SMPTE STANDARD

Material Exchange Format (MXF) — Mapping MPEG Streams into the MXF Constrained Generic Container

Table of Contents

Approved
August 23, 2011

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 Part XIII of its Administrative Practices.

SMPTE ST 381-2 was prepared by Technology Committee 31FS.

Intellectual Property

At the time of publication no notice had been received by SMPTE claiming patent rights essential to the implementation of this Standard. 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.

SMPTE ST 381 is a multi-part standard.

SMPTE ST 381-1 is the renumbered archived version of SMPTE 381M-2005. SMPTE ST 381-2, this part, has been created to update SMPTE 381M-2005 by providing clarification of text and updating existing provisions to promote interoperability. Users of SMPTE ST 381-1 and SMPTE ST 381-2 need to be aware that SMPTE ST 381-1 refers only to the Generic Container within SMPTE ST 379-1, not SMPTE ST 379-2, while SMPTE ST 381-2 refers only to the Generic Container within SMPTE ST 379-2, not SMPTE ST 379-1.

SMPTE ST 381-2, this part, defines the synchronization requirements and mechanisms for mapping MPEG material within MXF including the definition of the MPEG Video and Audio Descriptors, Essence Element Keys, and an Essence Container Label for MPEG streams into MXF along with CBE and VBE indexing of MPEG content using Index Tables.

The SMPTE ST 381 multi-part specification on the MXF MPEG mapping was developed using a number of fundamental requirements. These requirements are the following:

- Preserve timing in any MPEG system stream.
- Preserve timing when de-multiplexing a system stream and converting to MPEG ES wrapping.
- Preserve the MPEG buffer model so that when the MPEG is removed, the buffer model is preserved when this is required.
- Encapsulate MPEG PES without knowledge of the contents.
- Provide a mechanism for key frame identification.
- No change of original MPEG data order, this will facilitate recording an MPEG stream in MXF and playing out an MPEG stream from MXF.
- Small CPU and memory needs for both recording and playing out.
- Covers all possible types of MPEG data: MPEG-2 TS, PS, MPEG-1 system, MPEG-1 and MPEG-2 PES, video ES, audio ES, private streams and other derived format such as DVD-VOB (DVD Video Object), VCD (Video CD), etc.
- Facilitate access to individual Access Units.
- Possibility to wrap the MPEG stream based on main level elements for storage critical environments.

1 Scope

This standard specifies the mapping of video and audio MPEG ES, as identified by an ISO/IEC 13818-1 stream_id value, as well as the carriage of data streams extracted from higher level MPEG multiplexes, into the MXF Generic Container described in SMPTE ST 379-2. This includes, but is not limited to, ISO/IEC 13818-2 MPEG video stream, ISO/IEC 11172-2 MPEG video streams, ISO/IEC 14496-2 MPEG visual streams, ISO/IEC 14496-10 video streams, ISO/IEC 13818-3 MPEG audio streams, ISO/IEC 11172-3 MPEG audio streams, and ISO/IEC 14496-3 MPEG audio streams. It does not include ISO/IEC 14496-15 video streams. The document outlines single stream mapping and synchronized multi-stream mapping along with synchronization requirements for the MXF encapsulation of co-timed MPEG Essence streams. This standard defines SMPTE universal labels to be used to uniquely identify specific MPEG implementations. The Essence Element Keys and an Essence Container Label for MPEG streams are detailed along with CBE and VBE indexing of MPEG content using Index Tables.

In order to achieve interoperability within any given Operational Pattern, restrictions may be placed on the way in which this Essence Container can be implemented. The reader is advised to carefully study the appropriate Operational Pattern document before implementation.

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, followed by formal languages, then figures, and then any other language forms.

3 Normative References

Note: All references in this document to other SMPTE documents use the current numbering style (e.g. SMPTE ST 336:2007) although, during a transitional phase, the document as published (printed or PDF) may bear an older designation (such as SMPTE 336M-2007). Documents with the same root number (e.g. 336) and publication year (e.g. 2009) are functionally identical.

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

SMPTE ST 336:2007, Data Encoding Protocol using Key-Length-Value

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

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

ISO/IEC 13818-1:2007, Information Technology — Generic Coding of Moving Pictures and Associated Audio Information — Part 1: Systems

ISO/IEC 13818-2:2000, Information Technology — Generic Coding of Moving Pictures and Associated Audio Information — Part 2: Video

ISO/IEC 13818-3:1998, Information Technology — Generic Coding of Moving Pictures and Associated Audio Information — Part 3: Audio

ISO/IEC 13818-7:2004, Information Technology — Generic Coding of Moving Pictures and Associated Audio Information — Part 7: Advanced Audio Coding (AAC)

ISO/IEC 14496-2:2004, Information Technology — Coding of Audio Visual Objects — Part 2:Visual

ISO/IEC 14496-10:2009, Information Technology — Coding of Audio Visual Objects – Part 10: Advanced Video Coding or ITU-T Recommendation H.264 — Advanced Video Coding for Generic Audio-Visual Services

4 Glossary of Acronyms, Terms and Data Types

For the purposes of this document, the terms and definitions given in the previous section containing the normative references within this document and the following apply.

AAC: Advanced Audio Coding

Audio AU: The full definition of an Access Unit (AU) for audio is given in ISO/IEC 13818-1, but is summarized as the following: 1 Audio AU = The coded representation of an audio frame.

AVC/H.264: Advanced Video Coding – ISO/IEC 14496-10 (Part 10) | ITU-T Rec. H.264

BWF: Broadcast Wave Format for audio (see SMPTE ST 382)

ES: Elementary Stream

GOP: Group of Pictures

GOV: Group of VOP

MPEG-1: ISO/IEC 11172

MPEG-2: ISO/IEC 13818

MPEG-4 Visual: ISO/IEC 14496-2

Video AU: The full definition of an Access Unit (AU) for video is given in ISO/IEC 13818-1, but is summarized as the following: 1 Video AU = The coded data for a picture and any stuffing that follows it up to, but not including, the start of the next AU. If a picture is *not* preceded by a group_start_code or a sequence_header_code, the AU begins with a picture_start_code. If a picture is preceded by a group_start_code or a sequence_header_code, the AU begins with the first byte of these start codes. If it is the last picture preceding a sequence_end_code in the bit stream all bytes between the last bytes of the coded picture and the sequence_end_code (including the sequence_end_code) belong to the AU.

VO: Visual Object

VOL: Video Object Layer

VOP: Video Object Plane

VOS: Visual Object Sequence

5 MPEG Wrapping

5.1 Basics of Wrapping: Frames and Access Units

The MPEG video specifications enable the compression of picture-based video. These pictures can be frames or fields. The MPEG specification provides signaling for the number of pixels in the picture, the field/frame rate and other parameters including the aspect ratio of the pixels. Many of these parameters will be copied into the MXF Essence Descriptor.

The text in this document refers to Access Units (AUs) rather than pictures. The full definition of an AU for video and audio is given in ISO/IEC 13818-1 and a summary of an Audio AU and a Video AU is given in the glossary of this document.

Note: The Video AU Duration might not be the same as Audio AU Duration in an MPEG stream.

5.2 Wrapping Options

5.2.1 Overview

The MXF Generic Container described in SMPTE ST 379-2 consists of one or more contiguous KLV wrapped Content Packages. When wrapping every Video AU of long GOP MPEG, the frames shall be wrapped in transmission order rather than display order. When interleaving uncompressed audio samples in the same Content Package, the audio samples shall not be reordered to match the video. If the video is at a frame rate of 29.97 fps and the audio is sampled at 48 kHz, then there will be a varying number of audio samples in the Content Packages as a result of the complex relationship of the clocks.

5.2.2 Empty and Variable Elements

When a wrapping algorithm does not require data from a particular element, the element may be empty within the Content Package.

Elements within a Content Package may vary in size from one Content Package to another. These variations can be quite large and in the limit, the data for an element may not be present at all. In the case where a particular element has no Essence data to wrap, the KLV coded element shall be empty within the Content Package (i.e. the length will be zero and there will be no value data)..

SMPTE ST 379-2 contains further details on requirements for elements within a Content Package.

5.2.3 MPEG in the Generic Container

Figure 1 shows an example of a Generic Container, defined in SMPTE ST 379-2, to contain an MPEG stream. In general, the Generic Container was designed to create a streamable format where each repetition of the interleaved elements represents a single field or frame. Figure 1 shows an example that has one field of picture in every Content Package which is associated with an optional system item, some audio information and some data. This follows the Generic Container policy that the audio and data are co-timed with the video information in the same Content Package.

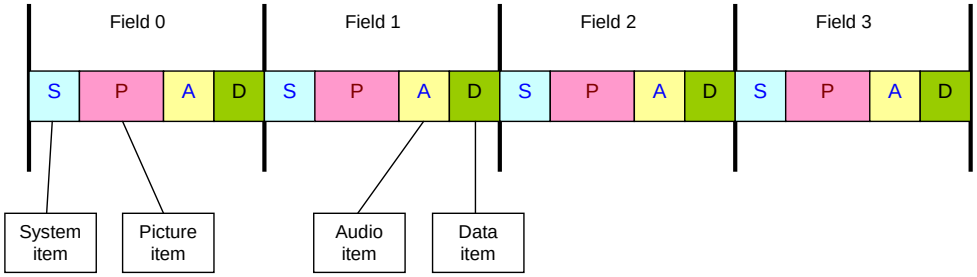

Figure 1 – Example of MPEG in a Generic Container

In some cases, different Content Packages will contain different Durations. When the compressed audio item is not synchronized with the picture item, the offset in time from the start of the audio Content Item to the Picture Content Item should be expressed using the fractional position offset mechanism defined in SMPTE ST 377-1 if an Index Table is constructed.

