45 vertical sampling rate (Hz): The rate at which the scanning of the whole extent of the image is repeated, even if each such scan is incomplete, i.e. is interlaced. Thus, for example, although 625/50 scanning has a true frame rate of 25Hz, its vertical sampling rate would be considered to be 50Hz. [65]

3.46 time offset from sync to first pixel (microseconds): Defines the edge of the digital image with respect to sync and the sampling phase which is necessary to reconstruct a composite image. The sync reference is the reference edge of horizontal sync. [67]

3.47 gamma: Defines the power law exponent that represents the gamma correction applied to a video image. In the expression Y = X 1/gamma , the default gamma for NTSC is 2.2. [68]

3.48 black level code value: Defines the digital code value representing reference black (camera lens capped, RGB signal set to 0 mV). For ITU-R 601-5, the default black level code value is 16. [69]

3.49 black gain: Defines the linear gain applied to signals below the breakpoint (this is 4.5 for SMPTE 274M). [70]

3.50 breakpoint: Defines the signal level above which the gamma law is applied (this is 0.018 of full scale for SMPTE 274M). [71]

3.51 reference white level code value: Defines the digital code value representing reference white (90% reflectance white card, RGB signal set to 700 mV). For ITU-R 601-5, the default reference white level code value is 235. [72]

3.52 integration time: Defines the temporal sampling aperture of the television camera; most useful for CCD cameras. [73]

4 File

4.1 The file contains four sections, the first three of which are header information:

– (i) Generic file information, image information, data format, and image origination information (fixed length) [Clause 5];

– (ii) Motion-picture and television industry-specific information (fixed length) [Clause 6];

– (iii) User-defined information. This section provides an extended area for customized information needed by some users. The format of this section is not defined by the standard. This section is variable length with a maximum length of 1 Mbyte. It may be of zero length; [Clause 7]

– (iv) Image data [Clause 8].

4.2 Each field in the file header contains data of specified types. The valid types (and undefined values) for each field are:

Type	Undefined value
U8 unsigned 8-bit integer	FF hex
U16 unsigned 16-bit integer	FFFF hex
U32 unsigned 32-bit integer	FFFFFFFF hex
R32 32-bit real number (IEEE floating point)	FFFFFFFF hex
ASCII	0 hex (NULL character)

4.3 To provide a streamlined path for implementation and testing, a core set of fields has been identified with a ‘‘C’’ in the field designation table. The rules necessary for interchange are:

– This core set contains the minimum amount of information that a reader needs to read and interpret a file;

– A core-compliant reader must read the core fields, but need not read the others;

– A core-compliant writer must fill the core fields with valid values (undefined values are not permitted). Non-core fields must be filled with UNDEFINED values if the correct value is not known.

4.4 Unless stated otherwise, all references in this standard to binary data, sizes, offsets, and lengths are in units of bytes. Positions within the file are specified in terms of the number of bytes from the beginning of the file, with the first byte designated as byte 0. Offsets to individual fields are specified from the first byte.

4.5 All ASCII character strings that do not fill a whole field are terminated by a NULL (zero) byte.

5 Generic headers

5.1 File information header

Field	Offset	Length	Type	Core	Content
1	0	4	U32	C	Magic number (SDPX ASCII)
2	4	4	U32	C	Offset to image data in bytes
3	8	8	ASCII	C	Version number of header format (V2.0)
4	16	4	U32	C	Total image file size in bytes (including file header)
5	20	4	U32		Ditto key (0 = same as previous frame; 1 = new)
6	24	4	U32		Generic section header length in bytes
7	28	4	U32		Industry specific header length in bytes
8	32	4	U32		User-defined header length in bytes
9	36	100	ASCII		Image filename
10	136	24	ASCII		Creation date/time: yyyy:mm:dd:hh:mm:ssLTZ
12	160	100	ASCII		Creator
13	260	200	ASCII		Project name
14	460	200	ASCII		Right to use or copyright statement
15	660	4	U32		Encryption key (FFFFFFFF unencrypted)
16	664	104	TBD		Reserved for future use

