SMPTE ST 2067-5:2020
Revision of SMPTE ST 2067-5:2013

SMPTE STANDARD

Interoperable Master Format — Essence Component

Table of Contents

Approved April 7, 2020

Foreword

SMPTE (the Society of Motion Picture and Television Engineers) is an internationally-recognized standards developing organization. Headquartered and incorporated in the United States of America, SMPTE has members in over 80 countries on six continents. SMPTE's Engineering Documents, including Standards, Recommended Practices, and Engineering Guidelines, are prepared by SMPTE's Technology Committees. Participation in these Committees is open to all with a bona fide interest in their work. SMPTE cooperates closely with other standards-developing organizations, including ISO, IEC and ITU.

SMPTE Engineering Documents are drafted in accordance with the rules given in its Standards Operations Manual. This SMPTE Engineering Document was prepared by Technology Committee 35PM.

Intellectual Property

At the time of publication no notice had been received by SMPTE claiming patent rights essential to the implementation of this Engineering Document. However, attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. SMPTE shall not be held responsible for identifying any or all such patent rights.

Introduction

This section is entirely informative and does not form an integral part of this Engineering Document.

The Interoperable Master Format (IMF) is a file-based framework designed to represent one or more high-quality versions of the same finished work destined for distribution channels worldwide. IMF architecture separates composition metadata and essence into separate file objects to promote re-use and efficient operation.

In IMF, essence data (and associated, essence-specific metadata) is stored within MXF Generic Containers contained in MXF files. This document provides a common set of characteristics (a common *formulation*) for these MXF files, termed Essence Components. An essence component is a mono essence Track File in an OP1a MXF Generic Container for audio, picture and data.

1 Scope

This standard defines the IMF Essence Component, an OP1a MXF structure for interchange of audio, picture and data essence. It defines data structures for interchange at the signal interfaces of networks or storage media, but does not define internal storage formats for compliant devices or mappings for particular essence encodings. It also defines metadata expressions used to communicate allowable variations in the formulation of an IMF Essence Component.

2 Conformance Notation

Normative text is text that describes elements of the design that are indispensable or contains the conformance language keywords: "shall", "should", or "may". Informative text is text that is potentially helpful to the user, but not indispensable, and can be removed, changed, or added editorially without affecting interoperability. Informative text does not contain any conformance keywords.

All text in this document is, by default, normative, except: the Introduction, any section explicitly labeled as "Informative" or individual paragraphs that start with "Note:"

The keywords "shall" and "shall not" indicate requirements strictly to be followed in order to conform to the document and from which no deviation is permitted.

The keywords, "should" and "should not" indicate that, among several possibilities, one is recommended as particularly suitable, without mentioning or excluding others; or that a certain course of action is preferred but not necessarily required; or that (in the negative form) a certain possibility or course of action is deprecated but not prohibited.

The keywords "may" and "need not" indicate courses of action permissible within the limits of the document.

The keyword “reserved” indicates a provision that is not defined at this time, shall not be used, and may be defined in the future. The keyword “forbidden” indicates “reserved” and in addition indicates that the provision will never be defined in the future.

A conformant implementation according to this document is one that includes all mandatory provisions ("shall") and, if implemented, all recommended provisions ("should") as described. A conformant implementation need not implement optional provisions ("may") and need not implement them as described.

Unless otherwise specified, the order of precedence of the types of normative information in this document shall be as follows: Normative prose shall be the authoritative definition; Tables shall be next; then formal languages; then figures; and then any other language forms.