Note: Index Table requirements for MPEG mappings are discussed in Section 11 of this document.

As stated previously, SMPTE ST 379-2 contains further details on requirements for elements within a Content Package and the use of a Generic Container.

5.2.4 Interleaving Guidelines

In order to give MXF encoder guidance for creating consistent Generic Containers between applications, interleaving guidelines are defined within this document. The principle of creating the interleaving guidelines is to allow the creation of MXF files containing a mixture of unspecified components. For example:

Long GOP MPEG ES + BWF sound;
Uncompressed pictures + long GOP MPEG proxy + AES sound.

Different mapping mechanisms are listed in this and other mapping documents. Specific interleaving guidelines for implementing those mapping schemes are provided to improve interoperability. In addition to these low level interleaving guidelines, there may be higher level Partition multiplexing rules operating which may be determined by the Operational Pattern of the file.

6 MXF Tracks

6.1 MXF Tracks and MPEG Content

Many MXF files containing MPEG content will contain a picture Track for the MPEG video Essence and sound Tracks for any audio Essence. Many professional applications will use uncompressed audio which will have a simple timing relationship with the MPEG video.

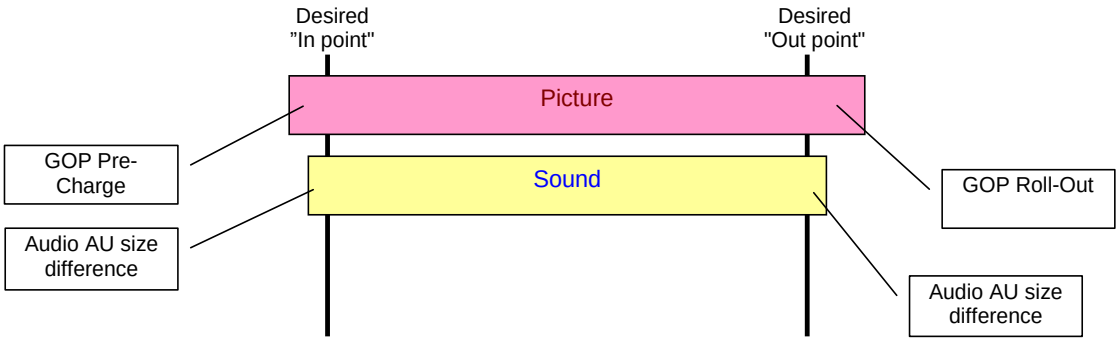

Figure 2 – Example of MXF Track Timing for MPEG Content

The example shown in Figure 2 shows the relationship between the desired start and end points of a stored generalized MPEG segment and the actual data start and end points which would be found in the stored Essence. The Track and sequence items within the header are used to accurately describe the synchronization scenario above.

As shown in Figure 2, the Pre-Charge used for inter-coded material represents the previous Edit Units that are needed before desired start Edit Unit of the Essence in order to correctly display the desired start picture. To display the last picture described by the Duration of the associated Essence Track, Roll-Out consisting of the count of Edit Units minus Origin minus the number of displayed pictures in the Essence Container may be needed for inter-coded material. Roll-Out is used to preserve the correct temporal order when displaying the last picture within the Essence. Taking Pre-Charge and Roll-Out into account, the value of the Container Duration Property within the File Descriptor, as described in SMPTE ST 377-1, equals the count of Edit Units starting at the first Content Package with a non-empty Essence Element to the last Content Package with a non-empty Essence Element.

In an OP1a file, the Material Package has the same Duration as the Top-Level File Package. The Duration Property of the Sequence Set for the Top-Level File Package consists of Pre-Charge and the Package Duration, but does not include Roll-Out. Using Pre-Charge and Roll-Out as described previously, unwanted frames, which were captured to allow decoding of the desired frames, are hidden.

Note: The audio can also be located in an external file. When the audio is external, there will be at least two File Packages.

If time code is present in the MPEG GOP headers or has been inserted in accordance with SMPTE ST 328 then, in the absence of specific application requirements, this time code should be used by the Material Package Timecode Track.

Note: Examples of an application specification is EBU R122.

7 Mapping MPEG into the Generic Container

7.1 Frame (Video AU) Wrapping

A sequence of images shall be KLV coded as defined in SMPTE ST 336. One type of KLV wrapping, defined as Frame Wrapping, is specified for the MPEG body structure and described in this section.

As described in SMPTE ST 379-2, Frame Wrapping has one or more Content Packages in the Essence Container. Examples of "Frame Wrapping" methods for MPEG are shown in Figures 3 and 4. Figure 3 shows an example of an MPEG video stream wrapped in a number of Content Packages with no other Generic Container elements in the container. Each Content Package has the Duration of one MPEG Video AU. Specifying the Duration through a MPEG Video AU determines where the MPEG headers will be found. Through these MPEG headers, the picture type can be determined as shown in Annex C of this document.

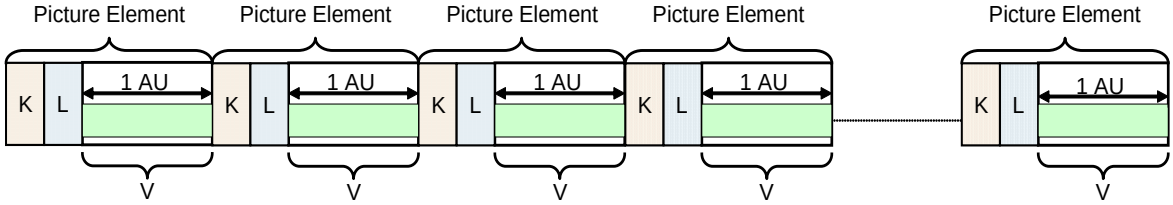

Figure 3 – Example of Frame Wrapping for MPEG Picture Elements

The Frame Wrapping method is intended to enable retrieval of individual AUs and the corresponding field/frame within them by MXF applications which process at the KLV level. This can be particularly useful for applications which support multiple Generic Container mapping types. Each MPEG Video AU is KLV wrapped using a Generic Container Picture Element Key. Using the Generic Container Picture Element Key, an individual MPEG Video AU within a KLV packet can be located as previously mentioned.

In some applications, the Frame Wrapped MPEG picture data can exist in Content Packages with other elements such as sound and data elements. An example Generic Container is shown in Figure 4.

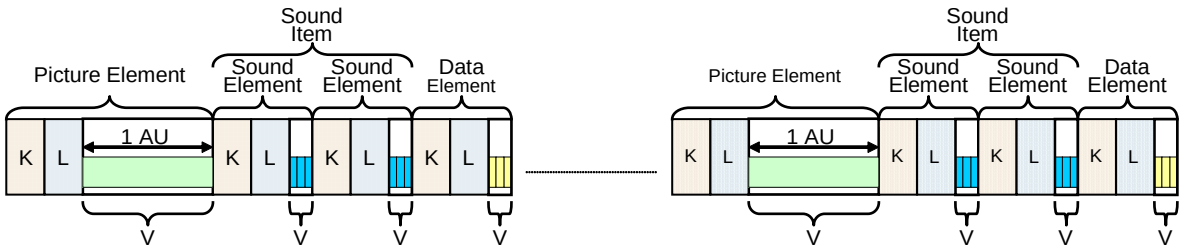

Figure 4 – Example of Frame Wrapping of MPEG Picture Elements with other Generic Container Elements

The Generic Container mapping specifications for the sound and data elements will detail the key values and format of the data within the elements. There are many situations where each wrapped frame of MPEG picture data will not exactly coincide with the sound/data elements within the same Content Package. These situations are described in more detail within SMPTE ST 379-2.

7.1.1 Interleaving Rules

When creating an MXF file from raw ES components, an MXF encoder has some flexibility. Easy access can be provided to all of the components by creating an MXF file for interchange. In this case, it is desirable to group together elements which are synchronized on the timeline. The overall goal is to keep the synchronized video, audio and associated data in the same Content Package. SMPTE ST 379-2 contains further details on requirements for elements within a Content Package and the use of a Generic Container.

7.1.2 Wrapping 3:2 Sequences

The MPEG coding of 3:2 sequences is defined in the MPEG specifications as explained in ISO/IEC 13818-2. KLV wrapping of the 3:2 AUs for interchange is, therefore, defined in the previous section. Ensuring that an MPEG Video AU is wrapped defines the KLV wrapping point regardless of the regularity of the 3:2 sequence.

Construction of Index Tables for 3:2 sequences requires the Index Table construction application to calculate the correct byte offset for a given temporal offset regardless of the MPEG encoding used. This may be done by the inspection of the MPEG headers for the occurrence of repeat_first_field flags or by some equivalent method.

7.2 Clip Wrapping

A sequence of images shall be KLV coded as defined in SMPTE ST 336. One type of KLV wrapping, defined as Clip Wrapping, is specified for the MPEG body structure and described in this section.

As described in SMPTE ST 379-2, Clip Wrapping is where the entire Duration of the Essence Container is contained with a single Content Package. An example of “Clip Wrapping” method for an MPEG Picture Element is shown in Figure 5. Figure 5 shows an MPEG video stream wrapped in a Content Package with no other Generic Container elements in the container.

