Federal Agencies
Digital Guidelines Initiative

Guidelines for Embedded Metadata within DPX File Headers for Digitized Motion Picture Film

August 14, 2017

The FADGI Audio-Visual Working Group

http://www.digitizationguidelines.gov/audio-visual/

Guidelines for Embedded Metadata within DPX File Headers for Digitized Motion Picture Film

By the Federal Agencies Digital Guidelines Initiative (FADGI) Audio-Visual Working Group
http://www.digitizationguidelines.gov/audio-visual/

Date: August 14, 2017

Table of Contents

1 WHAT IS THIS DOCUMENT?

This document outlines embedded file metadata recommendations for the DPX (Digital Picture Exchange) format, most often used to store imagery data from scanned motion picture film. It is a companion to another product from the Federal Agencies Digital Guidelines Initiative (FADGI) Audio-Visual Working Group Film Scanning Subgroup efforts on digitized motion picture film:

- *Digitizing Motion Picture Film: Exploration of the Issues and Sample SOW.* (Updated April 2016) (http://www.digitizationguidelines.gov/guidelines/Motion_pic_film_scan.html)

This document works together with other FADGI projects related to embedded metadata and file identifiers in audiovisual files. The companion documents are:

- *Embedding Metadata in Digital Audio Files: Introductory Discussion for the Federal Agencies Guideline*. (Updated April 2012). http://www.digitizationguidelines.gov/audio-visual/documents/Embed_Intro_20120423.pdf
- *Discussion paper: Identifiers: Types and Characteristics* (Updated November 2011). http://www.digitizationguidelines.gov/audio-visual/documents/IdentifiersTypesCharacteristics_20111121.pdf

2 SCOPE OF THIS DOCUMENT

This document defines FADGI implementations for embedded metadata in DPX file headers including *Core* fields defined by SMPTE as well as selected fields *Strongly Recommended*, *Recommended* or *Optional* for FADGI use. The non-core fields take advantage of existing header structures as well as define new metadata fields for the User Defined fields to document, among other things, digitization process history.

This document is limited in scope to embedded metadata guidelines and does not look to define other technical characteristics of what a DPX file might carry including image tonal settings, aspect ratios, bit depths, color models and resolution. Recommended capture settings are defined for a variety of source material in the companion FADGI document, *Digitizing Motion Picture Film: Exploration of the Issues and Sample SOW.*[1]

3 MAKING THE CASE FOR EMBEDDED METADATA

FADGI defines embedded metadata as "a component of a digital file that exists alongside the content (usually binary data) within the file, making the digital file self-describing."[2] It is this ability to allow the file to declare what it is at a high level, where to get more information about it and in some cases how it can be used that is the real value add for embedded metadata.

[1] See http://www.digitizationguidelines.gov/guidelines/Motion_pic_film_scan.html.
[2] See http://www.digitizationguidelines.gov/term.php?term=metadataembedded

FADGI has had a vested interest in embedded metadata for some time. In addition to defining embedded metadata guidelines for Broadcast Wave files, FADGI has also provided embedded metadata guidelines for complex moving image and other audiovisual content in the MXF AS-07 Application Specification for Archiving and Preservation[3] as well as raster image TIFF files[4].

In 2012's *Embedding Metadata in Digital Audio Files: Introductory Discussion for the Federal Agencies Guideline*[5], Carl Fleischhauer states that

> Embedded metadata can provide information to and support functionality for various persons and systems at a variety of points in the content life cycle. For example, it can help the digitizing unit or organization as it produces and preserves content. It can serve persons or systems who receive content that is disseminated by the digitizing unit or organization. Some metadata elements are especially valuable to internal actors, some to external, and some to both. Embedded metadata, of course, is rarely an agency's only metadata. In most archiving and preservation programs, workflow and archiving are supported by one or more databases, cataloging systems, finding aids, and the like, each of which contains metadata. Many if not all metadata elements turn up in more than one place, a good thing since redundancy supports long-term preservation. (Being in more than one place, however, can make it difficult to update metadata across the board, unless this is supported in an automated way by an organization's technical infrastructure.)

Why embed metadata at all? In short, embedded metadata serves to describe, identify and track digital media across systems and platforms. The International Press Telecommunications Council Photo Metadata Working Group provides a series of use cases for embedded metadata as part of their *Embedded Metadata Manifesto*.[6]

At a high level, embedded metadata ensures against obsolesce due to lack of context. Files with a minimum amount of embedded metadata will retain their full informational value even if they were the only existing copy of an item, especially in the case of the catastrophic loss of an archive's other metadata resources. This minimum set of metadata might typically include one or more identifiers that link to the external metadata which in some cases will be more current than the embedded metadata, a statement of ownership or contact information, a creation date and perhaps copyright and use status if known. These data points would provide enough context to understand what the digital object is and where to get more information about it.

At a more detailed level, a more robust set of embedded metadata such as data about the composition and structure of the file, authenticity information, language tags and more allows for more reliable data exchange between digital tools and systems, especially over the long term. If a system knows what to expect, it can determine how to process the file and assure its continuity.

---

[3] See http://www.digitizationguidelines.gov/guidelines/MXF_app_spec.html
[4] See http://www.digitizationguidelines.gov/guidelines/digitize-tiff.html
[5] See http://www.digitizationguidelines.gov/audio-visual/documents/Embed_Intro_20120423.pdf
[6] See http://www.embeddedmetadata.org/metadata-use-cases-00.php and http://www.embeddedmetadata.org/embedded-metatdata-manifesto.php

Of course in order to be successful, as the IPTC Metadata Working Group says, "embedded metadata must be structured and persistent across time, use and systems. The type of content information carried in a metadata field, and the values assigned, should not depend on the technology used to embed metadata into a file. If multiple technologies are available for embedding the same field the software vendors must guarantee that the values are synchronized across the technologies without causing a loss of data or ambiguity."[7]

The other words, the added value of embedded metadata is wholly dependent on the processes to create, input and manage the data over time and across multiple platforms. There needs to be a common understanding of what data should go where, how that data should be structured and formatted and how disparate systems should interact with the data including not changing or deleting the data even if the current system doesn't understand or require the data. If the data isn't trusted as accurate and consistent, its value is greatly diminished. Guidelines such as this and others seek to help create stabile frameworks for reliable embedded metadata content.

4 OVERVIEW OF THE DPX FILE FORMAT

Digital Picture Exchange (DPX) is a pixel-based (raster) file format intended for very high quality moving image content with attributes defined in a binary file header. There are two versions of the DPX format, version 1 defined by SMPTE ST 268M-1994[8] and version 2 defined by SMPTE ST 268M-2003 and Amd. 1:2012.[9] DPX images are produced by scanning motion picture film[10] or by using a camera that produces a DPX output.

Each DPX file represents a single image or frame in a sequence of a motion picture or video data stream with a single component, e.g., luma, or multiple components, e.g., red, green, blue; or Cb, Y, Cr (chroma-luma data). Many variations in multiple component data are supported. As a structured raster image format, DPX is intended to carry only picture or imagery data with corresponding sound carried in a separate format, typically WAVE files.[11] In practice, this means that a single digitized motion picture film will consists of a sequence of tens of thousands of individual DPX files, each file corresponding to a frame of scanned film with sequentially numbered file names as well as a separate audio file for sound data.

5 DPX FILE STRUCTURE

A DPX file has four sections (including one optional section):

- *Generic image data* (including file information, image information, and image orientation information). Within the "Generic image data" section, a set of identified "core" fields is required. This core set comprises a minimum amount of information that a reader needs to read and interpret. A core-compliant reader must read the core fields but not

[7] Ibid
[8] This version of the specification is no longer available from SMPTE.
[9] For an explanation of the differences between the two versions, see http://www.digitalpreservation.gov/formats/fdd/fdd000178.shtml#notes
[10] See Digitizing Motion Picture Film: Exploration of the Issues and Sample SOW: http://www.digitizationguidelines.gov/guidelines/Motion_pic_film_scan.html
[11] See http://www.digitalpreservation.gov/formats/fdd/fdd000001.shtml

necessarily be able to read non-core fields. A core-compliant writer must fill in the fore fields with valid values because blank or undefined values are not permitted. Non-core fields must be filled in with UNDEFINED values if the correct values are not known. In v.1 (268M-1994), this is further outlined in section 3; in v.2 (268M-2003), this is further detailed in section 4. The core fields are: magic number ("SDPX" or "XPDS"), offset to image data in bytes, version number ("V2.0" for version 2), total image size in bytes (including file header), image orientation, number of image elements, pixels per line, lines per image element, data sign, descriptor, transfer characteristic, colorimetric specification, bit depth, packing, encoding, and offset to data.