3 Normative References

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

SMPTE ST 330:2011, Unique Material Identifier (UMID)

SMPTE ST 377-1:2011, Material Exchange Format (MXF) — File Format Specification

SMPTE ST 378:2004, Television – Material Exchange Format (MXF) — Operational Pattern 1a (Single Item, Single Package)

SMPTE ST 379-1:2009, Material Exchange Format (MXF) — MXF Generic Container

SMPTE ST 379-2:2010, Material Exchange Format (MXF) — MXF Constrained Generic Container

SMPTE ST 380:2004, Television – Material Exchange Format (MXF) — Descriptive Metadata Scheme-1

SMPTE ST 410:2008, Material Exchange Format — Generic Stream Partition

SMPTE ST 436-1:2013, Television — MXF Mappings for VBI Lines and Ancillary Data Packets

4 Overview

The design of IMF Essence Component recognizes that some flexibility is required in the formulation of the MXF to accommodate a range of use cases to which IMF architecture may be applicable. To limit the variety of formulation a decoder must support (and to simplify the description of these variations), this specification includes a *shim* concept that describes the variant aspects of IMF Essence Component using keyword parameters (with constrained values) that can be used to describe a specific formulation. This has the goal that engineers can develop IMF Applications, using IMF Essence Component as a core component, knowing that allowable variations can be defined by the Application. This allows IMF applications to satisfy their respective requirements (by taking advantage of IMF Essence Component's flexibility) while reusing the same core specification and retaining interoperability.

4.1 File Format Requirements (Informative)

IMF Essence Component addresses the problem of having a common essence file format in an architecture that is expected to accommodate many essence encodings.

In an ideal world, a single essence representation would be standardized on. Unfortunately, over time, requirements for a facility change. The facility that started as SD introduces HD. The compression format of today gets replaced with a better one from tomorrow. Stereo audio gets upgraded to multi-channel and multi-lingual. In parallel, a facility operates with a constant pressure to reduce the cost of production, distribution and publishing.

In order to build systems that work at the MXF level rather than, *e.g.*, at the MPEG or JPEG2000 level, IMF Essence Component provides a set of rules that MXF encoders and MXF decoders must follow. This brings many advantages. It allows a device to query the picture size without having to implement a decoder for each essence type — you simply read the MXF metadata. It allows a business process to be specified with an action such as:

If (16:9), then transcode (center-cut-out)

This action can be specified without the executor of process, human or system, needing to know the details of underlying codec type – that is something for the transcoder or reproducer to address. In general, this allows business systems to deliver work order requests to content-manipulation devices, which are in turn configured to support delivery specifications.

4.2 General IMF Essence Component and Shims

To maximize commonality across applications, this specification is divided into general provisions that apply to all applications, and specific constraint sets, called *shims*, that apply to defined applications.

The general provisions shall apply to all IMF Essence Components, and thus represent the maximum required capability of ingest servers, catch servers, playout servers, transcoders and other IMF compatible devices. Within these general provisions however are many potential variants - a shim may be used to identify a particular variant.

A shim is not intended to be a runtime element of a system that processes files as defined by this standard. Rather, it is provided as a defined means of expressing constrained variants of this standard by application designers.

Each shim shall provide a further set of constraints that reduce the range of variability that may be needed in well-defined categories of applications. For example, these categories could address particular type of programming or programming genres, or they could address requirements of particular broadcast station groups.

Documents that reference this standard may extend the shim to address domain-specific variants, but shims shall not be used to add new required capability to the general provisions. They shall be limitations on the general provisions. Thus, the general provisions can be non-restrictive in some areas and this is intentional.

In any given facility design or interchange specification based on IMF Essence Component, there may be one or more tightly defined shims. Typical examples are to have a shim for SD content, a shim for HD content and a shim for web content. Each of these shims tightly defines the codecs and layout for the particular business application.

In this specification, shim parameters will be highlighted as Table 1:

Table 1. Shim parameter definition layout

Shim parameter	*Definition*
example	Semantic definition of the example shim parameter

4.3 Timecode in IMF Essence Components

The timecode used in IMF Essence Components is an annotation of elapse time along a track. This can be created to be numerically equal to the timecode time addresses of other standards such as SMPTE ST 12-1. Timecode can be used to convert a Position along a Track measured in Edit Units into a count of hours, minutes, seconds and frames.

5 Essence Component File Parameters and Constraints

5.1 Essence Track Parameters and Constraints

5.1.1 General

IMF Essence Components shall each contain a single MXF essence Track of a single essence type. They shall be subject to the general constraints listed below.