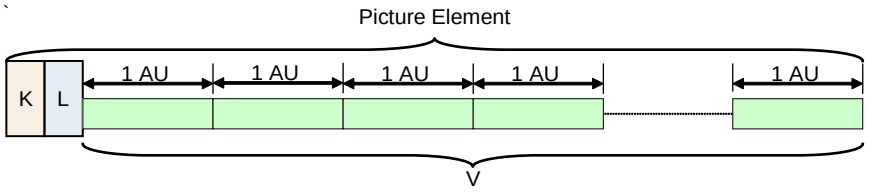

Figure 5 – Example of Clip Wrapping for an MPEG Picture Element

The Clip Wrapping method is intended for applications which carry the MPEG stream as a single large entity. This can be very useful in applications such as store and forward servers which process whole files and also in applications where it is desired to use the rich metadata structures of MXF as an annotation to MPEG data. The clip of MPEG data is KLV wrapped using an appropriate key as detailed in the KLV Coding for MPEG Streams other than Video or Audio section of this specification. When MPEG data is clip wrapped, there shall be only one clip per Generic Container. Multiple clips can be concatenated and edited using higher Operational Patterns detailed in SMPTE ST 377-1.

7.3 Custom Wrapping

A sequence of images shall be KLV coded as defined in SMPTE ST 336. One type of KLV wrapping, defined as Custom Wrapping, is specified for the MPEG body structure and described in this section.

In Custom Wrapping, KLV encoding wraps the MPEG data in an application specific way. All new wrapping modes shall be defined in a separate SMPTE standard and the associated key and label values registered in the SMPTE registry. Unless otherwise stated, each of the Custom Wrappings shall treat the MPEG video stream as the master for determining the start and end points of the Content Package. SMPTE ST 379-2 contains further details on requirements for elements within a Content Package and the use of a Generic Container.

A table of pre-defined Custom Wrapping modes shall be as defined in Table 1.

Table 1 – Custom wrapping descriptions

Name	Constraint	Application
Stripe	One KLV per macroblock row	Each Content Package contains an entire row of macroblocks. For some MPEG-1 video streams this mode may not be possible.
Fixed Audio Size	Fixed Audio Samples	The Sound Element is the Master Element. Wrapping is determined by a fixed number of audio samples. This can be used when the video rate is 1000/1001 * Integer_rate. The Picture Elements are fitted into the closest Content Package to their start time.
Splice	KLV at splice points	Each Content Package starts and ends at a valid MPEG splice point.
Closed GOP	KLV at closed GOPs	Each Content Package starts and ends at a closed GOP.
Slave	Slaved to another constraint	The audio is slave wrapped. For example a video elementary stream can be wrapped in chunks whose size corresponds to the original PES packet payload, and the associated audio is wrapped in chunks of approximately the same Duration as the video.
No specific constraint		No specific wrapping constraints were employed.

Note: Byte 16 of the MPEG Essence Container Label specified in Table 6 of this document describes the signalling for the specific custom wrapping listed in Table 1.

8 KLV Coding of MPEG Elements

8.1 KLV Coding for MPEG Video Streams

When using this specification, all MPEG video streams shall be wrapped using a Picture Element Key. All MPEG audio streams shall be wrapped using a Sound Element Key. All other MPEG streams shall be wrapped using a Data Element Key.

8.1.1 Picture Element Key

The values of the first 12 bytes of the Picture Element Key shall be as defined in SMPTE ST 379-2. The values of the last four bytes of the Picture Element Key shall be as defined in Table 2:

Table 2 – Key Value for the MPEG Picture Element

Byte No.	Description	Value (hex)	Meaning
1-12	Specified by the MXF Generic Container Specification, SMPTE ST 379-2		
13	Item Type Identifier	15h	Generic Container Picture Item (as defined in SMPTE ST 379-2)
14	Essence Element Count	kkh	Count of Picture Elements in this Picture Item
15	Essence Element Type	05h 06h 07h	Frame Wrapped Picture Element Clip Wrapped Picture Element Custom Wrapped Picture Element
16	Essence Element Number	nnh	The Number (used as an Index) of this Picture Element in this Picture Item

8.1.1.1 Essence Element Count – Byte 14

This shall be the count of the number of Picture Elements in the Picture Item of the Content Package.

8.1.1.2 Essence Element Type – Byte 15

For Frame Wrapped MPEG video AUs, this shall be 05_h. For Clip Wrapped MPEG video AUs, this shall be 06_h. For Custom Wrapped MPEG video AUs, this shall be 07_h.

8.1.1.3 Essence Element Number – Byte 16

This is a number used as an index to identify this instance of the element type within the item in each Content Package. The Generic Container specification requires that this byte be unique within an item. Each instance of the MPEG Picture Element shall have a constant Essence Element number throughout the entire Generic Container.

Note: The actual value of this number has no meaning; however, its uniqueness is important.

8.1.1.4 Picture Element Length

The length field of the KLV coded element shall be 4 bytes BER long-form encoded (i.e., 83_h.xx.yy.zz) for Frame Wrapping. The length field of the KLV coded element shall be 8 bytes BER long-form encoded (i.e., 87_h.aa.bb.cc.dd.ee.ff.gg) for Clip Wrapping. For Custom Wrapping, the length field shall be consistent and shall be either 4 bytes or 8 bytes as appropriate.

8.1.1.5 Picture Element Value

For Frame Wrapping, the value field of the KLV coded element shall be one Video AU or empty, having a Duration of one Video AU even if it is empty.. For Clip Wrapping, this shall be the entire MPEG video clip where the Duration shall be two or more Video AUs. For Custom Wrapping, the video will be wrapped in an application specific way, and the wrapping shall be defined in a separate SMPTE standard and the associated key and label values registered in the SMPTE registry.

8.2 KLV Coding for MPEG Audio Streams

8.2.1 Sound Element Key

The values of the first 12 bytes of the Sound Element Key shall be as defined in SMPTE ST 379-2. The values of the last four bytes of the Sound Element Key shall be as defined in Table 3.

Table 3 – Key Value for the MPEG Sound Element

Byte No.	Description	Value (hex)	Meaning
1-12	Specified by the MXF Generic Container Specification, SMPTE ST 379-2		
13	Item Type Identifier	16_h	Generic Container Sound Item (as defined in SMPTE ST 379-2)
14	Essence Element Count	kk_h	Count of Sound Elements in this Sound Item
15	Essence Element Type	05_h 06_h 07_h	Frame Wrapped Sound Element Clip Wrapped Sound Element Custom Wrapped Sound Element
16	Essence Element Number	nn_h	The Number (used as an Index) of this Sound Element in this Sound Item

8.2.1.1 Essence Element Count – Byte 14

This shall be the count of the number of Sound Elements in the Sound Item of the Content Package.

8.2.1.2 Essence Element Type – Byte 15

For Frame Wrapped MPEG audio AUs, this shall be 05_h. For Clip Wrapped MPEG audio AUs, this shall be 06_h. For Custom Wrapped MPEG audio AUs, this shall be 07_h.

8.2.1.3 Essence Element Number – Byte 16

This is a number used as an index to identify this instance of the element type within the item in each Content Package. The Generic Container specification requires that this byte be unique within an item. Each instance of the MPEG Sound Element shall have a constant Essence Element number throughout the entire generic container.

Note: The actual value of this number has no meaning; however, its uniqueness is important.

8.2.1.4 Sound Element Length

The length field of the KLV coded element shall be 4 bytes BER long-form encoded (i.e., 83_h.xx.yy.zz) for Frame Wrapping. The length field of the KLV coded element shall be 8 bytes BER long-form encoded (i.e., 87_h.aa.bb.cc.dd.ee.ff.gg) for Clip Wrapping. For Custom Wrapping the length field shall be consistent and shall be either 4 bytes or 8 bytes as appropriate.

8.2.1.5 Sound Element Value

For Frame Wrapping, the value field of the KLV coded element shall be the appropriate audio AU or audio AUs based on the requirements within SMPTE ST 379-2. For Clip Wrapping, this shall be the entire MPEG audio clip where the Duration shall be two or more audio AUs. For Custom Wrapping, the audio will be wrapped in an application specific way, and the wrapping shall be defined in a separate SMPTE standard and the associated key and label values registered in the SMPTE registry.

8.2.2 Multi-Channel Audio

There will frequently be multi-channel audio associated with MPEG video. This multi-channel audio may be coded as a single Essence Element (e.g., 5.1 AC-3 encoding) or may be coded as 2 or more separate Essence Elements.

Each audio Essence Element shall be mapped into its own Generic Container sound element, each of which follows the appropriate mapping and interleaving rules given elsewhere in this standard.

8.3 KLV Coding for MPEG Streams Other than Video or Audio

8.3.1 Data Element Key

The values of the first 12 bytes of the Data Element Key shall be as defined in SMPTE ST 379-2. The values of the last four bytes of the Data Element Key shall be as defined in Table 4.

Table 4 – Key Value for the MPEG Data Element

Byte No.	Description	Value (hex)	Meaning
1-12	Specified by the MXF Generic Container Specification, SMPTE ST 379-2		
13	Item Type Identifier	17_h	Generic Container Data Item (as defined in SMPTE ST 379-2)
14	Essence Element Count	kk_h	Count of Data Elements in this Data Item
15	Essence Element Type	05_h 06_h 07_h	Frame Wrapped Data Element Clip Wrapped Data Element Custom Wrapped Data Element
16	Essence Element Number	nn_h	The Number (used as an Index) of this Data Element in this Data Item

8.3.1.1 Essence Element Count – Byte 14

This shall be the count of the number of Data Elements in the Data Item of the Content Package.