- *Industry Specific Information* (motion picture film information, television information) which generally describes the film and camera source from which the image frame data was derived. The motion-picture film information header outlined in section 6.1 includes helpful tags for film-related data including information derived from film edge codes such as film manufacturing ID code, film type, offset in perforations, prefix and count. Other structured data fields include format (e.g. Academy), frame position in sequence (incremental integer sequence of DPX files), sequence length (frames), held count (1 = default), frame rate of original (frames/s), shutter angle of camera in degrees, frame identification (e.g. keyframe), and slate information.
- *User defined data* (optional). Format and structure is not defined in SMPTE ST 268M-2003 to allow for customized information needed by some users but typical uses might be processing logs, etc.
- *Image data.* SMPTE ST 268M:2003 does not include more specific information about the expected contents of this section aside from specifying "Image 8-K blocks are recommended for efficient use of tape-storage devices."

6 SMPTE CORE FIELDS

To provide a streamlined path for implementation and testing, SMPTE defines a core set of required fields which contains the minimum amount of information that a DPX reader needs to read and interpret a file. A list of the core fields is included in section 9 of this document, FADGI Use of SMPTE Core Fields.

According to SMPTE ST268, a core-compliant reader must read the core fields, but need not read the others; and 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.

7 ANALYSIS OF EMBEDDED METADATA IN SAMPLE FILES

Analysis of sample DPX files submitted by the FADGI community demonstrates that implementation of core and non-core fields are not always consistent according to the SMPTE ST268 specification set. The sample files were created by film scanners from a range of well-known manufactures and contain a variety of imagery content, resolutions, bit depths and more. Post capture workflows varied with only one FADGI member embedding metadata after scanning.

The non-conformance of core fields is the most problematic because, as mentioned above, a core

field contains the minimum amount of information that a DPX reader needs to read and interpret a file. If this minimum data is not present or incorrect, the long term sustainability of the file is in jeopardy.

A summary of the file analysis is available in the Appendix A and the anonymized raw data is available as an.xlsx file for download.[12]

Among core fields, non-conformance was demonstrated for the fields in Table 1 shown below.

Field number	Content	Summary of Analysis
2	Offset to image data in bytes	Inconsistent. Sometimes used correctly, sometimes this data is recorded in field 19 or 21.12 *See 7.1.3 for more information on this field's relationship to field 77, Image Data Block Boundary
4	Total image file size in bytes (including file header)	Inconsistent. Sometimes used correctly, sometimes blank
21.12	Offset to image data	Inconsistent. Sometimes used correctly, sometimes this data is recorded in field 2 or 19

Table 1. SMPTE Core Field Issues

Among non-core fields, one notable issue is that no non-core fields contain the value UNDEFINED the correct value is not known as defined by the specification. Typically, the field is blank or holds a null value. Other notable non-core issues are listed below in Table 2.

Field number	Content	Summary of Analysis
10	Creation date/time: yyyy:mm:dd:hh:mm:ssLTZ	Inconsistent. Sometimes used correctly but often blank or the formatting is incorrect with dashes instead of colons or data in the wrong order. This value is hard to determine in post-production so embedding at time of creation would be the best course of action. *See 7.1.2 for more detailed explanation on data/time formatting issues.
49	Format - e.g. Academy	Inconsistent. Sometimes used as summary for source format
56	Slate information	Inconsistent. Sometimes used correctly, sometimes used as summary of source format
66	Temporal sampling rate or	Inconsistent. Sometimes replicates data in field

[12] Please note that both the institution and manufacturer data are anonymized. The spreadsheet is quite large and would be challenging to reduce to a readable version for this report: http://www.digitizationguidelines.gov/audio-visual/documents/DPX-sampleAnalysis-2016-condensed-anon.xlsx

	frame rate (Hz)	53, sometimes blank

Table 2. Selected Non-Core Field Issues

Perhaps a contributing factor to non-compliance is the lack of definitions for each field listed in the specification. If the use and expected values of each field were declared, some of the confusion might be reduced. For fields of interest to FADGI, FADGI has created definitions for correct field use. FADGI also augments SMPTE definitions when needed.

7.1 ST 268 Data Definition Issues for Core and Non-Core Fields of Interest to FADGI

7.1.1 Magic Number Data Type Mismatch

ST 268 section 5.1 declares that the Magic Number (field 1) for DPX is U32 data type but this incorrect. The data type should be labelled as ASCII.

7.1.2 DATE/TIME Data Formatting Issues

The ST 268 instructions for fields that contain date and time data are problematic because although ST 268 2003 section 3.4 states that the Creation date/time (field 10) and Source data/time (field 37) data fields are to be formatted according to ISO 8601:2004(E), the listed value options are not entirely consistent with ISO 8601.

> **Issue 1:** SMPTE ST 268 requires the use of colons [:] between data components except for between seconds and the timezone designation between which, there is no separator.
>
> ISO 8601 however allows for either no data component separation marker (example: YYYYMMDDhhmmss) or a combination of hyphens [-] between date components followed by colons [:] separating the time components (example: YYYY-MM-DDThh:mm:ss). Colons [:] are used to separate the time elements "hour" and "minute", and "minute" and "second" whereas hyphens [-] are used to separate the time elements "year" and "month", "year" and "week", "year" and "day", "month" and "day", and "week" and "day". ISO 8601 does not permit colons to separate date components as ST 268 specifies.
>
> **Issue 2:** ST 268:2003 section 3.4 defines the “LTZ” component at the end of the string as “Local Time Zone” although the LTZ value is not defined in ISO 8601:
>
> > *LTZ means "Local Time Zone;" format is:*
> > *LTZ = Z (time zone = UTC), or LTZ = +/-hh, or LTZ = +/-hhmm (local time is offset from UTC)*
>
> For time of day, ISO 8601 expresses this value in terms of UTC (Coordinated Universal Time or Temps Universel Coordonné) with the option to map offset of the local time to UTC:
>
> *When it is required to indicate the difference between local time and UTC, the representation of the difference can be expressed in hours and minutes, or hours only. It shall be expressed as positive (i.e. with the leading plus sign [+]) if the local time is ahead of or equal to UTC and as negative (i.e. with the leading minus sign [–]) if it is behind*

UTC. The minutes component of the difference may only be omitted if the time difference is exactly an integral number of hours (ISO 8601 2000(E) section 5.3.4.1)

The results of this non-conformity to the defined standard means that the data is very often not formatted consistently according to neither the SMPTE ST 268 instructions nor ISO 8601 restrictions. The variability of the data means it can't be parsed correctly by downstream systems and will force errors in QA/QC systems. Following the standard ISO 8601 formatting will allow the data to be integrated into other systems with fewer migration and validation issues.

As described in following sections, Creation date/time: yyyy:mm:dd:hh:mm:ssLTZ (field 10) and Source image date/time: yyyy:mm:dd:hh:mm:ssLTZ (field 37), FADGI recommends following the published and standardized ISO 8601 rules for documenting date and time data instead of following the ST 268 rules. In addition, FADGI advocates that SMPTE update the ST 268 data and time data formatting rules in the next planned revision to confirm more strongly to ISO 8601 rules.

7.1.3 Image Data Offset Mismatches

Although SMPTE ST 268 does not provide a definition, Offset to Image Data (field 2) is the declaration of the byte offset to the beginning of the image data counting up from 0 bytes at the start of the file. If there is only one image element, this value is typically repeated in field 21.12, Offset to data. In theory, this value should match the byte offset value of the appearance of the image data in the file as described in field 77 which states that "Image data should start at block boundary (8-K blocks are recommended for efficient use of tape-storage devices)." In the sample files, this is often not the case. For example, in sample file[13] 1199WP_00100000.dpx from "Institution 1; Scanner Brand A; Outsourced Vendor," image data should start at byte offset 8196 as declared in field 2 but doesn't start until byte offset 31188. Two other files from the same institution, scanner and DPX image sequence were correctly parsed.

7.1.4 Byte Offset Overruns

An enduring issue with overall metadata compliance conformance checking is that often metadata fields overrun their allotted byte range, either starting before the defined offset or ending well after the defined offset. For example, in sample files from "Institution 3; Scanner Brand C; In-house[14]" data for Image Filename (field 9) started at byte offset 24, not 36 as defined.

User defined data fields can be especially problematic. As demonstrated in "Institution 2; Scanner Brand A; Outsourced Vendor", data for User Identification (field 75) started at byte offset 3072, whereas SMPTE defines the data range for this data as offset 2048-2079.