1. Essence in an IMF Essence Component shall be wrapped using the MXF Generic Container (GC), either SMPTE ST 379-1 or SMPTE ST 379-2.
2. An IMF Essence Component shall be a mono-essence MXF file.
3. An IMF Essence Component shall be frame wrapped for picture essence, per the designated GC.
4. An IMF Essence Component shall be clip wrapped for audio essence, per the designated GC.
5. An IMF Essence Component shall be clip wrapped or frame wrapped for data essence, per the designated GC.
6. An IMF Essence Component shall be regularly partitioned when frame wrapped.
7. Each body partition in an IMF Essence Component shall exclusively have metadata, index tables, or essence.
8. An IMF Essence Component shall be signaled as closed and complete in the header or footer partition.

9. An IMF Essence Component shall have distributed and complete index tables.
10. An IMF Essence Component shall be signaled as OP1a, per SMPTE ST 378.
11. An IMF Essence Component should not have a system item.
12. An IMF Essence Component shall have a KAG size of 1.
13. An IMF Essence Component shall have its primary package identifier set to the identity of the top-level file package within the essence component file. The UMID, per SMPTE ST 330, of the top-level file package in the essence component file shall be the unique identifier of the essence component.
14. On creation, a minimum 8k-Byte fill shall follow the header partition to allow for in-place extension.
15. All data sets shall follow MXF generation ID rules.
16. An IMF Essence Component may contain DMS-1 (SMPTE ST 380) metadata schemes.
17. An IMF Essence Component may contain private metadata schemes.

These parameters are described in the sections below. Parameters may be further constrained by shims as described in external documents.

5.1.2 Generic Container

IMF Essence Component files shall use the MXF Generic Container to wrap essence. Either the older unconstrained “MXF Generic Container” (SMPTE ST 379-1) or the newer “MXF Constrained Generic Container” (SMPTE ST 379-2) may be used. The selected GC specification shall exclusively define the use of the GC for a given application; the GC specification not selected shall be ignored in that same application.

Note: SMPTE ST 379-2 is recommended in the case where there is no need for compatibility with SMPTE ST 379-1.

Track files shall be associated with the shim parameter *gc_type* value specified in Table 2.

Table 2. Shim parameter *gc_type* value definition

Shim parameter	***Definition***
gc_type	Default value: **none** Type: enumeration Values: “379-1-gc”, “379-2-constrained-gc”

5.1.3 Mono Essence

Only one essence track shall exist in the file package. Only one type of essence element shall be present (*i.e.* data, audio, picture or compound) in the essence container.

Audio files containing interleaved channels shall be considered as mono-essence regardless of the number of channels and shall be described by a single track in the file package.

The mono-essence flag in the OP1a header metadata field shall be set.

5.1.4 Interleaving

IMF Essence Components shall not contain interleaved Generic Containers.

5.1.5 Partitions

A partition in an IMF Essence Component shall only have one of the following types of data in it:

- Header metadata;
- Essence;
- Index table;
- SMPTE ST 410 generic stream.

Track files shall be associated with the shim parameter *partition_spacing* value specified in Table 3.

Table 3. Shim parameter *partition_spacing* value definition

Shim parameter	***Definition***
partition_spacing	Default value: **60 seconds** Type: floating point Units: Seconds Values: strongly-constrained for frame wrapped files Body partitions in a frame wrapped file shall occur at regular temporal spacing indicated by this parameter. Regular shall mean that any variation in the regularity shall be less than the smaller of 1 second or 10% of the period *partition_spacing*.

Clip wrapped files shall have the following structure: a header partition; one body partition with an index table; one body partition with clip wrapped essence; a footer partition. This is shown in Figure 1.