8.3.1.2 Essence Element Type – Byte 15

For frame wrapped MPEG data, this shall be 05_h. For clip wrapped MPEG data, this shall be 06_h. For custom wrapped MPEG data, this shall be 07_h.

8.3.1.3 Essence Element Number – Byte 16

This is a number used as an index to identify this instance of the element type within the item in each Content Package. The Generic Container specification requires that this byte be unique within an item. Each instance of the MPEG Data Element shall have a constant Essence Element number throughout the entire generic container.

Note: The actual value of this number has no meaning; however, its uniqueness is important.

8.3.1.4 Data Element Length

The length field of the KLV coded element shall be 4 bytes BER long-form encoded (i.e., 83_h.xx.yy.zz) for Frame Wrapping. The length field of the KLV coded element shall be 8 bytes BER long-form encoded (i.e., 87_h.aa.bb.cc.dd.ee.ff.gg) for Clip Wrapping. For Custom Wrapping, the length field shall be consistent and shall be either 4 bytes or 8 bytes as appropriate.

8.3.1.5 Data Element Value

The value field of the KLV coded element shall be the Duration of one data unit for Frame Wrapping as described in the Interchange MXF file section of this document. For clip wrapping, this shall be the entire MPEG data clip where the Duration shall be two or more data units.

8.4 Use of KLV Fill Items

There may be KLV Fill items as defined in SMPTE ST 377-1 present in the Generic Container. For example, a KLV Fill item could be added to a VBE element to pad it out to a CBE Boundary or could be used to aid respecting KAG rules within particular implementations.

8.5 Use of KAG

There are no specific KAG requirements for this mapping. MXF encoders and decoders shall comply with the KAG rules in the SMPTE ST 377-1.

9 SMPTE Label for Essence Container Identification

9.1 MPEG Essence Container Label Definition

The values of the first 12 bytes of the MPEG Essence Container Label shall be as defined in SMPTE ST 379-2. The values for the MPEG Essence Container Label are given in Table 5.

Table 5 – Specification of the MPEG Essence Container Label

Byte No.	Description	Value (hex)	Meaning
1-12	Specified by the MXF Generic Container Specification, SMPTE ST 379-2		
13	Essence Container Kind	02_h	MXF Generic Container (as defined in SMPTE ST 379-2)
14	Mapping Kind	04_h $0F_h$ 10_h	MPEG ES AVC/H.264 NAL unit stream AVC/H.264 byte stream
15	Locally defined	xx_h	ISO/IEC 13818-1 stream_id bits 6..0
16	Locally defined	yy_h	Wrapping scheme Table 6

Note: The individual Essence Container ULs defined in this standard are registered in SMPTE RP 224 which readers can find in SMPTE RA web site at: http://smpte-ra.org/mdd/index.html.

This SMPTE label is the individual “Essence Container” Property used in the Partition Pack, in the Preface Set and in the appropriate File Descriptor.

Byte 14 identifies the container as the MPEG mapping into the Generic Container. It shall be set according to the values listed in Table 5.

Note: For more information on the mapping of the AVC/H.264 NAL unit stream specified as $0F_h$ in byte 14 and the AVC/H.264 byte stream specified as 10_h in byte 14, refer to SMPTE RP 2008 which describes the mapping of AVC/H.264 streams into the Generic Container.

Byte 15 enumerates the content of the element within the Essence Container. It is the value of bits 6..0 of the stream_id as specified by ISO/IEC 13818-1. Byte 15 shall have the correct value of stream_id of the PES packet(s) where the MPEG ES originated from, or is destined for.

Note: To uniquely determine what kind of Essence is within any container, an application uses the Essence Descriptor. The stream_id value used here is intended to give a rough guide as to the MPEG content. The stream_id may indicate private data or the stream_id extension mechanism. As is required by SMPTE ST 377-1, full details of the precise signaling will be found in the Essence Descriptor. Specifications which map new, extended or private MPEG data types can use this specification and an extension to one of the MXF Essence Descriptors to create the mapping.

Byte 16 is intended to carry basic information about the containment of the MPEG data. It shall be set according to Table 6.

Table 6 – Specification of the MPEG Essence Container Label, byte 16

Byte 16	Meaning
00h	Not used
01h	Frame Wrapping
02h	Clip Wrapping
03h	Custom: Stripe Wrapping
05h	Custom: Fixed Audio Size
06h	Custom: Splice
07h	Custom: Closed GOP
08h	Custom: Slave
09h - 7Eh	Reserved
7Fh	Custom: No specific wrapping constraints

10 Essence Descriptors for MPEG Mappings

10.1 MPEG Descriptors

In some MXF files, MPEG picture data may also exist with Sound and Data Elements in the Generic Container. This creates the need for a Multiple Descriptor in the File Package which describes the Essence. Accordingly, the Multiple Descriptor references a descriptor for each of the elements in the Generic Container.

Within SMPTE ST 377-1, a Generic Picture Essence Descriptor is used to describe Picture Essence, a Generic Sound Essence Descriptor is used to describe Sound Essence, and a Generic Data Essence Descriptor is used to describe Data Essence. Within the Generic Picture Essence Descriptor, the Picture Essence Coding Label shall be encoded to identify the precise coding of the MPEG Picture Essence. Within the Generic Sound Essence Descriptor, the Sound Essence Coding Label shall be encoded to identify the precise coding of the MPEG Sound Essence. Within the Generic Data Essence Descriptor, the Data Essence Coding Label is used to identify the precise coding of the Data Essence.

Note: In SMPTE ST 377-1, the Picture Essence Coding Label and Sound Essence Coding Label properties are D/req properties. However, for SMPTE ST 381-2 as specified above, the Picture Essence Coding Label is a Req property for MPEG Picture Essence, and the Sound Essence Coding Label is a Req property for MPEG Sound Essence because the essence coding labels are essential to determine the nature of the essence element payload within this standard.

Note: In SMPTE RP 2008, the values for the Picture Essence Coding Label for AVC/H.264, which is used in the Generic Picture Essence Descriptor, are described. The corresponding values are listed in the SMPTE Labels Register, RP 224.

An MPEG Descriptor provides a mechanism for mapping some MPEG Descriptors to MXF structural metadata. An MPEG Video Descriptor and an MPEG Audio Descriptor are described in the following sections.

10.2 MPEG Video Descriptor

For MPEG video, the MPEG Video Descriptor, which extends the CDCI Descriptor, should be used. The MPEG Video Descriptor consists of the CDCI Descriptor and MPEG video specific properties as shown in Table 7. The optional properties in Table 7 shall only be used when their values apply to the entire Essence described. Therefore, these properties allow fixed picture patterns to be identified. Dynamically varying descriptor properties shall be specified separately.

Table 7 – MPEG Video Descriptor

Item Name	Type	Len	Item UL	Req ?	Meaning	Default
MPEG Video Descriptor	Set UL	16	See table below	Req	Defines the MPEG Video Descriptor Set	
Length	BER Length	4		Req	Set length	
All items from the MXF CDCI Descriptor as specified in SMPTE ST 377-1						
Single Sequence Flag	Boolean	1	06.0E.2B.34.01.01.01.05.04.01.06.02.01.02.00.00	Opt	TRUE if all sequence headers in the Essence stream are identical. FALSE if there are differences among sequence headers within the Essence stream. This flag indicates that the sequence header information is not varying in the Essence stream.	
Constant B Picture Flag	Boolean	1	06.0E.2B.34.01.01.01.05.04.01.06.02.01.03.00.00	Opt	TRUE if the number of B frames is always constant.	
Coded Content Scanning Kind	Enum	1	06.0E.2B.34.01.01.01.05.04.01.06.02.01.04.00.00	Opt	0= "Unknown" 1= "Progressive" 2= "Interlaced" 3= "Mixed" An enumerated value which tells if the underlying content which was MPEG coded was of a known type.	
Low Delay Indicator	Boolean	1	06.0E.2B.34.01.01.01.05.04.01.06.02.01.05.00.00	Opt	TRUE if low delay mode is set in the sequence header extension	
Closed GOP Indicator	Boolean	1	06.0E.2B.34.01.01.01.05.04.01.06.02.01.06.00.00	Opt	TRUE if closed_gop is set in all GOP Headers, per ISO/IEC 13818-1 IBP descriptor.	
Identical GOP Indicator	Boolean	1	06.0E.2B.34.01.01.01.05.04.01.06.02.01.07.00.00	Opt	TRUE if every GOP in the sequence is constructed the same, per ISO/IEC 13818-1 IBP descriptor.	
Maximum GOP Size	UInt16	2	06.0E.2B.34.01.01.01.05.04.01.06.02.01.08.00.00	Opt	Specifies the maximum occurring spacing between I frames, per ISO/IEC 13818-1 IBP descriptor. A value of 0 or the absence of this Property indicates either there is no limit to the maximum GOP or the maximum GOP is unknown.	
Maximum B Picture Count	UInt16	2	06.0E.2B.34.01.01.01.05.04.01.06.02.01.09.00.00	Opt	Specifies the maximum number of B pictures between P or I frames, equivalent to ISO/IEC 13818-2 annex D (M-1).	
Bit Rate	UInt32	4	06.0E.2B.34.01.01.01.05.04.01.06.02.01.0B.00.00	Opt	Maximum bit rate of MPEG video ES in bit/s. An example is defined in ISO/IEC 13818-2 bit_rate Property.	
Profile And Level	UInt8	1	06.0E.2B.34.01.01.01.05.04.01.06.02.01.0A.00.00	Opt	Specifies the MPEG video profile and level. The value is taken directly from the profile_and_level_indication in the MPEG sequence header extension. For example, within the main profile @ main level within MPEG-2, the value is 0x48, and for the 4:2:2 profile @ main level within MPEG-2, the value is 0x85.	