Inconsistencies such as these make automated conformance checking extremely difficult because the data is not where it is supposed to be according to the defined ruleset.

8 OVERVIEW OF FADGI EMBEDDED METADATA GUIDELINES FOR DPX

The FADGI Audio-Visual Working Group has defined guidelines embedded metadata in DPX file headers including *core* fields defined by SMPTE as well as fields *Strongly Recommended*,

[13] Ibid
[14] Ibid

Recommended or *Optional* for FADGI use. A summary by obligation is available in Appendix B. The non-core fields take advantage of existing header structures as well as define new metadata fields for the User Defined fields to document, among other things, digitization process history.

It should be noted that FADGI does not assign an obligation for ALL fields outlined in SMPTE ST268. Non-core fields not discussed in this guideline are left unchanged.

For all data fields addressed in this guideline, FADGI has included definitions including for those fields which do not have a definition in SMPTE ST268. For all core fields, the FADGI use complies with the SMPTE use. In non-core fields, especially those with no definition from SMPTE, FADGI defines the use. One example is the Creator field (field 12) which FADGI proposes to be used for the name of the institution or entity responsible for the creation, maintenance, and preservation of this digital item. This use aligns with other FADGI embedded metadata guidelines for BWF and MXF AS-07 files.[15]

FADGI draws inspiration from EBU R98-1999: Format for the <CodingHistory> field in Broadcast Wave Format[16] document for defining a use for field 76, *User defined data header*, to summarize data on the digitizing process including signal chain specifics and other elements. The Digitization process history field employs a defined string variable for each parameter of the digitization process: the first line documents the source film reel, the second line contains data on the capture process, the third line of data records information on the storage of the file.

8.1 Explanation of Data Table Descriptions

Name of Data Field: Identification of content within Header Data Field as defined by ST 268	
SMPTE ST 268 Definition	Description of Data Field from ST 268 1998 or ST 268: 2003. Not all Data Fields have definitions or descriptions of the content.
FADGI Use	Description of FADGI implementation of Data Field. When ST 268 does not include a description of the data in the field, FADGI provides a definition. Where FADGI Use differs from the SMPTE defined use, this is noted.
Field number	The sequential one-up label number for each data field assigned by SMPTE starting at 1.

[15] BWF: Originator (bext chunk) (see http://www.digitizationguidelines.gov/audio-visual/documents/Embed_Guideline_20120423.pdf); MXF AS-07: AS_07_Core_DMS_ResponsibleOrganizationName; AS_07_Core_DMS_ResponsibleOrganizationCode (see http://www.amwa.tv/downloads/specifications/AS-07_Proposed_Application_Specification.pdf)
[16] See http://tech.ebu.ch/docs/r/r098.pdf

<table>
<tr><td>Byte offset start</td><td>The first byte of the data field counting up from zero bytes at the beginning of the file.</td></tr>
<tr><td>Byte offset end</td><td>The last byte of the data field counting up from zero bytes at the beginning of the file.</td></tr>
<tr><td>Byte limit</td><td>The number of bytes allocated to the data field as permitted by ST 268; the difference between the Byte offset start and Byte offset end.</td></tr>
<tr><td>Data type</td><td>The classification or category of data permitted in the Values field and defined in section 4.2 of ST 268:2003. Typical examples include</td></tr>
<tr><td>Header location</td><td>Identification of the location within file in which the data occurs:
Generic file header: file information, image information, data format, and image origination information
Motion-picture and television industry-specific information;
User-defined information;
Image data</td></tr>
<tr><td>Mandatory/optional</td><td>The requirement for the file to contain the data. All SMPTE CORE files are required. In addition, FADGI requires additional fields for FADGI compliance.</td></tr>
<tr><td>Values</td><td>Describes the allowable values for the data field as defined by ST 268:<table><tr><th>Data Type</th><th>Undefined Value</th></tr><tr><td>U8 unsigned 8-bit integer</td><td>FF hex</td></tr><tr><td>U16 unsigned 16-bit integer</td><td>FFFF hex</td></tr><tr><td>U32 unsigned 32-bit integer</td><td>FFFFFFFF hex</td></tr><tr><td>R32 32-bit real number (IEEE floating point)</td><td>FFFFFFFF hex</td></tr><tr><td>ASCII</td><td>0 hex (NULL character)</td></tr></table></td></tr>
</table>

9 FADGI USE OF SMPTE CORE FIELDS

Magic number	
SMPTE ST 268 Definition	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.
FADGI Use	Same as SMPTE ST 268
Field number	1
Byte offset start	0
Byte offset end	3
Byte limit	4
Data type	ASCII (ST 268 declares that the Magic Number for DPX is U32 data type but this incorrect. The data type should be labelled as ASCII.)
Header location	File information header
Mandatory/optional	Required by SMPTE ST 268
Values	SDPX; XDPX

Offset to image data in bytes	
SMPTE ST 268 Definition	*[ST268 has no formal definition for this field]*
FADGI Use	Same intended use as SMPTE ST 268. Declaration of the byte offset to the beginning of the image data counting up from 0 bytes at the start of the file. Note: If there is only one image element, this value is typically repeated in field 21.12, Offset to data.
Field number	2
Byte offset start	4
Byte offset end	7

Byte limit	4
Data type	U32
Header location	File information header
Mandatory/optional	Required by SMPTE ST 268
Example	00 00 08 00 [translated from hex values = 2048]

Version number of header format (V1.0 or V2.0)	
SMPTE ST 268 Definition	*[ST268 has no formal definition for this field]*
FADGI Use	Same intended use as SMPTE ST 268. Declaration of the version number of the DPX format. There are currently two versions of DPX: Version 1 (V1.0) defined by ST 268:1998 and Version 2 (V2.0) defined by ST 268:2003 and Amd 1.
Field number	3
Byte offset start	8
Byte offset end	15
Byte limit	8
Data type	ASCII
Header location	File information header
Mandatory/optional	Required by SMPTE ST 268
Values permitted by ST 268	V1.0 V2.0

Total image file size in bytes (including file header)	
SMPTE ST 268 Definition	Indicates the size of the entire file, i.e. containing both header and image data.
FADGI Use	Same use as SMPTE ST 268.
Field number	4
Byte offset start	16
Byte offset end	19
Byte limit	4
Data type	U32
Header location	File information header
Mandatory/optional	Required by SMPTE ST 268
Example	00 01 00 00 [translated from hex values = 65536]

Image orientation	
SMPTE ST 268 Definition	Indicates the orientation of the image data required for display. The possible orientations are listed in ST 268 table 2. The standard orientation for core set images (code 0) is left to right (line direction) and top to bottom (frame direction).
FADGI Use	Same use as SMPTE ST 268.
Field number	17
Byte offset start	768
Byte offset end	769
Byte limit	2
Data type	U16
Header location	Image information header
Mandatory/optional	Required by SMPTE ST 268
Example	00 00
Values	From SMPTE ST 268 Table 2. 0 = Left to right Top to bottom (Orientation 0 is the only one supported in the core set file format.) 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

Number of image elements (1 - 8)	
SMPTE ST 268 Definition	Each file represents a single image with up to eight (8) image elements. Image elements are defined as a single component (e.g. luma) or multiple components (e.g. red, green, and blue). *NB: This field declares how many image elements are present in the file.*
FADGI Use	Same intended use as SMPTE ST 268. Declaration of the number of image elements present and described in fields 21 and subfields.
Field number	18
Byte offset start	770

Byte offset end	771
Byte limit	2
Data type	U16
Header location	Image information header
Mandatory/optional	Required by SMPTE ST 268
Example	00 01 [one image element described in fields 21 and subfields]

Pixels per line	
SMPTE ST 268 Definition	Active number of pixels per line
FADGI Use	Same use as SMPTE ST 268.
Field number	19
Byte offset start	772
Byte offset end	775
Byte limit	4
Data type	U32
Header location	Image information header
Mandatory/optional	Required by SMPTE ST 268
Example	00 00 08 00 [translated from hex values = 2048 for 2K]
Example	00 00 10 00 [translated from hex values = 4096 for 4K]

Lines per image element	
SMPTE ST 268 Definition	*[ST268 has no formal definition for this field]*
FADGI Use	Active number of lines for each image element.
Field number	20
Byte offset start	776
Byte offset end	779
Byte limit	4
Data type	U32
Header location	Image information header
Mandatory/optional	Required by SMPTE ST 268
Example	00 00 06 14 [translated from hex values = 1556 for 2K]
Example	00 00 08 70 [translated from hex values = 2160 for 4K]

Data sign	
SMPTE ST 268 Definition	*[ST268 has no formal definition for this field]*
FADGI Use	Identification of the image's data type. Signed data can hold both positive and negative values while unsigned types can hold large positive values but cannot hold negative values. DPX Core set images are unsigned by default as specified in ST 268.
Field number	21.1
Byte offset start	780
Byte offset end	783
Byte limit	4
Data type	U32
Header location	Image information header: Data structure for image element
Mandatory/optional	Required by SMPTE ST 268
Values	0 = unsigned; 1 = signed; Core set images are unsigned.