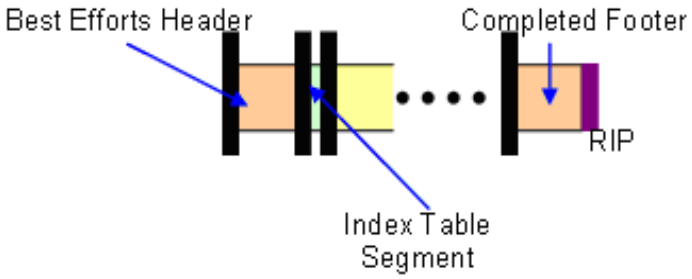

Figure 1. Clip wrapped file partition structure

All IMF Essence Components, when closed and complete, shall contain a Random Index Pack (RIP) as per SMPTE ST 377-1.

5.1.6 Index Tables

Frame wrapped, variable bit-rate essence shall have distributed index tables to aid process-during-write applications.

Track files shall be associated with the shim parameter values specified in Table 4.

Table 4. Index table shim parameter values definitions

Shim parameter	*Definition*
index_strategy_frame	Default value: "**lead"** Type: enumeration Values: "lead", "follow" or "file_specific" At each partition point in a given frame wrapped essence component file, the index partition shall either lead (*i.e.* precede) the essence partition that it indexes or follow the essence partition that it indexes. This shall be specified application by application. The "file_specific" option covers applications where the strategy cannot be defined.
Index_strategy_clip	Default value: "**lead"** Type: enumeration Values: "lead", "follow" or "file_specific" As above, but applies to clip wrapped files. When the index tables are calculated (PCM audio), lead means that the calculated index table precedes the header partition.

5.1.7 Picture

Picture essence in IMF Essence Components may be of one of a selection of encodings and compression families and acceptable variability should be defined by individual shims.

Track files for picture essence shall be associated with the shim parameter values specified in Table 5.

Table 5. Picture essence shim parameter values definitions

Shim parameter	*Definition*
picture_family	Default value: **none** Type: list of strings A list of supported picture essence compression families including any family-specific elements such as GOP structures and profiles, for example "MPEG-2 H.264 JPEG2000".
picture_bitrate	Default value: **none** Type: list of integer ranges A list of supported bitrates or ranges of bitrates for compressed pictures.
picture_format	Default value: **none** Type: list of formatted strings A list of permitted picture frame sizes and rates, for example "1920/1080/50p/16:9".
picture_custom_ANC	Default value: **false** Type: boolean When true, VBI data shall be encapsulated in the video essence using a defined method (e.g. carriage in MPEG picture user data) as well as being placed in a separate VBI atom.
picture_render_ANC	Default value: **false** Type: boolean When true, VBI data shall be encoded as active video within the video image. This data may also be placed in a separate VBI atom.

Picture files shall be frame wrapped with complete index tables.

5.1.8 Audio

PCM audio files shall be clip-wrapped.

Multi-channel encoded or compressed audio carried as data in an audio file (*i.e.*, as a Generic Container Data Track) shall be clip-wrapped.

Track files for audio essence shall be associated with the shim parameter values specified in Table 6.

Table 6. Audio essence file shim parameter values definitions

Shim parameter	*Definition*
audio_family	Default value: **none** Type: list of strings A list of supported audio essence families including bitrates, permitted soundfield arrangements, compression types and any format specific elements such as delay setting.
audio_file_arrangement	Default value: **none** Type: free text A description of the physical storage strategy for audio. Typically, this will be of the form “mono only files”, “stereo only files”, “stereo files with optional Dolby E in a file” etc.

Constant bitrate audio files shall have a calculated index table.

Variable bitrate audio files shall have a variable bitrate (VBR) index table.

5.1.9 Data

Data wrapping shall be defined by the application (i.e., this standard allows virtually any data wrapping supported by the Generic Container).

Data track files shall be associated with the shim parameter values specified in Table 7.

Table 7. Data essence shim parameter values definitions

Shim parameter	*Definition*
data_family	Default value: **none** Type: list of strings A list of supported data essence families including text formats, wrapping methods any format specific elements.
data_file_arrangement	Default value: **none** Type: free text A description of the physical storage strategy for data. Typically, this will be of the form “SMPTE ST 2052-1” clip-wrapped, or “CEA 708 in SMPTE ST 436-1” etc.