Note: In some legacy cases, the CDCI Descriptor has been used to describe MPEG video instead of the MPEG Video Descriptor.

In addition, this Descriptor is called the MPEG Video Descriptor because it carries many of the video properties which were originally standardized in the MPEG-1 and MPEG-2 standards. These properties can be used with any Picture Essence which can be mapped by this document, such as MPEG-1, MPEG-2, MPEG-4 Visual, AVC/H.264, or in documents that use this document as a reference, such as SMPTE RP 2008 which describes the mapping of AVC/H.264 streams.

Note: For MPEG-4 Visual, the MPEG-4 Visual Sub Descriptor, as described in Section 10.2, ought to be used.

Note: Many of these properties in Table 7 appear in the ISO/IEC 13818-1 IBP descriptor. They can be specified even if the descriptor is not present in the MPEG stream.

10.2.1 Key for the MPEG Video Descriptor

The key (UL) for this Local Set shall be as defined in Table 8.

Table 8 – Key for MPEG Video Descriptor

Byte No.	Description	Value (hex)	Meaning
1-13	Defined in the Structural Header Metadata Implementation section of SMPTE ST 377-1		
14	Set Kind (1)	01h	MPEG Video Descriptor
15	Set Kind (2)	51h	
16	Reserved	00h	Reserved

10.3 MPEG-4 Visual Sub Descriptor

For MPEG-4 Visual, the MPEG-4 Visual Sub Descriptor, which is (strongly) referenced from the CDCI Descriptor or RGBA Descriptor, should be used. The MPEG-4 Visual Sub Descriptor consists of MPEG-4 Visual specific properties as shown in Table 9. The optional properties in Table 9 shall only be used when their values apply to the entire Essence described. Therefore, these properties allow fixed picture patterns to be identified. Dynamically varying descriptor properties shall be specified separately.

Note: In some legacy cases, the CDCI Descriptor or the MPEG Video Descriptor has been used to describe MPEG-4 Visual instead of the MPEG-4 Visual Sub Descriptor.

Table 9 – MPEG-4 Visual Sub Descriptor

Item Name	Type	Len	Item UL	Req ?	Meaning	Default
MPEG-4 Visual Sub Descriptor	Set UL	16	See table below	Req	Defines the MPEG-4 Visual Sub Descriptor Set	
Length	BER Length	4		Req	Set length	
All items from the Sub Descriptor set (except for the Group UL) as specified in SMPTE ST 377-1						
MPEG-4 Visual Single Sequence	Boolean	1	06.0E.2B.34.01.01.01.0D.04.01.06.02.02.02.00.00	Opt	TRUE if all configuration information (i.e. VOSs, VOs and VOLs) in the Essence stream are identical except vbv_occupancy (i.e. first_half_vbv_occupancy and latter_half_vbv_occupancy). FALSE if there are differences among configuration information within the Essence stream except vbv_occupancy.	
MPEG-4 Visual Constant B-VOPs	Boolean	1	06.0E.2B.34.01.01.01.0D.04.01.06.02.02.03.00.00	Opt	TRUE if the number of B-VOPs is always constant.	
MPEG-4 Visual Coded Content Type	Enum	1	06.0E.2B.34.01.01.01.0D.04.01.06.02.02.04.00.00	Opt	0= "Unknown" 1= "Progressive" 2= "Interlaced" 3= "Mixed" an enumerated value which tells if the underlying content which was MPEG coded was of a known type.	
MPEG-4 Visual Low Delay	Boolean	1	06.0E.2B.34.01.01.01.0D.04.01.06.02.02.05.00.00	Opt	TRUE if the VOL contains no B-VOPs.	
MPEG-4 Visual Closed GOV	Boolean	1	06.0E.2B.34.01.01.01.0D.04.01.06.02.02.06.00.00	Opt	TRUE if closed_gov is set in all GOV headers. An example is defined in closed_gop_flag of ISO/IEC 13818-1 IBP descriptor.	
MPEG-4 Visual Identical GOV	Boolean	1	06.0E.2B.34.01.01.01.0D.04.01.06.02.02.07.00.00	Opt	TRUE if every GOV in the sequence is constructed the same. An example is defined in identical_gop_flag of ISO/IEC 13818-1 IBP descriptor.	
MPEG-4 Visual Max GOV	UInt16	2	06.0E.2B.34.01.01.01.0D.04.01.06.02.02.08.00.00	Opt	Specifies the maximum occurring spacing between I frames. An example is defined in max_gop_length of ISO/IEC 13818-1 IBP descriptor. A value of 0 or the absence of this Property implies no limit to the maximum GOV or the maximum GOV is unknown.	
MPEG-4 Visual B-VOP Count	UInt16	2	06.0E.2B.34.01.01.01.0D.04.01.06.02.02.09.00.00	Opt	Specifies the maximum number of B-VOPs between P or I-VOPs.	
MPEG-4 Visual Bit Rate	UInt32	4	06.0E.2B.34.01.01.01.0D.04.01.06.02.02.0B.00.00	Opt	Maximum bit rate of the MPEG-4 Visual stream in bit/s	
MPEG-4 Visual Profile And Level	UInt8	1	06.0E.2B.34.01.01.01.0D.04.01.06.02.02.0A.00.00	Opt	Specifies the MPEG-4 Visual profile and level. The meaning of the bits is given in Table G.1 of ISO/IEC 14496-2.	

10.3.1 Key for the MPEG-4 Visual Sub Descriptor

The key (UL) for this Local Set shall be as defined in Table 10.

Table 10 – Key for MPEG-4 Visual Sub Descriptor

Byte No.	Description	Value (hex)	Meaning
1-13	Defined in the Structural Header Metadata Implementation section of SMPTE ST 377-1		
14	Set Kind (1)	01_h	MPEG-4 Visual Sub Descriptor
15	Set Kind (2)	68h	
16	Reserved	00_h	Reserved

10.3.2 Class model of the MPEG-4 Visual Sub Descriptor (Informative)

The Inheritance hierarchy of the CDCI Descriptor, the RGBA Descriptor and the MPEG-4 Visual Sub Descriptor is illustrated in Figure 6 using UML notation. Figure 6 also illustrates how the MPEG-4 Visual Sub Descriptor can be strongly referenced from either the CDCI Descriptor or the RGBA Descriptor.

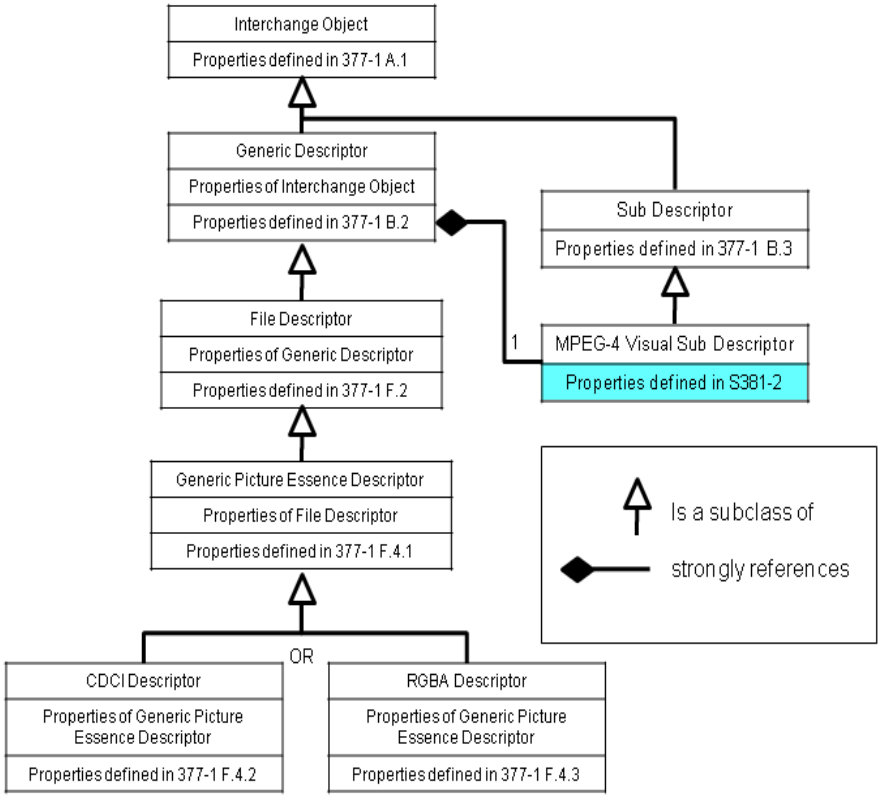

Figure 6 – Class diagram for the MPEG-4 Visual Sub Descriptor

10.4 MPEG Audio Descriptor

For MPEG audio, the MPEG Audio Descriptor, which extends the Sound Essence Descriptor, shall be used. The MPEG Audio Descriptor consists of the Sound Essence Descriptor and MPEG audio specific properties as shown in Table 11. The MPEG Audio Descriptor is used to describe MPEG audio streams, such as MPEG-1 Layer II compressed audio as described in MPEG-1 Part 3 or AAC as described in MPEG-2 Part 7 or MPEG-4 Part 3. The MPEG Audio Bit Rate Property in Table 11 shall be used only when the audio bitstream has a constant bitrate throughout the audio bitstream described. Therefore, if the Property is not present, the bitstream has a variable bitrate.