Descriptor	
SMPTE ST 268 Definition	Defines the components that make up an image element and their pixel packing order. The valid components are listed in table 1.
FADGI Use	Same use as SMPTE ST 268.
Field number	21.6
Byte offset start	800
Byte offset end	801
Byte limit	1
Data type	U8
Header location	Image information header: Data structure for image element
Mandatory/optional	Required by SMPTE ST 268

Values	From SMPTE ST 268 table 1. 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 (CB, CR, 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 = CB, Y, CR, Y (4:2:2) ---- based on SMPTE 125M 101 = CB, Y, A, CR, Y, A (4:2:2:4) 102 = CB, Y, CR (4:4:4) 103 = CB, Y, CR, A (4:4:4:4) 104 -149 = Reserved for future cBYeR ++ 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 a-component element 157 - 254 = Reserved for future formats

Transfer characteristic	
SMPTE ST 268 Definition	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.
FADGI Use	Same use as SMPTE ST 268.
Field number	21.7
Byte offset start	801
Byte offset end	802
Byte limit	1
Data type	U8
Header location	Image information header: Data structure for image element

Mandatory/optional	Required by SMPTE ST 268
Values	Listed in SMPTE ST 268M-2003 Amendment 1:2012, table 5A. 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 = SMPTE ST 2065-3 Academy Density Exchange Encoding (ADX) 14-254 = Reserved for future use

Colorimetric specification	
SMPTE ST 268 Definition	Defines the appropriate color reference primaries (for additive color systems like television) or color responses (for printing density).
FADGI Use	Same use as SMPTE ST 268.
Field number	21.8
Byte offset start	802
Byte offset end	803
Byte limit	1
Data type	U8
Header location	Image information header: Data structure for image element
Mandatory/optional	Required by SMPTE ST 268

Values	Listed in SMPTE ST 268M-2003 Amendment 1:2012, table 5B. 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 = SMPTE ST 2065-3 Academy Density Exchange Encoding (ADX) 14-254 = Reserved for future use

Bit depth	
SMPTE ST 268 Definition	Defines the number of bits for each component in the image element. All components must have the same bit depth.
FADGI Use	Same use as SMPTE ST 268.
Field number	21.9
Byte offset start	803
Byte offset end	804
Byte limit	1
Data type	U8
Header location	Image information header: Data structure for image element
Mandatory/optional	Required by SMPTE ST 268
Values	Valid bit depths are 1-, 8-, 10-, 12-, and 16-bit integer, and 32- and 64-bit IEEE floating point (see table 3A) as listed in ST 268 table 3A.

Packing	
SMPTE ST 268 Definition	For image element n, defines the data packing mode.
FADGI Use	Same use as SMPTE ST 268.
Field number	21.10
Byte offset start	804
Byte offset end	805
Byte limit	2
Data type	U16
Header location	Image information header: Data structure for image element
Mandatory/optional	Required by SMPTE ST 268
Values	Listed in ST 268 table 3B. 0 = Packed into 32-bit words 1 = Filled to 32-bit words, method A (note: this is standard) 2 = Filled to 32-bit words, method B (note: this is non-standard) 3 = Reserved for future use

Encoding	
SMPTE ST 268 Definition	For image element n, defines whether or not the element is run-length encoded..
FADGI Use	Same use as SMPTE ST 268.
Field number	21.11
Byte offset start	806
Byte offset end	807
Byte limit	2
Data type	U16
Header location	Image information header: Data structure for image element
Mandatory/optional	Required by SMPTE ST 268
Values	Listed in ST 268 table 3C. 0 = No encoding applied 1 = Run-length encoded 2-7 = Reserved for future use

Offset to data	
SMPTE ST 268 Definition	To data for image element n, defines the offset in bytes to the image data for element n from the beginning of the file.
FADGI Use	Same use as SMPTE ST 268. Note: Field is required as a CORE field but is especially helpful when there is more than one image element in the file. Otherwise, the data is typically a repeat of the data in field 2, Offset to image data in bytes.
Field number	21.12
Byte offset start	808
Byte offset end	811
Byte limit	4
Data type	U32
Header location	Image information header: Data structure for image element
Mandatory/optional	Required by SMPTE ST 268
Example	00 00 08 00 [translated from hex values = 2048]

10 FADGI STRONGLY RECOMMENDED FIELDS

If the Working Group had the authority to do so, these would be "required" although they are not core or required fields according the SMPTE ST268.

Image filename	
SMPTE ST 268 Definition	*[ST268 has no formal definition for this field]*
FADGI Use	This element should match the filename of the file. The filename must differentiate one DPX frame from another within the same DPX sequence at the file level. Filenames should be unique within the same DPX sequence and ideally within the collection and holding institution.
Field number	10
Byte offset start	36
Byte offset end	135
Byte limit	100
Data type	ASCII string

Mandatory/optional	Strongly recommended (if the Working Group had authority: "required")
Values	Free text. Filename.extension. Follow widely accepted filenaming protocols to avoid spaces and special characters in filename.
Example LC/AFC	Ranch_Pictures_Reel_One_DPX00086424.dpx
Example NMAAHC	[HouseRuleDesignator]_0086400.dpx
Notes	*Film scanners typically include an option to auto-embed this at time of capture. This also could be batch embedded post-process, especially if there's a desire to include a preface to the file name or change information in the file name (e.g., to include project name or file path for example).*

Creation date/time: yyyy:mm:dd:hh:mm:ssLTZ	
SMPTE ST 268 Definition	Defined as yyyy:mm:dd:hh:mm:ssLTZ
FADGI Use	This data element contains the digital file creation date and time, not of the creation of the source object which is recorded in SourceDateTime (field 37). FADGI advises that data and timed data follow ISO 8601 formatting rather than the non-standard formatting that is defined in ST 268 2003. Following the standard ISO 8601 formatting will allow the data to be integrated into other systems with fewer migration and validation issues. See 7.1.2 DATE/TIME Data Formatting Issues for more information. Specifically, FADGI recommends the use of ISO 8601 data component separators: colons [:] are used to separate the time elements "hour" and "minute", and "minute" and "second"; and hyphens [-] are used to separate the time elements "year" and "month", "year" and "week", "year" and "day", "month" and "day", and "week" and "day". In addition, while ST 268 declares “LTZ” or “local time zone” for entity or institution that created the file, FADGI recommends that the creation time is recorded as Coordinated Universal Time (UTC) as defined in ISO 8601:2004(E) section 4.2.4. Finally, for full expressions of data with both date and time information, FADGI recommends the character [T] be used as time designator to indicate the start of the representation of the time of day component . When the enter date/time string is unknown, ISO 8601 allows for any number of values to be dropped from the date/time representations but data must be in the order from the least-to-most significance or precision.
Field number	10
Byte offset start	136
Byte offset end	159
Byte limit	24
Data type	ASCII string

Mandatory/optional	Strongly recommended (if the Working Group had authority: "required")
Values	YYYY-MM-DDThh:mm:ssZ Year is defined from 0000 to 9999; Month is defined from 01 to 12 (use leading zeroes if less than 10); Day is defined from 01 to 28, 29, 30 or 31 (use leading zeroes if less than 10); Hour is defined from 00 through 24 (am/pm NOT allowed); Minute is defined from 00 through 59; Second is defined from 00 through 60. The UTC designator [Z] if use is made of UTC of day. The character [T] shall be used as time designator to indicate the start of the representation of the time of day component in these expressions. The hyphen [-] and the colon [:] shall be used as separators within the date and time of day expressions, respectively, when required. Provide as much accurate data as is available, making use of abbreviated implementations of the full ISO date/time string. Examples are acceptable abbreviated strings are available: https://www.w3.org/TR/NOTE-datetime
Example	2016-03-22T19:07:12Z [full string]
Example	2016-12-01 [only year month day known]
Example	2016-12 [only year month known]
Example	2016 [only year known]
Notes	*This field should be populated by data generated at time of capture via the film scanner. Sample file analysis demonstrates that while is data is often present in the field, it may not follow ISO 8601 formatting rules or ST 268 formatting rules to the letter. This data would not be easy to batch embed post process.*