5.2 Image information header

Field	Offset	Length	Type	Core	Content
17	768	2	U16	C	Image orientation (see table 2)
18	770	2	U16	C	Number of image elements (1 – 8)
19	772	4	U32	C	Pixels per line
20	776	4	U32	C	Lines per image element

21	Data structure for image element 1				
21.1	780	4	U32	C	Data sign (0 = unsigned; 1 = signed)
					(core set images are unsigned)
21.2	784	4	U32		Reference low data code value
21.3	788	4	R32		Reference low quantity represented
21.4	792	4	U32		Reference high data code value
21.5	796	4	R32		Reference high quantity represented
21.6	800	1	U8	C	Descriptor (see table 1)
21.7	801	1	U8	C	Transfer characteristic (see table 5A)
21.8	802	1	U8	C	Colorimetric specification (see table 5B)
21.9	803	1	U8	C	Bit depth (see table 3A)
21.10	804	2	U16	C	Packing (see table 3B)
21.11	806	2	U16	C	Encoding (see table 3C)
21.12	808	4	U32	C	Offset to data
21.13	812	4	U32		End-of-line padding
21.14	816	4	U32		End-of-image padding
21.15	820	32	ASCII		Description of image element

5.2 Image information header (continued)

Field	Offset	Length	Type	Content
22	852	72	Structure	Data structure for image element 2
23	924	72	Structure	Data structure for image element 3
24	996	72	Structure	Data structure for image element 4
25	1068	72	Structure	Data structure for image element 5
26	1140	72	Structure	Data structure for image element 6
27	1212	72	Structure	Data structure for image element 7
28	1284	72	Structure	Data structure for image element 8
29	1356	52	TBD	Reserved for future use

5.3 Image source information header

Field	Offset	Length	Type	Content
30	1408	4	U32	X offset
31	1412	4	U32	Y offset
32	1416	4	R32	X center
33	1420	4	R32	Y center
34	1424	4	U32	X original size
35	1428	4	U32	Y original size
36	1432	100	ASCII	Source image filename
37	1532	24	ASCII	Source image date/time: yyyy:mm:dd:hh:mm:ssLTZ
38	1556	32	ASCII	Input device name
39	1588	32	ASCII	Input device serial number
40	1620	8	U16*4	Border validity: XL, XR, YT, YB border
41	1628	8	U32*2	Pixel aspect ratio (horizontal:vertical)

42	Data structure for additional source image information			
42.1	1636	4	R32	X scanned size
42.2	1640	4	R32	Y scanned size
42.3	1644	20	TBD	Reserved for future use

6 Industry-specific headers

6.1 Motion-picture film information header

Field	Offset	Length	Type	Content
43	1664	2	ASCII	Film mfg. ID code (2 digits from film edge code)
44	1666	2	ASCII	Film type (2 digits from film edge code)
45	1668	2	ASCII	Offset in perfs (2 digits from film edge code)
47	1670	6	ASCII	Prefix (6 digits from film edge code)
48	1676	4	ASCII	Count (4 digits from film edge code)
49	1680	32	ASCII	Format – e.g. Academy
50	1712	4	U32	Frame position in sequence
51	1716	4	U32	Sequence length (frames)
52	1720	4	U32	Held count (1 = default)
53	1724	4	R32	Frame rate of original (frames/s)
54	1728	4	R32	Shutter angle of camera in degrees
55	1732	32	ASCII	Frame identification – e.g. keyframe
56	1764	100	ASCII	Slate information
57	1864	56	TBD	Reserved for future use