Table 11 – MPEG Audio Descriptor

Item Name	Type	Len	Item UL	Req ?	Meaning	Default
MPEG Audio Descriptor	Set UL	16	See table below	Req	Defines the MPEG Audio Descriptor Set	
Length	BER Length	4		Req	Set length	
All items from the MXF SoundEssence Descriptor as specified in SMPTE ST 377-1						
MPEG Audio Bit Rate	UInt32	4	06.0E.2B.34.01.01.01.0A.04.02.04.03.01.02.00.00	Opt	Specifies the rate of the compressed audio bitstream in kbit/s.	

10.4.1 Key for the MPEG Audio Descriptor

The key (UL) for this Local Set shall be as defined in Table 12.

Table 12 – Key for MPEG Audio Descriptor

Byte No.	Description	Value (hex)	Meaning
1-13	Defined in the Structural Header Metadata Implementation section of SMPTE ST 377-1		
14	Set Kind (1)	01h	MPEG Audio Descriptor
15	Set Kind (2)	5Eh	
16	Reserved	00h	Reserved

11 Index Table Requirements for MPEG Mappings

Index Tables for MPEG mappings are discussed within Annex B of this document. Annex B gives clarifications and some worked examples for the use of Index Tables for MPEG mappings. This section gives some requirements when Index Tables are used for MPEG mappings.

11.1 Setting the Properties in an Index Entry

There are several properties in the IndexEntry which have specific meanings for a long GOP MPEG Index Table. The Flags Property, shown in Table 13, shall be correctly set according to the text below Table 13 which describes the conditions for setting the flags. Annex C in this document describes how these parameters can be used to correctly identify the MPEG picture type from these properties once they have been correctly set.

Note: Flag bits 3-0 are reserved for use in SMPTE Essence mapping specifications in SMPTE ST 377-1 and, therefore, the definitions for MPEG Essence mapping are given in this section.

Table 13 – Index Table Entry Properties

Parameter	Type	Meaning	Use
Temporal Offset	Int8	Offset in Edit Units from Display Order to Coded Order (see SMPTE ST 377-1 for usage)	Used to find the IndexEntry for a stored Picture given its Display Temporal position.
Key Frame Offset	Int8	Offset in Edit Units to previous Key Frame (i.e. Key Frame or I frame). The value is zero if this is a Key frame.	The offset to the Key Frame
Flags	EditUnitFlag	Flags for this Edit Unit Bit 7: Random Access Bit 6: Sequence Header Bit 5: Forward prediction flag Bit 4: Backward prediction flag Bit 3: Offsets out of range Bit 2: Not used by MPEG Bits 0,1: MPEG Frame type	e.g. naïve settings for Bits 5, 4: 00 == I frame 10 == P frame 11 == B frame is identical to the MPEG frame type bits 1,0 00 == I frame 10 == P frame 11 == B frame

When using Index Tables for MPEG mappings, the Edit Rate of the indexed Tracks is further constrained so the Temporal Offset and Key Frame Offset properties shall be integer numbers.

Temporal Offset: This is an offset used to allow lookups in the Index Table. It corresponds to the difference between the display order and transmission order for the indexed picture measured in Edit Units, not in frames. This is explained in the text of SMPTE ST 377-1.

If the numerical range of this Property is exceeded, then bit 3 of the flags shall be set, and this Property shall be clipped to the maximum value which can be represented. The actual temporal offset can be determined by inspecting IndexEntries for neighboring pictures until a valid temporal offset is found. The value of temporal offset can be determined by the number of index entries inspected, their picture types and the valid temporal offset value found.

Key Frame Offset: This is the offset measured in Edit Units (not frames) to the key frame (I frame) required for decoding the indexed picture. If there are no key frames used in the MPEG coding, then this parameter shall be set to zero. If the numerical range of this Property is exceeded, then bit 3 of the flags shall be set, and this Property shall be clipped to the maximum value which can be represented. The actual key frame offset can be determined by inspecting IndexEntries for neighboring pictures until a valid key frame offset is found. The value of key frame offset can be determined by the number of IndexEntries inspected, their picture types and the valid key frame offset value found.

If the stream consists of I pictures only, then this parameter shall be set to zero since there would be no key frames required to decode an indexed picture. If there are no I pictures in the stream, then this parameter shall be set to a constant value which allows the current indexed frame to be decoded. This will depend on the percentage of I macroblocks and their distribution throughout the image.

Key frame offset is always negative for MPEG-2 streams. The offset is measured in Edit Units and is the transmission order offset of the key frame index entry. This keeps the meaning of the sign of key frame offset consistent with the sign of temporal offset. It also means that the only display order process is to use the temporal offset to find the appropriate index entry. Index entries are always stored in transmission order.

Flag Bit 7: This is the random access bit and shall be set when a random access point in the long GOP MPEG stream is encountered. A random access point is one where decoding can commence at that point in the stream. A sequence_header and a closed_GOP shall be present at the indexed picture.

Flag Bit 6: This bit marks a sequence_header in the long GOP MPEG stream. This bit shall be set if the random access bit is set. If the indexed picture has a sequence_header, but occurs at an open_GOP, then this bit shall be set and the random access bit shall not be set. While decoding may begin at this point, there may be some invalid content decoded until a key frame is displayed, or all the macroblocks on the screen have been refreshed.

Flag Bits 5 and 4: These bits indicate the temporal dependence of the indexed frame. These bits may be set naïvely or may be set strictly by an encoder depending on picture type. The strict settings are used by applications which require higher levels of random access performance. The naïve settings shall be used by an encoder when strict setting is not possible. The naïve setting of these bits by an encoder shall be as follows:

Naïve setting of bits 5 and 4:

I frame	00	(no prediction)
P frame	10	(forward prediction from previous frame)
B frame	11	(forward and backward prediction)

Strict setting of bits 5 and 4 is used to improve performance of the system. The MPEG bitstream needs to be parsed and each of the motion vectors for each macroblock inspected. They are categorized into “no prediction”, “forward prediction” and “backward prediction”. The strict setting of these bits by an encoder shall be as follows:

00 No prediction. This is always the case for I frames. P frames and B frames consisting only of I macroblocks should also set these flags to 0.

10 Forward prediction from previous frame. This is the case for P frames which have non zero motion vectors, or B frames which have only forward prediction motion vectors.

01 Backward prediction to future frame. This is the case for B frames which commence a closed GOP.

11 Forwards and backwards prediction. This is the general case for bi-directionally predicted B frames.

Flag Bit 3: This bit shall indicate that the numerical range of the temporal offset or key frame fields has been exceeded. In this case, the offset and/or key frame need to be determined by inspecting other index entries to determine the precise values.

Flag Bit 2: This bit shall not be used by the MPEG mapping and is reserved for future use or use by other mappings. Therefore, it shall be set to 0.

Flag Bits 1 and 0: These bits indicate MPEG picture type. The setting of these bits shall be as follows:

I frame	00	(no prediction)
P frame	10	(forward prediction from previous frame)
B frame	11	(forward and backward prediction)

Annex A Bibliography (Informative)

Note: All references in this document to other SMPTE documents use the current numbering style (e.g. SMPTE ST 298:2009) although, during a transitional phase, the document as published (printed or PDF) may bear an older designation (such as SMPTE 298-2009). Documents with the same root number (e.g. 298) and publication year (e.g. 2007) are functionally identical.

SMPTE ST 298:2009, Universal Labels for Unique Identification of Digital Data

SMPTE ST 312:2001, Television — Splice Points for MPEG-2 Transport Streams

SMPTE ST 328:2000, Television — MPEG-2 Video Elementary Stream Editing Information

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

SMPTE ST 382:2007, Material Exchange Format (MXF) — Mapping AES3 and Broadcast Wave Audio into the MXF Generic Container

SMPTE RP 210, Metadata Dictionary Registry of Metadata Element Descriptions

SMPTE RP 224, SMPTE Labels Register

SMPTE RP 2008:2008, Material Exchange Format – Mapping AVC Streams into the MXF Generic Container

Advanced Authoring Format (AAF): http://www.AAFassociation.org

AES3-2009, AES Standard for Digital Audio Engineering — Serial Transmission Format for Two-Channel Linearly Represented Digital Audio Data

EBU R122-2007, Material Exchange Format Timecode Implementation

ETSI EN 300 743 (2006-11) – Digital Video Broadcasting (DVB); Subtitling Specification

ISO/IEC 11172-2:1993, Information Technology — Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to about 1,5 Mbit/s — Part 2: Video

ISO/IEC 11172-3:1993, Information Technology — Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to about 1,5 Mbit/s — Part 3: Audio

ISO/IEC 14496-3:2009, Information Technology — Coding of Audio Visual Objects — Part 3:Audio

ITU-R BS.1196-2 (03/10), Audio Coding for Digital Terrestrial Television Broadcasting

Annex B Index Tables for MPEG Mappings (Informative)

The MXF Index Table mechanism provides a deterministic method of identifying the key frames and dependencies of each indexed frame within the file. While Index Tables are optional, the user recommendations for the design of the MXF specification stated that Index Tables ought to be present in a file. The Index Tables are the primary mechanism for determining key frames and dependencies. SMPTE ST 377-1 contains details on the Index Table specification.