5.2 Header Metadata and Operational Pattern Constraints

5.2.1 Baseline operational pattern

The operational pattern of an IMF Essence Component shall be signaled as OP1a per SMPTE ST 377-1.

The qualifier bits of the operational pattern in IMF Essence Components shall be set as follows:

- bit 1: internal essence set to “0” to signal internal essence only.
- bit 2: streamable set to “0” to signal that the essence in the file is streamable.

- bit 3: uni-track set to “0” to signal that the essence in the file is atomic, i.e. there is only one track.

Note: This provision is intended to maximize compatibility with legacy MXF decoder applications.

5.2.2 Essence container label

IMF Essence Components shall each specify one essence container label that shall correspond to the essence container used in that file.

5.2.3 System item

The Generic Container system item may be present but should not be used by IMF Essence Components. Metadata in the system item that is to be used in IMF operation should be copied or moved to other containers such as the header metadata or SMPTE ST 436-1 data tracks.

The sys_item_tc shim parameter value shall be as specified in Table 8, if present.

Table 8. Shim parameter *sys_item_tc* value definition

Shim parameter	*Definition*
sys_item_tc	Default value: **false** Type: boolean To capture timecode in any generic container system, it shall be copied to the file package header metadata, including any discontinuities therein. This provision is intended to capture source timecodes and does not affect the material package of any version file.

5.2.4 Timecode

There may be several timecode tracks in the header metadata of an essence component. Timecode mode - drop-frame or non-drop frame - may be specified in each shim.

5.2.5 KAG size

IMF Essence Components shall have a KAG size of 1 unless this requirement conflicts with an underlying essence container specification. When a conflict exists, the value in that essence container specification shall be used.

5.3 Header Metadata Parameters and Constraints

The header partition of an essence component file shall start at the first byte of the file, with no run-in. Header metadata shall exist in the header partition and may have optional fill. All files shall comply with SMPTE ST 377-1.

The footer of an essence component file may contain a copy of the header metadata and, when present, it shall be marked closed and complete. All files shall have a footer partition unless:

1. the file is in the process of being written;
2. the underlying essence container specification forbids the existence of a footer.

The specific constraints on the header metadata shall be as specified in Table 9.

Table 9. Header metadata constraints

Set name	*Constraint*
Preface	As per SMPTE ST 377-1. The primary package shall be the top-level File Package.
Identification	Each device altering the file shall add 1 identification set.
Content Storage	1 material package, 1 top-level file package, no restriction on lower level packages. Lower level source packages shall be preserved by applications that edit or modify component files.
Essence Container Data	One single entry corresponding to the single Generic Container. BodySID and IndexSID shall be present and shall be different.
Material Package	Compliant with OP1a mono essence – shall contain only a single essence track. ***Shim parameter*** / ***Definition*** *track_tag_policy*: Default value: **none** Type: free text Specifies how material package tracks in the essence component file are tagged with metadata. This specification may be a private xml specification or a SMPTE specification.
Source Package (File / Physical)	There shall be a single essence descriptor for the essence.
Track (Timeline)	• The EditRate property of a track for material that is frame wrapped shall be equal to the rate of content packages in the essence. For progressive material, this is the underlying frame rate. • Clip wrapped audio shall have the edit rate set to the audio sampling rate except as otherwise defined by a specific application.
Event Track	Only used for descriptive metadata tracks. Note that metadata on an event track shall have its event duration equal to the validity of the metadata on the timeline. This may often be the duration of the track.
Static Track	Metadata may be present on a static track. This is discouraged. To prevent metadata morphing, static metadata should be placed on an event track with the duration set to the duration of the file. Applications manipulating files should be aware that metadata described as static may no longer be static after splicing and editing operations.
Sequence (all cases)	Unknown duration may be flagged as –1 in the header during writing. Known duration values shall be accurately signaled. Duration values shall be updated in footer (and optionally header) partitions when a file writing operation finishes.
SourceClip (Picture, Audio, Data)	Duration values shall use the same rule as for sequences.
MPEG Descriptor	When present for MPEG video component files, shall include picture essence coding, display width and height (frame height), aspect ratio, constant Bframes, MaxGOP, BpictureCount, Bitrate, ProfileAndLevel, VerticalSubsampling.
Wave Audio Essence Descriptor	Shall be present for all uncompressed PCM audio component files.
VBI Data Descriptor	Shall be present for all VBI components (See ST 436-1).
ANC Data Descriptor	Shall be present for all ANC components (See ST 436-1).
Multiple Descriptor	Shall be present only for component files that have VBI and ANC data present, otherwise it shall not be used.
Network Locator	Should not be present in essence component files.
Text Locator	Should not be present in essence component files.