6.2 Television information header

Field	Offset	Length	Type	Content
58	1920	4	U32	SMPTE time code (Table 6)
59	1924	4	U32	SMPTE user bits (Table 6)
60	1928	1	U8	Interlace (0 = noninterlaced; 1 = 2:1 interlace)
61	1929	1	U8	Field number
62	1930	1	U8	Video signal standard (see table 4)
63	1931	1	U8	Zero (for byte alignment)
64	1932	4	R32	Horizontal sampling rate (Hz)
65	1936	4	R32	Vertical sampling rate (Hz)
66	1940	4	R32	Temporal sampling rate or frame rate (Hz)
67	1944	4	R32	Time offset from sync to first pixel (ms)
68	1948	4	R32	Gamma
69	1952	4	R32	Black level code value
70	1956	4	R32	Black gain
71	1960	4	R32	Breakpoint
72	1964	4	R32	Reference white level code value
73	1968	4	R32	Integration time (s)
74	1972	76	TBD	Reserved for future use

7 User defined data

Field	Offset	Length	Type	Content
75	2048	32	ASCII	User identification
76	2080	xx	TBD	User defined – Postage stamp, processing logs, etc.
				(length is variable with maximum length of 1 Mbyte)

8 Image data

Field	Offset	Length	Type	Content
77	xx	xx	Array U8*4	Image data should start at block boundary (8-K blocks are recommended for efficient use of tape-storage devices).

Table 1 – Image element descriptors

Value	Components (and order in unpacked stream)
0	User defined (or unspecified single component)
1	Red (R)
2	Green (G)
3	Blue (B)
4	Alpha (matte)
6	Luma (Y) – Note 1
7	Color Difference (C_B, C_R, subsampled by two)
8	Depth (Z)
9	Composite video
10 – 49	Reserved for future single components
50	R,G,B – Note 2
51	R,G,B, Alpha (A) – Note 2
52	A, B, G, R – Note 3
53 – 99	Reserved for future RGB ++ formats
100	C_B, Y, C_R, Y (4:2:2) ---- based on SMPTE 125M
101	C_B, Y, A, C_R, Y, A (4:2:2:4)
102	C_B, Y, C_R (4:4:4)
103	C_B, Y, C_R, A (4:4:4:4)
104 – 149	Reserved for future C_BYC_R ++ formats
150	User-defined 2-component element
151	User-defined 3-component element
152	User-defined 4-component element
153	User-defined 5-component element
154	User-defined 6-component element
155	User-defined 7-component element
156	User-defined 8-component element
157 – 254	Reserved for future formats

NOTES

1 In a Y-only (black and white) image file packed by inserting three Y (single component) image elements into the file space of one RGB (multi component) image element, the first image datum shall be the first pixel according to the image orientation code shown in table 2 (see annexes B and C).

2 The first datum shall represent the Blue component of an RGB multi-component image element; the second datum shall represent the Green component; the third datum shall represent the Red component. An Alpha component, if present, shall be represented by the fourth datum. This sequence shall continue until the end of the file is reached.

3 Sequence as per note 2, but reversed.

General

These values describe the components that make up an image element and their order. A pixel consists of 1 – 8 components as specified by field 21.6. All components in an image element shall have the same number of bits and the same data metric.

For any of the subsampled C_B, Y, C_R formats, a pixel for the purposes of run-length encoding and component packing is really two picture elements. The pixels per line specified in field 19 refer to the number of picture elements in the original image and must be an even number.

Table 2 – Image orientation code

Code	Line direction	Frame direction
0[1]	Left to right	Top to bottom
1	Right to left	Top to bottom
2	Left to right	Bottom to top
3	Right to left	Bottom to top
4	Top to bottom	Left to right
5	Top to bottom	Right to left
6	Bottom to top	Left to right
7	Bottom to top	Right to left
8 – 254	Reserved for future use	

[1] Orientation 0 is the only one supported in the core set file format.

Table 3A – Valid bit depths for image elements

1	integer
8	integer
10	integer
12	integer
16	integer
32	IEEE floating point (R32)
64	IEEE floating point (R64)

Table 3B – Component data packing method

0	Packed into 32-bit words [1]
1	Filled to 32-bit words, method A [2], [4]
2	Filled to 32-bit words, method B [3], [4]
3 – 7	Reserved for future use

NOTE – This table contains the values for field 21.10, component data packing. Note that all components in a pixel (including the run-length flag if used) are the same bit size (a diagram illustrating the packing of 8-, 10-, 12-, and 16-bit channels into 32-bit words is included in Annex C).