This annex gives clarifications and some worked examples for the use of Index Tables for CBE Essence and Index Tables for VBE Essence in MXF. Entries in Index Tables will often point to start of the synchronized data for each element. Due to frame reordering in long GOP MPEG and that audio AUs might not be the same Duration as video AUs, the synchronized data could be physically located in different Content Packages. As stated previously, when the compressed audio item is not synchronized with the picture item, the offset in time from the start of the audio Content Item to the Picture Content Item can be expressed using the fractional position offset mechanism defined in SMPTE ST 377-1 if an Index Table is constructed.

B.1 Edit Units and Index Tables

For interleaved Essence, SMPTE ST 377-1 constrains the Edit Rates of all indexed Tracks to be identical. This mapping document further constrains the Edit Rate of the indexed Tracks so the temporal offset and key frame offset properties are integer numbers as specified in Section 11. Usually the Edit Rate of a Track will be an MPEG video frame, but sometimes greater flexibility is achieved if the Edit Rate is a video field.

B.2 Index Table Example for Frame Wrapped CBE MPEG Essence

Figure B.1 illustrates an example of the Index Table relationships for the two Edit Units in an interleaved CBE MPEG stream using frame wrapping. In this example, each interleaved Essence Container consists of System, Picture, Sound, and Data Elements. Since the overall length of all the elements in each frame is constant, "Edit Unit Byte Count" item and Delta Entries are sufficient to index each Edit Unit and provide information to find the elements within an interleaved Content Package.

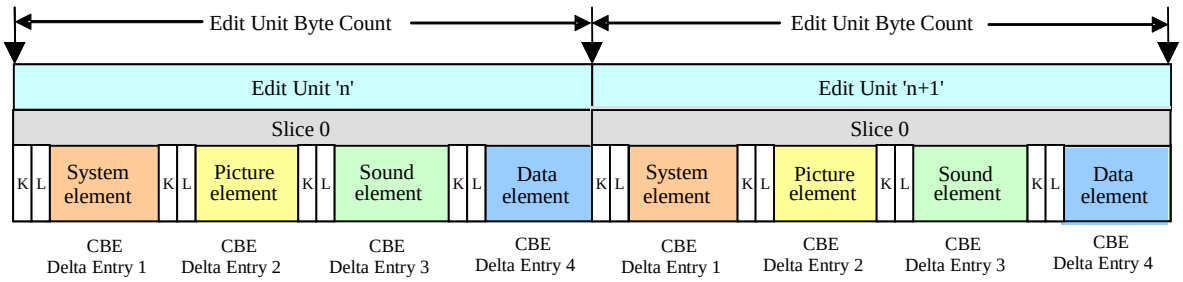

Figure B.1 – Index Table Relationships for Frame Wrapped CBE MPEG Essence

Table B.1 - Frame Wrapped Index Table Segment Set for CBE Example

Item Name	Req ?	Meaning	Use
Index Table Segment	Req	An Index Table Segment Set	See MXF Format Specification – SMPTE ST 377-1
Length	Req	Set Length	See MXF Format Specification – SMPTE ST 377-1
Instance ID	Req	Unique ID of this instance	See MXF Format Specification – SMPTE ST 377-1
Index Edit Rate	Req	Edit Rate copied from the Essence Tracks of the Essence Container	See MXF Format Specification – SMPTE ST 377-1
Index Start Position	Req	The first Edit Unit indexed by this Index Table Segment measured in Top-Level File Package Edit Units relative to the first Edit Unit in the Essence Container	This defines the temporal start point for an Index Table Segment
Index Duration	Req	Time Duration of this Index Table Segment measured in Edit Units of the referenced Package	Combined with Start Position, this allows an application to determine if this Index Table Segment spans a particular Position value
Edit Unit Byte Count	D/Req	Defines the byte count of each and every Edit Unit. A value of 0 defines the byte count of Edit Units is only given in the Index Entry Array	Byte count of each Edit Unit
IndexSID	D/Req	Stream Identifier (SID) of Index Table	See MXF Format Specification – SMPTE ST 377-1
BodySID	Req	Stream Identifier (SID) of the indexed Essence Container	See MXF Format Specification – SMPTE ST 377-1
Slice Count	D/Req	Number of Slices minus 1 (NSL)	0: Slice 0
Delta Entry Array	Opt	Map Elements onto Slices	See Table B.3 below
Index Entry Array	D/Req	Index from Edit Unit number to stream offset	Not encoded

Table B.2 - Frame Wrapped Delta Entry Array for CBE Example

	Field Name	Type	Meaning	Use
	NDE	UInt32	Number of Delta Entries	4
	Length	UInt32	Length of each Delta Entry	6
System Delta Entry	PosTableIndex	Int8	0= No reordering	0
	Slice	UInt8	Slice number in IndexEntry	0
	Element Delta	UInt32	Delta from start of slice to this Element	0
Picture Delta Entry	PosTableIndex	Int8	0= No reordering	0
	Slice	UInt8	Slice number in IndexEntry	0
	Element Delta	UInt32	Delta from start of slice to this Element	sizeof(KL) + sizeof(System)
Sound Delta Entry	PosTableIndex	Int8	0= No reordering	0
	Slice	UInt8	Slice number in IndexEntry	0
	Element Delta	UInt32	Delta from start of slice to this Element	sizeof(KL) + sizeof(System) + sizeof(KL) + sizeof(Picture)
Data Delta Entry	PosTableIndex	Int8	0= No reordering	0
	Slice	UInt8	Slice number in IndexEntry	0
	Element Delta	UInt32	Delta from start of slice to this Element	sizeof(KL) + sizeof(System) + sizeof(KL) + sizeof(Picture) + sizeof(KL) + sizeof(Sound)

B.3 Index Table Example for Clip Wrapped CBE MPEG Essence

Figure B.2 illustrates an example of the Index Table relationships for a CBE MPEG stream using clip wrapping. In this example, the Essence Container consists of a Picture, Sound and Data Element, which is wrapped in independent KLV. The Delta Entry values are limited to a UInt32 number range and will therefore be incapable of specifying the offset from the Picture Item of frame n to the other Items of frame n. Therefore, the Index Table Segment is sliced, even though the offset to the other Items are constant. If a single Essence Element is contained in the Essence Container, the "Edit Unit Byte Count" item is sufficient to index each Edit Unit.

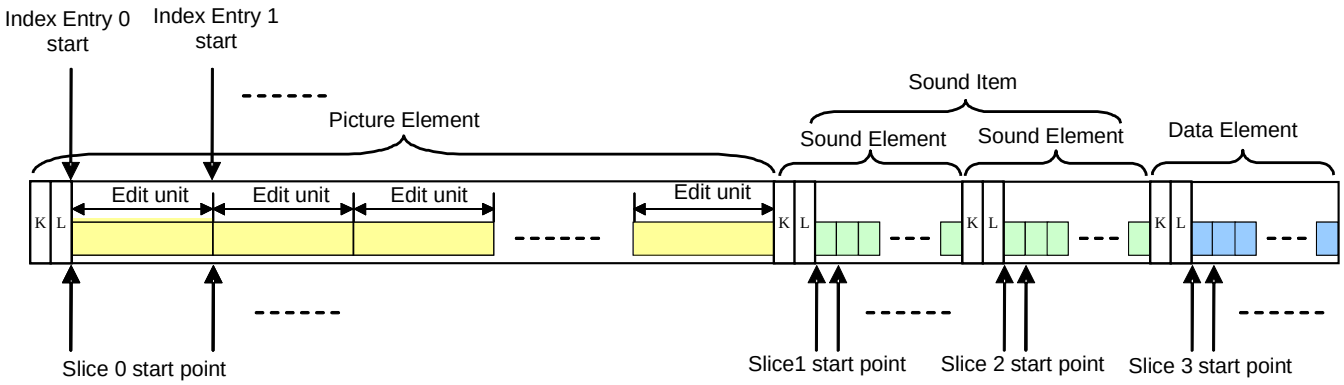

Figure B.2 – Index Table Relationships for Clip Wrapped CBE MPEG Essence

Table B.3 - Clip Wrapped Index Table Segment Set for CBE Example

Item Name	Req ?	Meaning	Use
Index Table Segment	Req	An Index Table Segment Set	See MXF Format Specification – SMPTE ST 377-1
Length	Req	Set Length	See MXF Format Specification – SMPTE ST 377-1
Instance ID	Req	Unique ID of this instance	See MXF Format Specification – SMPTE ST 377-1
Index Edit Rate	Req	Edit Rate copied from the Tracks of the Essence Container	See MXF Format Specification – SMPTE ST 377-1
Index Start Position	Req	The first Edit Unit indexed by this Index Table Segment measured in Top-Level File Package Edit Units relative to the first Edit Unit in the Essence Container	This defines the temporal start point for this Index Table Segment
Index Duration	Req	Time Duration of this Index Table Segment measured in Edit Units of the referenced Package	Combined with Start Position, this allows an application to determine if this Index Table Segment spans a particular Position value
Edit Unit Byte Count	D/Req	Defines the byte count of each and every Edit Unit. A value of 0 defines the byte count of Edit Units is only given in the Index Entry Array	0 : Given in the Index Entry Array
IndexSID	D/Req	Stream Identifier (SID) of Index Table	See MXF Format Specification – SMPTE ST 377-1
BodySID	Req	Stream Identifier (SID) of the indexed Essence Container	See MXF Format Specification – SMPTE ST 377-1
Slice Count	D/Req	Number of Slices minus 1 (NSL)	3 (one slice per element)
PosTableCount	Opt	Number of PosTable Entries minus 1 (NPE)	In this example, there is only a single Content Package and the audio and video start synchronously, so NPE=0
Delta Entry Array	Opt	Map Elements onto Slices	See Table B.5
Index Entry Array	D/Req	Index from Edit Unit number to stream offset	See Table B.6

In Clip wrapping mode, the Element Delta values do not include the lengths of the "KL" for each Element. The result is that each Element Delta points to the first byte of the value in the KLV which wraps an element.