Set name	*Constraint*	
Timecode		
	Shim parameter	***Definition***
	ingest_TC	Default value: **none** Type: extensible enumeration Values: “LTC”, “VITC”, “DVITC”, “external”, “source” or <value> Indicates that ingest (file creation) applications shall start the material package timecode and the top-level file package timecode at a given value, or derive it from the specified source.
	lead_TC	Default value: “**FP”** Type: enumeration Values: “MP”, “FP” Indicates that an application requiring the timecode of the component file shall use the lowest numbered file package timecode track (default) or the material package timecode track of the essence component file.
	dropframe_tc	Default value: **none** Type: Boolean Indicates presence of drop frame or non-drop frame timecode

The header metadata shall always be valid in a component file. This means that the process shown in Figure 2 shall be followed:

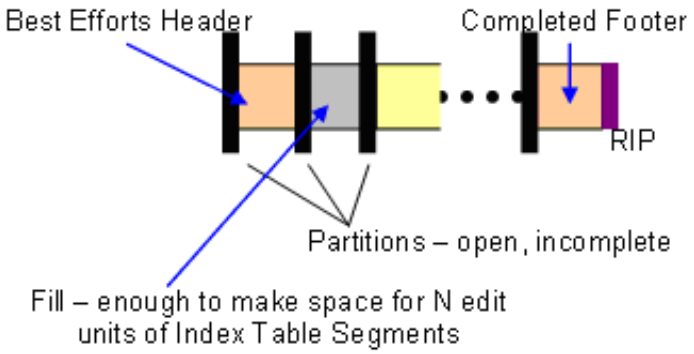

Figure 2. Valid header metadata process

Open and incomplete partitions are created during file writing operation. When the writing operation is finished, the header or footer shall be marked closed and complete. The footer may contain a copy of the header metadata and it shall be marked closed and complete.

5.3.1 Package labeling

All data sets shall include a generation ID that shall reference an identification set that shall be within the same file that describes the generation of the file when the data set was last modified.

All applications that modify an MXF file shall create a new identification set in the file and shall preserve existing identification sets. Full details are given in SMPTE ST 377-1.

Note: This feature can be used to audit metadata modification in the essence component file. It can therefore allow the detection of differences between "master" metadata and "derived" metadata when an MXF process modifies a file. This information can be used to flag the fact that metadata reconciliation might be required.

5.4 Descriptive Metadata Parameters and Constraints

5.4.1 Other descriptive metadata schemes

When present, private, application-defined metadata schemes, and extensions to SMPTE ST 380 DMS-1-based schemes, shall be identified in the preface.

6 Generic Shim

6.1 Shim Identifier

To aid IMF Application developers, the empty IMF Essence Component generic shim is presented in this section. In the following tables the terms “Strong”, “Moderate”, “Gentle”, and “Weak” are intended to convey the importance of defining the respective value in a shim that describes a particular variant. Application designers are advised to define those shim parameters marked Strong or Moderate. Application designers are encouraged to define shim parameters marked Gentle or Weak.