[1] For 1-bit components, the component pixels are first packed into bytes with the left-most (first) pixel bit in the least significant bit of the byte. The bytes are then sequenced according to the order specified by field 21.10 and packed into 32-bit words in the same manner as 32-bit data.

[2] Filling method A is normal: padding bits precede data within 32-bit word boundaries (10-bit image components) or within 16-bit word boundaries (12-bit image components. See Annex C, figs C.3 and C.4.

[3] Filling method B is now non-standard: padding bits follow data within word boundaries. See Annex C, figs C.7/C.8.

[4] 1-, 8-, and 16-bit data never needs filling; therefore, the corresponding states are not needed.

Table 3C – Component data encoding method

0	No encoding applied
1	Run-length encoded [1)]
2 – 7	Reserved for future use

NOTE – This table contains the values for field 21.11, component data encoding. Only run-length encoding is specified at this time, but there is provision for future expansion.

[1)] With run-length encoding, the components of consecutive pixels are grouped into ‘‘runs’’ which are preceded by a run-length flag. The RL flag has the same size as each component. Once again, the resulting data stream is packed as specified by field 21.10.

The least significant bit of the run-length flag signals a run of pixels which are all the same if set, and a run of pixels which are all different if clear. The remaining bits indicate the number of pixels in the run. In the case of a run of all the same pixels, the flag word is followed by a single pixel which is to be replicated to fill out the run. In the case of a run of all different pixels, the flag is followed by a run-length of pixels.

Runs will always break at scan line boundaries. Packing will always break to the next 32-bit word at scan line boundaries.

Table 4 – Video signal standard

Code	Signal standard
0	Undefined [1)]
1	NTSC
2	PAL
3	PAL-M
4	SECAM
5 – 49	Reserved for other composite video
50	YCBCR ITU-R 601-5 525-line, 2:1 interlace, 4:3 aspect ratio
51	YCBCR ITU-R 601-5 625-line, 2:1 interlace, 4:3 aspect ratio
52 – 99	Reserved for future component video
100	YCBCR ITU-R 601-5 525-line, 2:1 interlace, 16:9 aspect ratio
101	YCBCR ITU-R 601-5 625-line, 2:1 interlace, 16:9 aspect ratio
102 – 49	Reserved for future widescreen
150	YCBCR 1050-line, 2:1 interlace, 16:9 aspect ratio
151	YCBCR 1125-line, 2:1 interlace, 16:9 aspect ratio (SMPTE 274M)
152	YCBCR 1250-line, 2:1 interlace, 16:9 aspect ratio
153	YCBCR 1125-line, 2:1 interlace, 16:9 aspect ratio (SMPTE 240M)
154 – 199	Reserved for future high-definition interlace
200	YCBCR 525-line, 1:1 progressive, 16:9 aspect ratio
201	YCBCR 625-line, 1:1 progressive, 16:9 aspect ratio
202	YCBCR 750-line, 1:1 progressive, 16:9 aspect ratio (SMPTE 296M)
203	YCBCR 1125-line, 1:1 progressive, 16:9 aspect ratio (SMPTE 274M)
204 – 254	Reserved for future high-definition progressive

[1)] For an undefined video signal standard, it is necessary to specify the following fields that would otherwise be fully specified by selecting one of the video signal standards:

68	Gamma
69	Black level code value
70	Black gain
71	Breakpoint
72	Reference white level code value

Table 5A – Transfer characteristic

Code	Transfer characteristic
0	User defined
1	Printing density
2	Linear
3	Logarithmic [to be defined by SMPTE I23 Technology Committee, sub-group on “Transfer Characteristics”]
4	Unspecified video
5	SMPTE 274M
6	ITU-R 709-4
7	ITU-R 601-5 system B or G (625)
8	ITU-R 601-5 system M (525)
9	Composite video (NTSC); see SMPTE 170M
10	Composite video (PAL); see ITU-R 624-4
11	Z (depth) – linear
12	Z (depth) – homogeneous (distance to screen and angle of view must also be specified in user-defined section)
13 – 254	Reserved for future use