Creator	
SMPTE ST 268 Definition	*[ST268 has no formal definition for this field]*

FADGI Use	This element contains the institution or entity responsible for the creation, maintenance, and preservation of this digital item. Entity designations should be as specific as possible including a two-character county code to avoid the potential for conflict in the responsible organization's name. If space permits within the 100 byte limit, the archival entity should be identified at the most specific level within the institution. Use a standard abbreviation of entity names such as those found in the Guide to Government Acronyms & Abbreviations. If an entity is not on this list, use a familiar abbreviation. Use the standard two-character ISO 3166 alpha 2 country code list. NB: Information about the film scanner is recorded in Input Device Name and Input Device Serial Number (fields 38 and 39)
Field number	12
Byte offset start	160
Byte offset end	259
Byte limit	100
Data type	ASCII string
Mandatory/optional	Strongly recommended (if the Working Group had authority: "required")
Values	[Country code]comma space[Entity name]
Example LC	US, LOC/AFC [AFC = American Folklife Center]
Example NARA	US, NARA
Example NMAAHC	US, NMAAHC

Project name	
SMPTE ST 268 Definition	*[ST268 has no formal definition for this field]*

FADGI Use	This element is recommended as a container for identifiers for the work represented by the DPX sequence at hand and/or as pointers to additional, non-embedded (externally maintained) metadata. Members of the Working Group have repeatedly encountered the need to provide multiple identifiers for a given item. In some cases, the 200 byte limit will prevent an agency from listing all of its identifiers; the most important or helpful should be provided starting with the principal identifier or the "best" identifier which uniquely differentiates one object from another. For more information about identifiers, see FADGI's 2012 document Identifiers: Types and Characteristics Do not embed identifiers that could pose a possible security risk, e.g., by exposing exact pathnames. NOTE: The Working Group perceived value in the practice of tagging of identifiers (see examples, typically URLs) to permit them to be properly understood but wished to leave this as optional.
Field number	13
Byte offset start	260
Byte offset end	459
Byte limit	200
Mandatory/optional	Strongly recommended (if the Working Group had authority: "required")
Values	If labeled: Identifier [comma space] type [comma space] comment [semicolon-space if more than one identifier] If no labeling: Identifier
Example NMAAHC	2012.79.1.16.1a, Alice Coltrane, Pearl Bowser collection Object component ID #, Title, Local collection name

11 FADGI RECOMMENDED FIELDS

Right to use or copyright statement	
SMPTE ST 268 Definition	*[ST268 has no formal definition for this field]*

FADGI Use	Information about copyright and other restrictions (donor, privacy, etc.). Usage by federal agencies will often refer to the documentation of restrictions provided by other, non-embedded metadata. If used, this element may contain the information as known at the time of embedding. The Working Group understands that metadata may be updated over time. Users who refer to an embedded block of metadata should understand that this may not represent the latest and best information, and they should be counseled to follow one of the identifiers in order to obtain later and/or better data.
Field number	14
Byte offset start	460
Byte offset end	659
Byte limit	200
Data type	ASCII
Mandatory/optional	Recommended for FADGI use
Values	Free text. If there are multiple copyrights or other restriction statements, separate them by a semicolon followed by a space.
Example LC	See Copyright Restriction Statement. [Used by LC, American Folklife Center]
Example NMAAHC	To copy or reproduce this asset in any form and/or to obtain credit information, contact Smithsonian NMAAHC Rights and Reproductions department.

Source image date/time: yyyy:mm:dd:hh:mm:ssLTZ	
SMPTE ST 268 Definition	Defines the creation time of the source image from which the image was extracted or processed. Formatting is as per clause 3.4.

FADGI Use	This data element contains the digital file creation date and time, not of the creation of the source object which is recorded in SourceDateTime (field 37). FADGI advises that data and timed data follow ISO 8601 formatting rather than the non-standard formatting that is defined in ST 268 2003. Following the standard ISO 8601 formatting will allow the data to be integrated into other systems with fewer migration and validation issues. See 7.1.2 DATE/TIME Data Formatting Issues for more information. Specifically, FADGI recommends the use of ISO 8601 data component separators: colons [:] are used to separate the time elements "hour" and "minute", and "minute" and "second"; and hyphens [-] are used to separate the time elements "year" and "month", "year" and "week", "year" and "day", "month" and "day", and "week" and "day". In addition, while ST 268 declares “LTZ” or “local time zone” for entity or institution that created the file, FADGI recommends that the creation time is recorded as Coordinated Universal Time (UTC) as defined in ISO 8601:2004(E) section 4.2.4.
Field number	37
Byte offset start	1532
Byte offset end	1555
Byte limit	24
Data type	ASCII string
Mandatory/optional	Recommended for FADGI use

Values	YYYY-MM-DDThh:mm:ssZ Year is defined from 0000 to 9999; Month is defined from 01 to 12 (use leading zeroes if less than 10); Day is defined from 01 to 28, 29, 30 or 31 (use leading zeroes if less than 10); Hour is defined from 00 through 24 (am/pm NOT allowed); Minute is defined from 00 through 59; Second is defined from 00 through 60. The UTC designator [Z] if use is made of UTC of day. The character [T] shall be used as time designator to indicate the start of the representation of the time of day component in these expressions. The hyphen [-] and the colon [:] shall be used as separators within the date and time of day expressions, respectively, when required. Provide as much accurate data as is available, making use of abbreviated implementations of the full ISO date/time string. Examples are acceptable abbreviated strings are available: https://www.w3.org/TR/NOTE-datetime
Example	2016-03-22T19:07:12Z [full string]
Example	2016-12-01 [only year month day known]
Example	2016-12 [only year month known]
Example	2016 [only year known]

Input device name	
SMPTE ST 268 Definition	*[ST268 has no formal definition for this field]*
FADGI Use	Manufacturer name and model name of film scanner that digitally scanned the motion picture film and produced the DPX file.
Field number	38
Byte offset start	1556
Byte offset end	1587
Byte limit	32
Data type	ASCII string
Mandatory/optional	Recommended for FADGI use
Values	Free text.
Example NARA	Scanner4K (anonymized scanner name)

Input device serial number	
SMPTE ST 268 Definition	*[ST268 has no formal definition for this field]*
FADGI Use	The serial number of film scanner named in Input Device Name (field 38) that digitally scanned the motion picture film and produced the DPX file.
Field number	39
Byte offset start	1588
Byte offset end	1619
Byte limit	32
Data type	ASCII string
Mandatory/optional	Recommended for FADGI use
Values	Free text.
Example	LSQ7D5LPS1

Frame position in sequence	
SMPTE ST 268 Definition	Defines the frame number in the image sequence. 4 byte U32 unsigned 32-bit integer.
FADGI Use	Same as ST 268. Defines the frame number within the sequence of DPX files.
Field number	50
Byte offset start	1712
Byte offset end	1715
Byte limit	4
Data type	U32 unsigned 32-bit integer
Mandatory/optional	Recommended for FADGI use
Values	This number is often the same as the file name when auto assigned as a one-up integer during the digitization process. NOTE: The values are captured as hex values so would need to be translated to decimal values to compare against the auto assigned film name.
Example NMAAHC	00 01 5F 90 [translated from hex values = 90000 decimal value]

Sequence length (frames)	
SMPTE ST 268 Definition	Defines the total number of frames in the image sequence. 4 byte U32 unsigned 32-bit integer
FADGI Use	Same as ST 268. Defines the total number of frames in the DPX sequence. Sequence length and frames per second can be used to derive duration/run time.
Field number	51
Byte offset start	1716
Byte offset end	1719
Byte limit	4
Data type	U32 unsigned 32-bit integer
Mandatory/optional	Recommended for FADGI use
Values	This value would be derived post-capture, after the entire sequence has been processed.
Example	01 00 00 00 [translated from hex values = 16777216 decimal value]

12 FADGI OPTIONAL FIELDS

Film mfg. 10 code (2 digits from film edge code)	
SMPTE ST 268 Definition	Encodes data from machine readable portion of film edge code, according to SMPTE ST 254:2008 [Motion-Picture Film (35-mm) — Manufacturer-Printed Latent-Image Identification Information.]
FADGI Use	The 2 digit code to identify the manufacturer of the digitized film
Field number	43
Byte offset start	1664
Byte offset end	1665
Byte limit	2
Data type	ASCII string
Mandatory/optional	Optional for FADGI use