A shim identifier (*shim_id*) shall identify the shim. This identifier is intended to signal the version of the shim that was in use when the file was created or modified.

The *shim_id* shim parameter value shall be as specified in Table 10.

Table 10. Shim parameter *shim_id* value definition

Shim parameter	*Definition*
shim_id	Default value: **none** Type: depends on identifier type, *i.e.*, URI It is recommended that the organization creating the shim uses an XML namespace URI as a shim_id. This minimizes the chances of a shim identifier clash with other organizations. Publishing the shim document at that URL may also encourage other organizations to copy the shim and hence minimize unnecessary invention in the industry. *e.g.* `http://example.smpte-ra.org/names/20XX/YYYY/IMF-MPEG-SSTP`

The sections below are organized as a sheet to be filled in by a system designer who is constraining their application. Not all the parameters are relevant in every design. Some parameters can be left as none or not applicable (*n/a*).

The entry in the parameter name column is intended for use as the name of an XML element in a machine-readable template for the shim.

6.2 Shim

Table 11. Entry for shim ID

Dimension	*Description*	*Parameter name*	*IMF constraint*	*default values*	*Shim-specific value*
Shim ID	The identifier of this shim.	*shim_id*	Strong	none	

6.3 General Essence

Table 12. Entries for general essence

Dimension	*Description*	*Parameter name*	*IMF constraint*	*default values*	*Shim-specific constraint*	*Shim-specific values*
Generic Container (GC) Type	Essence wrapping framework.	*gc_type*	Strong	none		
Partition spacing	Regular partition spacing.	*partition_spacing*	Strong	60s		
Indexing strategy	Frame wrapping strategy.	*index_strategy_frame*	Strong	lead		
Indexing strategy	Clip wrapping strategy.	*Index_strategy_clip*	Strong	lead		

6.4 Picture Components

Table 13. Entries for picture components

Dimension	*Description*	*Parameter name*	*IMF constraint*	*default values*	*Shim-specific constraint*	*Shim-specific values*
Picture essence schemes	What picture signal schemes (compression, sampling or other) are present in the Essence Component?	*picture_family*	Gentle	none		
Picture bitrate	How many bits/second at real time?	*picture_bitrate*	Gentle	none		
Picture format	Picture raster and aspect ratio?	*picture_format*	Gentle	none		
Picture custom ANC	Custom methods for VBI / VANC?	*picture_custom_ANC*	Moderate	false		
Picture render ANC	VBI / VANC in-picture?	*picture_render_ANC*	Moderate	false		

6.5 Audio Components

Table 14. Entries for audio components

Dimension	*Description*	*Parameter name*	*IMF constraint*	*default values*	*Shim-specific constraint*	*Shim-specific values*
Audio essence schemes	What audio signal schemes (compression, sampling or other) are present in the Essence Component?	***audio_family***	None	none		
Audio file arrangement	Physical storage strategy for audio.	***audio_file_arrangement***	None	none		

6.6 Data Components

Table 15. Entries for data essence

Dimension	*Description*	*Parameter name*	*IMF constraint*	*default values*	*Shim-specific constraint*	*Shim-specific values*
Data essence schemes	What data (timed-text) schemes are present in the Essence Component?	***data_family***	None	none		
Data file arrangement	Physical storage strategy for data.	***data_file_arrangement***	None	none		

6.7 Header Metadata

Table 16. Entries for header metadata

Dimension	*Description*	*Parameter name*	*IMF constraint*	*default values*	*Shim-specific constraint*	*Shim-specific values*
System item timecode handling	Is the system item timecode copied to the header?	***sys_item_tc***	Moderate	false		
Track tag metadata	How are tracks tagged?	***track_tag_policy***	None	None		
Ingest timecode handling	What timecode source shall be dominant when creating files?	***ingest_TC***	Strong	none		
Timecode precedence	What timecode source shall be dominant when using essence files?	***lead_TC***	Strong	FP		
Timecode Drop-Frame Mode	Is drop-frame timecode present?	***dropframe_tc***	None	none		