Table 5B – Colorimetric specification

Code [1)]	Colorimetric Specification
0	User defined
1	Printing density
2	Not applicable
3	Not applicable
4	Unspecified video
5	SMPTE 274M
6	ITU-R 709-4
7	ITU-R 601-5 system B or G (625)
8	ITU-R 601-5 system M (525)
9	Composite video (NTSC); see SMPTE 170M
10	Composite video (PAL); see ITU-R 624-4
11	Not applicable
12	Not applicable
13 – 254	Reserved for future use

[1)] The codes are assigned to correspond to those in table 5A, except where there is no appropriate colorimetric specification.

Table 6 – Time code and user bits

SMPTE 12M timecode

31..28	27..24	23...20	19...16	15...12	11...8	7...4	3...0
h	h	m	m	s	s	F	F

SMPTE 12M userbits

31..28	27..24	23...20	19...16	15...12	11...8	7...4	3...0
UB8	UB7	UB6	UB5	UB4	UB3	UB2	UB1

Annex A (informative)
Structure of 268M file and representation in document

Item	Starting point in file (Offset in bytes)	Total space used in file (bytes)	Document section No.
Generic headers:			
File information header	0	768	5.1
Image information header	768	640	5.2
Image source information Header	1408	256	5.3
Industry-specific headers:			
Motion-picture film Information header	1664	256	6.1
Television information header	1920	128	6.2
Data:			
User defined data	2048	Variable to max of 1 MByte	7
Image data	Variable (specified in File information header)	Variable (computable from File information header)	8.

Annex B (informative)
Byte-order conventions

Digital computers save information in a form commonly known as bits. For convenient (and fast) information manipulation, most computers manipulate more than one bit at a time. They manipulate multiple bits as a symbol, the most common of which is a byte (8 bits). A further extension of this concept is to manipulate more than one byte at a time, working with multibyte words. The word size is built into the computer hardware and cannot be altered by software.

As computers were developed, there was no standardization between different types of computers and data format. Two different orderings of bytes in words were developed in parallel. Early on, there were arguments for standardization of byte order. However, the arguing proponents became entrenched and now there is an equal number of both types of systems in use.

When one generates files on one type of computer for exchange with another, the receiver of the file must know what type of computer generated it in order to interpret it properly. The most common automatic method for doing this is to create a magic number. The magic number is a multibyte word with the largest number of bytes per word that the file will contain. Each byte of the magic number is different from all others and the magic number is published with the file format specification. This magic number is then coded into the file reader software so that the reader can define the byte order of the computer generating the file. If the file-reading computer sees the magic number in its correct format, then it has the same byte order as the file-generating computer. If not, the reading computer must convert all the multibyte words in the file into its own byte order before it can use the data.

The reason therefore that most file formats (including SMPTE DPX) do not dictate a particular byte order is that it would unfairly burden one half of the computers in use. These computers, even when communicating between themselves exclusively, would have to convert all of the multibyte words when creating a file and convert them back when reading a file.

Annex C (informative)
Data-packing diagrams (Figures C.1 through C.8)

These diagrams illustrate the packing of 8-, 10-, 12-, and 16-bit components into 32-bit and 16-bit words, using the most-significant- byte-first convention.

For image component sets that do not align to word boundaries, both filled (justified to a file byte boundary) and non-filled packing is shown.

For the *nonfilled,* non-aligned formats (figures C.2 and C.4), the zero datum (component) is placed in the least significant n bits of the first 32-bit word. The next datum is placed in the next most significant n bits. When a datum no longer fits in the remaining bits of a 32-bit word, it is broken, with as many least significant bits as will fit placed in the first 32-bit word, and the remaining bits placed in the low-order bits of the next 32-bit word. Any bits in the last word of a scan line left over will be filled with zeroes. That is to say that the packing is broken on scan line boundaries.