Values	Values from SMPTE ST 254:2008, table 3 01 = AGFA-GEVAERT, N.V 02 = EASTMAN KODAK CO 03 = FUJIFILM CO 04 = ILFORD LTD 00 = Other or nondesignated
Example	02 53 = Kodak Acetate-base 35 mm (or larger) lab/print film [example includes film manufacturer and film type]

Film type (2 digits from film edge code)	
SMPTE ST 268 Definition	Encodes data from machine readable portion of film edge code, according to SMPTE ST 254:2008 Motion-Picture Film (35-mm) — Manufacturer-Printed Latent-Image Identification Information.
FADGI Use	The 2 digit code to identify the type of film that was digitized. Although the data represented by the code varies by manufacturer, the code may define information about the gauge, emulsion, and film base and more.
Field number	44
Byte offset start	1666
Byte offset end	1667
Byte limit	2
Data type	ASCII string
Mandatory/optional	Optional for FADGI use
Values	The second character shall be a two-digit product specification code assigned at the discretion of the manufacturer. If the manufacturer does not wish to identify the product, the digits 00 shall be encoded. A list of film stocks is available on Wikipedia: https://en.wikipedia.org/wiki/List_of_motion_picture_film_stocks
Example	02 53 = Kodak Acetate-base 35 mm (or larger) lab/print film [film manufacturer and film type]

12.1 Implementation of User Defined Data Field for FADGI's Digitization Process History Information

Valid implementation of User Defined Data requires a trilogy of connected fields:

- Field 8 | Offset 32 | Length 4 | Type U32 | User-defined header length in bytes
- Field 75 | Offset 2048 | Length 32 | Type ASCII | User identification
- Field 76 | Offset 2080 | Length XX | Type TBD | User Defined - can be anything up to 1 Mbyte

Field 8 must equal the sum of Field 75 + Field 76.

Note: To the best of FADGI's knowledge, only one instance of User Defined data is permitted per file because the User Identification (field 75) is only declared once at offset 2048, although in theory, multiple identifiers could be contained with the 32 bytes allotted. Practically, it might not be feasible to have multiple blocks of data within the User Defined fields.

When data already exists in the User Defined fields (fields 75 and 76), a user must decide if the data should be kept or whether the user should replace that data with FADGI process history data. Perhaps a future tool could help the user export the existing data (or wrap the existing data into the FADGI process history) for future use.

Comments welcome!

FADGI's implementation of the User Defined Data field creates the structure to document digitization process history using coded values to summarize the source film reel, data on the capture process and data about the characteristics of the digital file. The User Identification field (field 75) labels the owner or type of information that exists in the User Defined field (field 76). When implementing FADGI's Digitization Process History (field 76) structure, the value of User identification (field 75) should be "FADGI digitization process history".

User identification	
SMPTE ST 268 Definition	*[ST268 has no formal definition for this field]*
FADGI Use	Label to identify the FADGI Digitization Process History structured data block in field 76.
Field number	75
Byte offset start	2048
Byte offset end	2079
Byte limit	32
Data type	ASCII string
Mandatory/optional	Optional for FADGI use but Required if implementing Digitization process history in field 76
Values	FADGI digitization process history [31 bytes]

Digitization process history	
SMPTE ST 268 Definition	User defined data header: User defined - Postage stamp, processing logs, etc. (length is variable with maximum length of 1 Mbyte)
FADGI Use	This FADGI-defined element is designed to summarize data on the digitizing process including signal chain specifics and other elements. Based on the 'Coding History' element defined by EBU R98-1999: Format for the <CodingHistory> field in Broadcast Wave Format files for the constrained space within the Broadcast Wave 'bext' chunk, this field employs a defined string variable for each parameter of the digitization process. Data in this field defined as a collection of strings, each presented on a separate line, containing a history of the coding processes applied to the file. The first line documents the source film reel, the second line contains data on the capture process, the third line of data records information on the storage of the file. A new line is added when the coding history related to the file is changed.
Field number	76
Byte offset start	2080
Byte offset end	xx
Byte limit	Length is variable with maximum length of 1 Mbyte
Data type	ASCII string
Mandatory/optional	Optional for FADGI use but Required if field 75 contains "FADGI digitization process history"

Values	The first line documents the source film reel, the second line contains data on the capture process and the third line contains data on the storage of the file. A new line is added when the coding history related to the file is changed. Each variable within a string is separated by a comma-space and each line should end with a carriage return and line feed. Each variable is optional, to be used when needed. O=format (reversal, print, positive, negative, DPXv1, DPXv2, etc.) G=gauge (super8mm, 8mm,16mm, 35mm, etc.) C=color (color, BW) S=sound (silent, composite optical, composite mag, separate optical reel, separate mag reel, etc.) D=summary of condition issues, especially if condition impacts visual quality of digitized image F=frames per second A=aspect ratio L=timing, grading (one-light, scene) W=bit depth (12-bit, 10-bit, 8-bit, etc.) R=resolution (2K, 4K, 8K, etc.) M=color model (RGB Log, etc.) N=name of vendor or operator who scanned film (if applicable) T=free ASCII text string; contains no commas but semicolons may be used.
Example	O=positive, G=16mm, C=color, S=silent, F=24, A=4:3,D=warped O=DPXv1, L=one-light, W=10-bit, R=2K, M=RGB Log, T=FilmScannerA; SN123456; in-house O=DPXv1, W=10-bit, R=2K, M=RGB Log [Explanation: Line 1 reads: a 16mm positive color print, with no associated soundtrack, at 24fps and 4:3 aspect ratio (1.375:1). The film was warped and impacted the visual quality of the image. Line 2 reads: film was digitized to a DPX version 1 file. One-light grading was employed. The image is 10-bit at 2K resolution (2048x1556) with RGB Log color model. The film was digitized via FilmScannerA (anonymized name of film scanner), serial number 123456, which is an in-house film scanner. Line 3 reads: the file is stored as DPXv1, 10-bit 2K RGB log]

Example	O=positive, G=35mm, C=BW, S=optical, F=24, A=4:3, T=composite optical O=DPXv2, L=one-light, W=10-bit, R=4K, M=RGB Log, T= FilmScannerB; SN98765; SoftwareX; soundtrack in frame; offsite , N=ScanningVendor1 O=DPXv2, W=10-bit, R=4K, M=RGB Log [Explanation: Line 1 reads: a 35mm positive black and white print, with a composite optical soundtrack, at 24fps and 4:3 aspect ratio (1.375:1). The film had some shrinkage. Line 2 reads: film was digitized to a DPX version 2 file. One-light grading was employed. The image is 10-bit at 4K resolution (4096x3112) with RGB Log color model. The film was digitized via FilmScannerB (anonymized name of film scanner), serial number 98765 and processed through SoftwareX (anonymized name of workstation processing software). The optical soundtrack is captured within the image frame. The film was digitized offsite by ScanningVendor1 (anonymized name of digitization vendor). Line 3 reads: the file is stored as DPXv2, 10-bit 4K RGB log]

13 SPECIFICATIONS AND REFERENCES

European Broadcast Union. Technical Recommendation R98-1999. Format for the <CodingHistory> field in Broadcast Wave Format files, BWF. Online: http://tech.ebu.ch/docs/r/r098.pdf. Accessed: August 4, 2016.

FADGI. AS-07 MXF Archive and Preservation Format. Online: http://www.amwa.tv/downloads/specifications/AS-07_Proposed_Application_Specification.pdf. Accessed: August 4, 2016.

FADGI. Embedding Metadata in Digital Audio Files: Guideline for Federal Agency Use of Broadcast WAVE Files Version 2. Online: http://www.digitizationguidelines.gov/audio-visual/documents/Embed_Guideline_20120423.pdf. Accessed: August 4, 2016.

ISO. ISO 8601:2000(E): Data elements and interchange formats — Information interchange — Representation of dates and times.

Library of Congress. Sustainability of Digital Formats: Digital Moving-Picture Exchange (DPX), Version 2.0. Online: http://www.digitalpreservation.gov/formats/fdd/fdd000178.shtml. Accessed: August 4, 2016.

ST 254:2008 - SMPTE Standard - Motion-Picture Film (35-mm) — Manufacturer-Printed Latent-Image Identification Information

ST 268:1994, SMPTE Standard for File Format for Digital Moving-Picture Exchange (DPX), Version 1.0. No longer available for purchase from SMPTE.

ST 268:2003, SMPTE Standard for File Format for Digital Moving-Picture Exchange (DPX), Version 2.0

Amendment 1:2012 to ST 268:2003, File Format for Digital Moving Picture Exchange (DPX) - Amendment 1

Wikipedia. List of motion picture film stocks. Online: https://en.wikipedia.org/wiki/List_of_motion_picture_film_stocks. Accessed: November 29, 2016.

14 OTHER HELPFUL REFERENCES.

FADGI. Digitizing Motion Picture Film: Exploration of the Issues and Sample SOW. Online: http://www.digitizationguidelines.gov/guidelines/FilmScan_PWS-SOW_20160418.pdf. Accessed: August 4, 2016.

FADGI. Embedding Metadata in Digital Audio Files: Introductory Discussion for the Federal Agencies Guideline. Online: http://www.digitizationguidelines.gov/audio-visual/documents/Embed_Intro_20120423.pdf. Accessed: August 4, 2016.

FADGI. Identifiers: Types and Characteristics. Online: http://www.digitizationguidelines.gov/audio-visual/documents/IdentifiersTypesCharacteristics_20111121.pdf. Accessed: August 4, 2016.

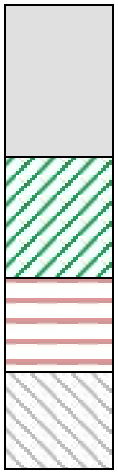
International Press Telecommunications Council Photo Metadata Working Group. Embedded Metadata Manifesto. Online: http://www.embeddedmetadata.org/embedded-metatdata-manifesto.php. Accessed: May 4, 2017.

Smith, Kelly L. GovSpeak: A Guide to Government Acronyms & Abbreviations. Online: http://libguides.ucsd.edu/govspeak. Accessed: August 4, 2016.

WC3. Date and Time Formats. Online: https://www.w3.org/TR/NOTE-datetime. Accessed: December 1, 2016.

15 APPENDIX A: SUMMARY OF SAMPLE FILE ANALYSIS

Header elements from SMPTE ST268:2003 | **Sample summary** | **SMPTE ST 268M definitions**



SMPTE Core Field

SMPTE Core Field Typically Implemented as Specified

SMPTE Core Field NOT Typically Implemented as Specified

Non SMPTE Core Field NOT Typically Implemented as Specified

File information header

Field	Offset	Length	Type	SMPTE Core?	Content	Sample summary	SMPTE ST 268M definitions
1	0	4	U32	**Core**	Magic number (SDPX ASCII)	**Typically implemented according to spec**	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.
2	4	4	U32	**Core**	Offset to image data in bytes	**Inconsistent. Sometimes used correctly, sometimes this data is recorded in field 19 or 21.12**	*no definition*
3	8	8	ASCII	**Core**	Version number of header format (V1.0 or V2.0)	**Consistently implemented according to spec**	*no definition*
4	16	4	U32	**Core**	Total image file size in bytes (including file header)	**Inconsistent. Sometimes used correctly, sometimes blank**	Indicates the size of the entire file, i.e. containing both header and image data.

5	20	4	U32		Ditto key (0 = same as previous frame; 1 = new)	Inconsistent. Sometimes used correctly, sometimes blank	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 runlength 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.
6	24	4	U32		Generic section header length in bytes	Inconsistent. Sometimes used correctly, sometimes not.	*no definition*
7	28	4	U32		Industry specific header length in bytes	Inconsistent. Sometimes used correctly, sometimes not.	*no definition*
8	32	4	U32		User-defined header length in bytes	Inconsistent. Sometimes used correctly, sometimes not.	*no definition*
9	36	100	ASCII		Image filename	Inconsistent. Sometimes used correctly, sometimes not.	*no definition*
10	136	24	ASCII		Creation date/time: yyyy:mm:dd:hh:mm:ssLTZ	Inconsistent. Sometimes used correctly, sometimes not. Often formatting is not correct	Defined as yyyy:mm:dd:hh:mm:ssLTZ, formatted according to ISO 8601.
12	160	100	ASCII		Creator	Inconsistent. Sometimes contains scanner name, sometimes holding institution, sometimes blank	*no definition*
13	260	200	ASCII		Project name	Inconsistent. Sometimes contains identifier, often blank	*no definition*
14	460	200	ASCII		Right to use or copyright statement	Inconsistent. Sometimes contains identifier, often blank	*no definition*
15	660	4	U32		Encryption key (FFFFFFFF unencrypted)	Inconsistent. Sometimes contains identifier, often blank	*no definition*
16	664	104	TBD		Reserved for future use		*no definition*

Image information header

Field	Offset	Length	Type	Core	Content		

17	768	2	U16	**Core**	Image orientation (see table 2)	**Consistently implemented according to spec**	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).
18	770	2	U16	**Core**	Number of image elements (1 - 8)	**Consistently implemented according to spec**	*no definition*
19	772	4	U32	**Core**	Pixels per line	**Consistently implemented according to spec**	Active number of pixels per line
20	776	4	U32	**Core**	Lines per image element	**Consistently implemented according to spec**	*no definition*
Data structure for image element 1							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.
21.1	780	4	U32	**Core**	Data sign (0 = unsigned; 1 = signed) (core set images are unsigned)	**Consistently implemented according to spec**	*no definition*
21.2	784	4	U32		Reference low data code value	Consistently implemented according to spec - usually to default 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.

21.3	788	4	R32		Reference low quantity represented	Consistently implemented according to spec - usually to default value	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]
21.4	792	4	U32		Reference high data code value	Inconsistent. Sometimes used correctly, sometimes not.	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.
21.5	796	4	R32		Reference high quantity represented	Inconsistent. Sometimes used correctly, sometimes not.	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.
21.6	800	1	U8	**Core**	Descriptor (see table 1)	**Consistently implemented according to spec**	Defines the components that make up an image element and their pixel packing order. The valid components are listed in table 1.
21.7	801	1	U8	**Core**	Transfer characteristic (see table 5A)	**Consistently implemented according to spec**	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).
21.8	802	1	U8	**Core**	Colorimetric specification (see table 5B)	**Consistently implemented according to spec**	Defines the appropriate color reference primaries (for additive color systems like television) or color responses (for printing density) (see table 5A).
21.9	803	1	U8	**Core**	Bit depth (see table 3A)	**Consistently implemented according to spec**	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).
21.10	804	2	U16	**Core**	Packing (see table 3B)	**Consistently implemented according to spec**	For image element n, defines the data packing mode. The valid options are listed in table 38.

21.11	806	2	U16	**Core**	Encoding (see table 3C)	**Consistently implemented according to spec**	For image element n, defines whether or not the element is run-length encoded. The valid options are listed in table 3C.
21.12	808	4	U32	**Core**	Offset to data	**Inconsistent. Sometimes used correctly, sometimes not.**	To data for image element n, defines the offset in bytes to the image data for element n from the beginning of the file.
21.13	812	4	U32		End-of-line padding	Consistently implemented according to spec	end-of-line padding: Specifies the number of padded bytes at the end of each line. The default is 0 (no padding).
21.14	816	4	U32		End-of-image padding	Consistently implemented according to spec	Specifies the number of padded bytes at the end of each image element. The default is 0 (no padding).
21.15	820	32	ASCII		Description of image element	Inconsistent. Sometimes used correctly, sometimes not.	*no definition*
22	852	72			Structure Data structure for image element 2	Typically no data	See 21
23	924	72			Structure Data structure for image element 3	Typically no data	See 21
24	996	72			Structure Data structure for image element 4	Typically no data	See 21
25	1068	72			Structure Data structure for image element 5	Typically no data	See 21
26	1140	72			Structure Data structure for image element 6	Typically no data	See 21
27	1212	72			Structure Data structure for image element 7	Typically no data	See 21
28	1284	72			Structure Data structure for image element 8	Typically no data	See 21
29	1356	52			TBD Reserved for future use	Typically no data	

Image source information header

Field	Offset	Length	Type	Core	Content	Sample summary	SMPTE ST 268M definitions
30	1408	4	U32		X offset	Typically no data	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.
31	1412	4	U32		Y offset	Typically no data	Defines the frame offset (in lines) from the first line in the original contiguous image. The default is 0.
32	1416	4	R32		X center	Typically no data	Defines the X image center in pixel units (floating point).
33	1420	4	R32		Y center	Typically no data	Defines the Y image center in line units (floating point).
34	1424	4	U32		X original size	Typically no data	Defines the number of pixels per line in the original image.
35	1428	4	U32		Y original size	Typically no data	Defines the number of lines per image in the original image.
36	1432	100	ASCII		Source image filename	Inconsistent. Sometimes used correctly, sometimes not.	Defines the source image from which this image was extracted or processed.
37	1532	24	ASCII		Source image date/time: yyyy:mm:dd:hh:mm:ssLTZ	Inconsistent. Sometimes used correctly, sometimes not.	Defines the creation time of the source image from which the image was extracted or processed. Formatting is as per clause 3.4
38	1556	32	ASCII		Input device name	Inconsistent. Sometimes used correctly, sometimes not.	*no definition*
39	1588	32	ASCII		Input device serial number	Inconsistent. Sometimes used correctly, sometimes not.	*no definition*
40	1620	8	U16*4		Border validity: XL, XR, YT, YB border	Typically no data	Defines the region of an image that is eroded due to edge-sensitive filtering operations. The X-left, X-right. V-top, and V-bottom value defines the width of the eroded border. The default is 0,0,0,0 in pixel units (no erosion).