For *filled* packing of non-aligned components, two methods are shown: figures C.3 and C.5 show filling implemented according to the provisions of this version of this standard; figures C.7 and C.8 show filling according to an earlier version of this standard. The adopted method must be signaled in the Data Structure fields in the header for each image element.

Component sets that *do* align to word boundaries need no filling or breaking (figures C.1 and C.6)

Annex C (informative) Data Packing Diagrams - Including “Method A” Filling

Fig. C.1 8-bit component(s):

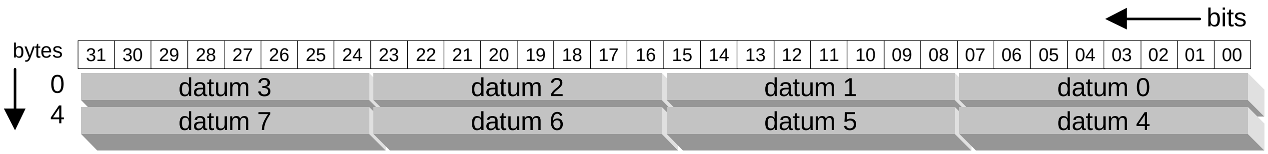


Fig. C.2 10-bit component(s):

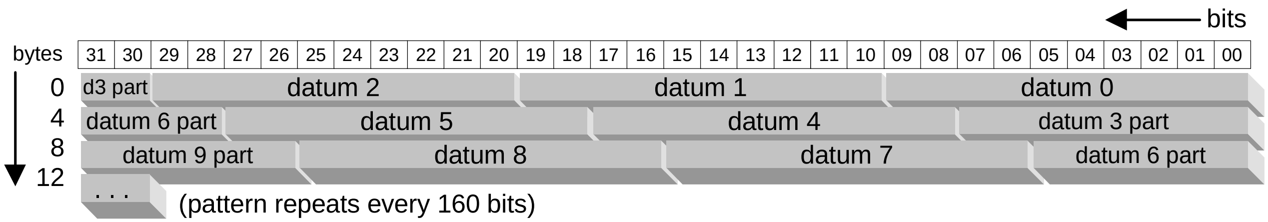


Fig. C.3 10-bit component(s) filled to 32-bit word boundaries, leading padding bits (Method A):

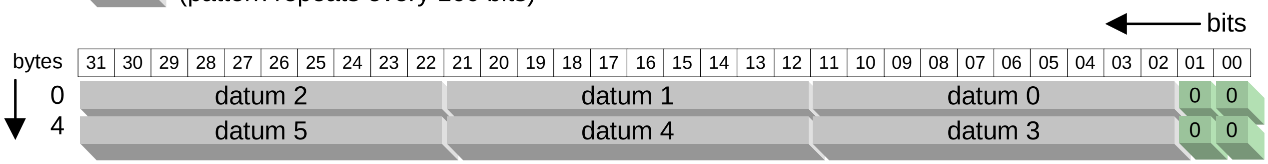


Fig. C.4 12-bit component(s):

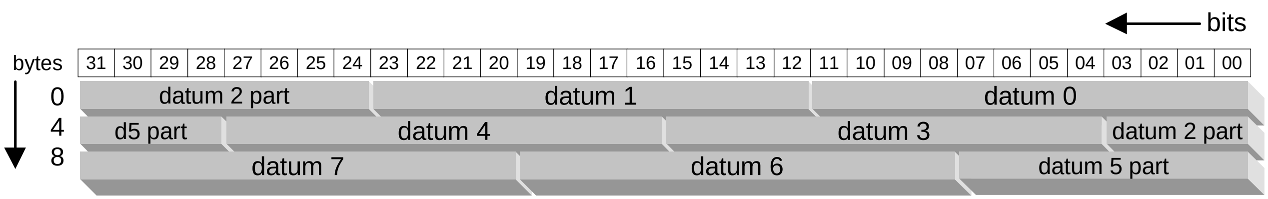


Fig. C.5 12-bit component(s) filled to 16-bit word boundaries, leading padding bits (Method A):