41	1628	8	U32*2		Pixel aspect ratio (horizontal:vertical)	Typically no data	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.

Data structure for additional source image information						Sample summary	SMPTE ST 268M definitions
42.1	1636	4	R32		X scanned size	Typically no data	Defines the horizontal size of the original scanned optical image in millimeters.
42.2	1640	4	R32		Y scanned size	Typically no data	Defines the vertical size of the original scanned optical image in millimeters.
42.3	1644	20	TBD		Reserved for future use	Typically no data	

Industry-specific headers

Motion-picture film information header

Field	Offset	Length	Type	Core	Content	Sample summary	SMPTE ST 268M definitions
43	1664	2	ASCII		Film mfg. 10 code (2 digits from film edge code)	Typically no data	Encodes data from machine readable portion of film edge code, according to SMPTE 254.
44	1666	2	ASCII		Film type (2 digits from film edge code)	Typically no data	Encodes data from machine readable portion of film edge code, according to SMPTE 254.
45	1668	2	ASCII		Offset in perfs (2 digits from film edge code)	Typically no data	Encodes data from machine readable portion of film edge code, according to SMPTE 254.
47	1670	6	ASCII		Prefix (6 digits from film edge code)	Typically no data	Encodes data from machine readable portion of film edge code, according to SMPTE 254.

48	1676	4	ASCII		Count (4 digits from film edge code)	Typically no data	Encodes data from machine readable portion of film edge code, according to SMPTE 254.
49	1680	32	ASCII		Format - e.g. Academy	Inconsistent. Sometimes used as summary for source format	Encodes data from machine readable portion of film edge code, according to SMPTE 254.
50	1712	4	U32		Frame position in sequence	Inconsistent. Sometimes used correctly, sometimes no data	Defines the frame number in the image sequence.
51	1716	4	U32		Sequence length (frames)	Inconsistent. Sometimes used correctly, sometimes no data	Defines the total number of frames in the image sequence.
52	1720	4	U32		Held count (1 = default)	Inconsistent. Sometimes used correctly, sometimes no data	Specifies how many sequential frames for which to hold the current frame. In animation, it is often desirable to hold identical frames.
53	1724	4	R32		Frame rate of original *(frames/s)*	Inconsistent. Sometimes used correctly, sometimes no data	*no definition*
54	1728	4	R32		Shutter angle of camera in degrees	Inconsistent. Sometimes used correctly, sometimes no data	Defines the shutter angle in degrees of the motion-picture camera. This specifies the temporal sampling aperture.
55	1732	32	ASCII		Frame identification - e.g. keyframe	Inconsistent. Sometimes used correctly, sometimes no data	A user-defined field that labels select frames as key frames or wedge frames, etc.
56	1764	100	ASCII		Slate information	Inconsistent. Sometimes used correctly, sometimes used as summary of source format	A user-defined ASCII field for recording production information from the camera slates.
57	1864	56	TBD		Reserved for future use	Typically no data	*no definition*

Television information header

Field	Offset	Length	Type	Core	Content		
58	1920	4	U32		SMPTE time code (Table 6)		The characters are encoded into the 32-bit word according to table 6.
59	1924	4	U32		SMPTE user bits (Table 6)		These are encoded according to table 6.

60	1928	1	U8		Interlace (0 = noninterlaced; 1 = 2:1 interlace)	Typically no data	*no definition*
61	1929	1	U8		Field number	Typically no data	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.
62	1930	1	U8		Video signal standard (see table 4)	Typically no data	Defines the video source. Video signal standards are listed in table 4.
63	1931	1	U8		Zero (for byte alignment)	Typically no data	*no definition*
64	1932	4	R32		Horizontal sampling rate (Hz)	Typically no data	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.
65	1936	4	R32		Vertical sampling rate (Hz)	Typically no data	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.
66	1940	4	R32		Temporal sampling rate or frame rate (Hz)	Inconsistent. Sometimes replicates data in field 53, sometimes blank	*no definition*
67	1944	4	R32		Time offset from sync to first pixel (ms)	Inconsistent. Sometimes 00 00 00 00, sometimes blank	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.

68	1948	4	R32		Gamma	Inconsistent. Sometimes 00 00 00 00, sometimes blank	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.
69	1952	4	R32		Black level code value	Inconsistent. Sometimes 00 00 00 00, sometimes blank	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.
70	1956	4	R32		Black gain	Inconsistent. Sometimes 00 00 00 00, sometimes blank	Defines the linear gain applied to signals below the breakpoint (this is 4.5 for SMPTE 274M).
71	1960	4	R32		Breakpoint	Inconsistent. Sometimes 00 00 00 00, sometimes blank	Defines the signal level above which the gamma law is applied (this is 0.018 of full scale for SMPTE 274M).
72	1964	4	R32		Reference white level code value	Inconsistent. Sometimes 00 00 00 00, sometimes blank	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.
73	1968	4	R32		Integration time (s)	Inconsistent. Sometimes 00 00 00 00, sometimes blank	Defines the temporal sampling aperture of the television camera; most useful for CCD cameras.
74	1972	76	TBO		Reserved for future use	Inconsistent. Sometimes blank, sometimes other data	

User defined data

Field	Offset	Length	Type	Core	Content	Sample summary	SMPTE ST 268M definitions
75	2048	32	ASCII		User identification	Inconsistent. Sometimes blank, sometimes other data	*no definition*
76	2080	xx	TBD		User defined - Postage stamp, processing logs, etc. (length is variable with maximum length of 1 Mbyte)		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

	2304	42+	ASCII				

Image data

Field	Offset	Length	Type	Core	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).	Consistently implemented according to spec	*no definition*

16 APPENDIX B: SUMMARY OF METADATA FIELDS BY OBLIGATION

Field #	Content	**Required/Recommended/Optional**
1	Magic number (SDPX ASCII)	Core Field, required by SMPTE ST268
2	Offset to image data in bytes	Core Field, required by SMPTE ST268
3	Version number of header format (V1.0 or V2.0)	Core Field, required by SMPTE ST268
4	Total image file size in bytes (including file header)	Core Field, required by SMPTE ST268
9	Image filename	Strongly recommended for FADGI
10	Creation date/time: yyyy:mm:dd:hh:mm:ssL TZ	Strongly recommended for FADGI
12	Creator	Strongly recommended for FADGI
13	Project name	Strongly recommended for FADGI
14	Right to use or copyright statement	Recommended for FADGI
17	Image orientation (see table 2)	Core Field, required by SMPTE ST268
18	Number of image elements (1 - 8)	Core Field, required by SMPTE ST268
19	Pixels per line	Core Field, required by SMPTE ST268
20	Lines per image element	Core Field, required by SMPTE ST268
21.1	Data sign (0 = unsigned; 1 = signed)	Core Field, required by SMPTE ST268
21.6	Descriptor (see table 1)	Core Field, required by SMPTE ST268
21.7	Transfer characteristic (see table 5A)	Core Field, required by SMPTE ST268
21.8	Colorimetric specification (see table 5B)	Core Field, required by SMPTE ST268
21.9	Bit depth (see table 3A)	Core Field, required by SMPTE ST268
21.10	Packing (see table 3B)	Core Field, required by SMPTE ST268
21.11	Encoding (see table 3C)	Core Field, required by SMPTE ST268
21.12	Offset to data	Core Field, required by SMPTE ST268
37	Source image date/time: yyyy:mm:dd:hh:mm:ssL TZ	Recommended for FADGI
38	Input device name	Recommended for FADGI
39	Input device serial number	Recommended for FADGI
43	Film mfg. 10 code (2 digits from film edge code)	Optional for FADGI
44	Film type (2 digits from film edge code)	Optional for FADGI
50	Frame position in sequence	Recommended for FADGI

51	Sequence length (frames)	Recommended for FADGI
76	User defined - Postage stamp, processing logs, etc. (length is variable with maximum length of 1 Mbyte)	Optional for FADGI