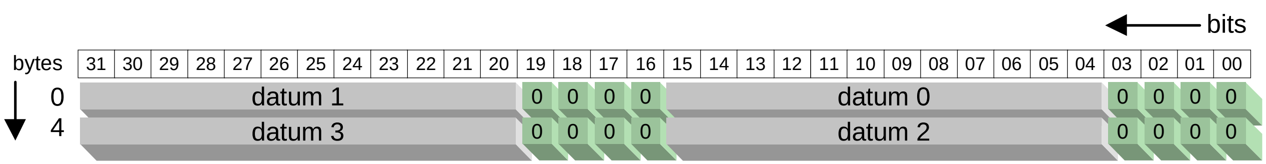
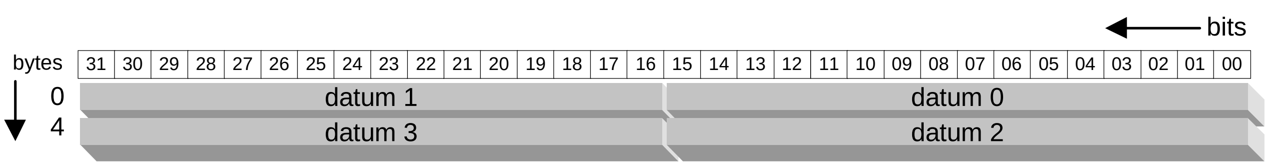


Fig. C.6 16-bit component(s):



Annex C (informative) Data Packing "Method B"

Fig. 10-bit filled to 32-bit boundaries, padding (Method

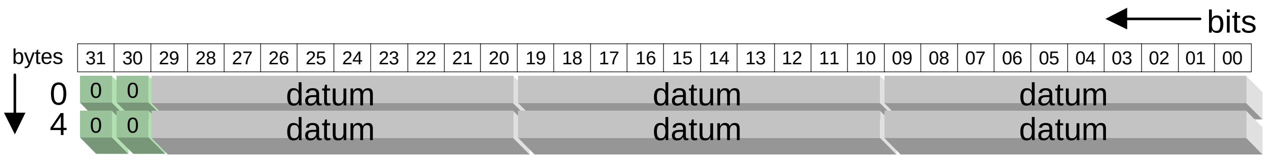
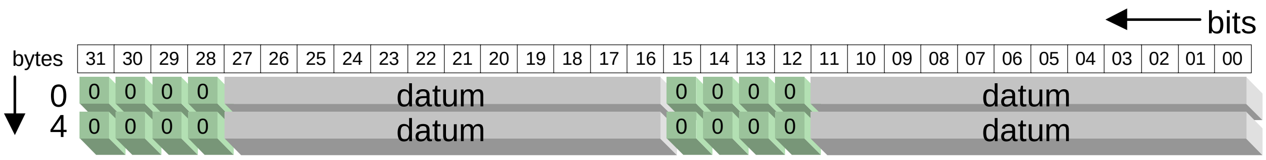


Fig. 12-bit filled to 16-bit boundaries, padding (Method



Annex D (informative)
Relationship of metadata items to SMPTE metadata dictionary RP 210

This Standard reflects established industry practice. The items described in Section 3 Definitions would be referred to as “metadata” items in Standards and Recommended Practices written later. An example is SMPTE Recommended Practice RP 210 Metadata Dictionary. This contains definitions of metadata items whose relationship to Section 3 Definitions may in some cases appear indirect. Other future SMPTE documents may also include such indirect relationships and may provide mapping from this document to a registry.

Annex E (informative)
Bibliography

SMPTE 96-1999, Television – 35- and 16-mm Motion- Picture Film – Scanned Image Area

SMPTE 170M-1999, Television — Composite Analog Video Signal — NTSC for Studio Applications

ITU-R Report 624-4 (MOD F), Characteristics of Television Systems

Tag Image File Format Specification, Revision 6.0, copyright 1986-1988, 1992, Aldus Corporation