Table B.4 - Clip Wrapped Delta Entry Array for CBE Example

	Field Name	Type	Meaning	Use
	NDE	UInt32	Number of Delta Entries	4
	Length	UInt32	Length of each Delta Entry	6
Picture Delta Entry	PosTableIndex	Int8	0= No reordering	0
	Slice	UInt8	Slice number in IndexEntry	0
	Element Delta	UInt32	Delta from start of slice to this Element	0
Sound Delta Entry	PosTableIndex	Int8	0= No reordering	0
	Slice	UInt8	Slice number in IndexEntry	1
	Element Delta	UInt32	Delta from start of slice to this Element	0
Sound Delta Entry	PosTableIndex	Int8	0= No reordering	0
	Slice	UInt8	Slice number in IndexEntry	2
	Element Delta	UInt32	Delta from start of slice to this Element	0
Data Delta Entry	PosTableIndex	Int8	0= No reordering	0
	Slice	UInt8	Slice number in IndexEntry	3
	Element Delta	UInt32	Delta from start of slice to this Element	0

Table B.5 - Clip Wrapped Index Entry Array for CBE Example

	N	Field Name	Type	Meaning	Use
	1	NIE	UInt32	Number of index entries	Equal to number of frames
	1	Length	UInt32	Length of each Index Array entry	23 in this example
One Index Entry for every frame	NIE	Temporal Offset	Int8	Offset in Edit Units from Display Order to Coded Order	Set according to picture type
		Key Frame Offset	Int8	Offset in Edit Units to previous Key Frame. The value is zero if this is a Key Frame.	Set according to picture type
		Flags	EditUnitFlag	Flags for this Edit Unit	Set according to picture type, Default is 80h
		Stream Offset	UInt64	Offset in bytes from the first KLV element in this Edit Unit within the Essence Container Stream	Offset from the first byte of the value of the KLV for the first MPEG frame to the first byte of the stored data for this MPEG frame in Figure B.2
		SliceOffset	UInt32	The offset in bytes from the Stream Offset to the start of slice 1	Offset from the first byte of the stored data for this MPEG frame to the first byte of the corresponding audio sample in the first Sound Element in Figure B.2
		SliceOffset	UInt32	The offset in bytes from the Stream Offset to the start of slice 2	Offset from the first byte of the stored data for this MPEG frame to the first byte of the corresponding audio sample in the second Sound Element in Figure B.2

N	Field Name	Type	Meaning	Use
	SliceOffset	UInt32	The offset in bytes from the Stream Offset to the start of slice 3	Offset from the first byte of the stored data for this MPEG frame to the first byte of the corresponding audio sample in the Data Element in Figure B.2
	PosTable	NPE *Rational	The fractional position offset from the start of the Content Package to the synchronized sample in the Content Package	In this example, there is only a single Content Package and the audio and video start synchronously. NPE is zero so this is zero

B.4 Index Table Example for Frame Wrapped VBE MPEG Essence with Extra Items in the Generic Container

This is a case where the Index Table points to the first byte of the MPEG Picture Element Generic Container Key. The other Generic Container elements can be indexed by correct use of the delta entries and index entries. This example assumes that the Sound Elements which are indexed require the use of the fractional position mechanism defined in SMPTE ST 377-1. Figure B.3 shows an example of a Content Package being indexed.

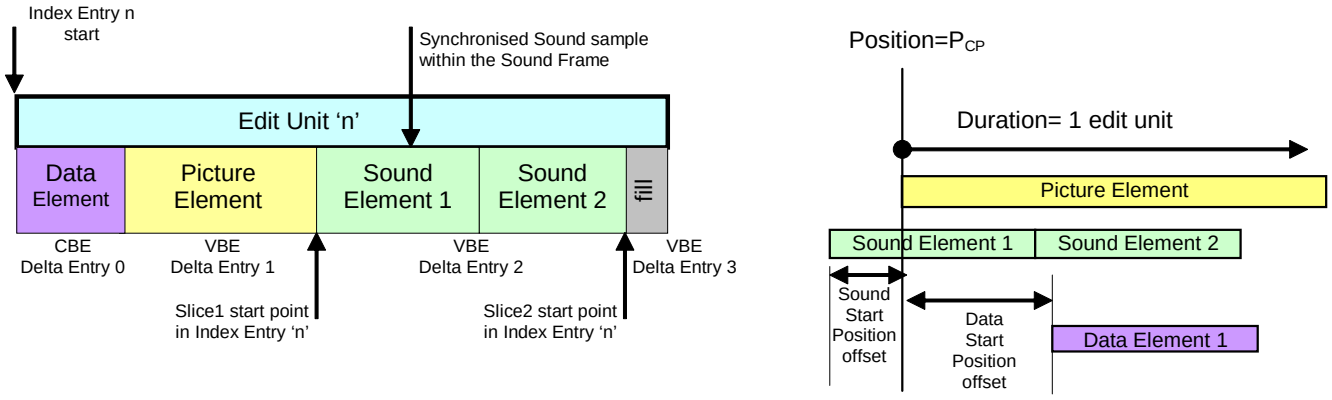

Figure B.3 – Content Package for Index Table VBE Example

In this example, the picture, sound and fill are all VBE. The fill is indexed so that it can be eliminated from any Essence byte counting based solely on calculations in the Index Table.

Table B.6 – Frame Wrapped Index Table Segment Set for Frame Wrapped VBE Example

Item Name	Req ?	Meaning	Use
Index Table Segment	Req	An Index Table Segment set	See MXF Format Specification – SMPTE ST 377-1
Length	Req	Set Length	See MXF Format Specification – SMPTE ST 377-1
Instance ID	Req	Unique ID of this instance	See MXF Format Specification – SMPTE ST 377-1
Index Edit Rate	Req	Edit Rate copied from the Tracks of the Essence Container	See MXF Format Specification – SMPTE ST 377-1
Index Start Position	Req	The first Edit Unit indexed by this Index Table Segment measured in Top-Level File Package Edit Units relative to the first Edit Unit in the Essence Container	This sets the temporal start point for an Index Table Segment
Index Duration	Req	Time Duration of this Index Table Segment measured in Edit Units of the referenced Package	Combined with Start Position, this allows an application to determine if this Index Table Segment spans a particular Position value
Edit Unit Byte Count	D/Req	Defines the byte count of each and every Edit Unit. A value of 0 defines the byte count of Edit Units is only given in the Index Entry Array	0 unless the total length of all the Generic Container Elements are of constant size. In this example it is assumed that only the Data Elements are VBR so the value is 0.
IndexSID	D/Req	Stream Identifier (SID) of Index Table	See MXF Format Specification – SMPTE ST 377-1
BodySID	Req	Stream Identifier (SID) of the indexed Essence Container	See MXF Format Specification – SMPTE ST 377-1
Slice Count	D/Req	Number of Slices minus 1 (NSL)	2
PosTableCount	Opt	Number of PosTable Entries minus 1 NPE	1
Delta Entry Array	Opt	Map Elements onto Slices	See Table B.7
Index Entry Array	D/Req	Index from Edit Unit number to stream offset	See Table B.8

The Delta Entry Array contains an entry for every indexed element in the Generic Container. The order of the elements in the Delta Entry Array matches the order of the elements in the Generic Container. The example below is a Delta Entry Array designed to match the example in Figure B.3. Actual implementations would construct a Delta Entry Array according to the properties of the actual Essence in the file.

In frame wrapping mode, the Element Delta values need to include the lengths of the "KL" for each element. The result is that each Element Delta points to the first byte of the key in the KLV which wraps an element. If the overall length of all elements in each frame is constant, then a Delta Entry Array and an "Edit Unit Byte Count" item are sufficient to define the Index Table Segment. In this example, we have several variable length elements so that an Index Entry Array is required.

The Delta Entry Array does not distinguish which element is which in the Index Table. To know which element is indexed, the following rules apply when an MPEG long GOP stream is indexed:

- Each Content Package starts with the same number and order of elements as the previous Content Package;
- If new elements are introduced for whatever reason, they need to be appended to the end of the existing Content Package Elements;
- If elements in the Content Package have no data, then an Index Entry for a 0 length VBE Element needs to be created;
- Index Tables need to have the same number of Delta Entries as the maximum number of elements in any Content Package;
- The data type of an Index Entry can be determined by inspecting the key which is pointed to by a **non-zero length** Index Entry.

Table B.7 – Frame Wrapped Delta Entry Array for Frame Wrapped VBE Example

	Field Name	Type	Meaning	Use
	NDE	UInt32	Number of Delta Entries	4
	Length	UInt32	Length of each Delta Entry	6
Data Delta Entry	PosTableIndex	Int8	0= No reordering +ve = PosTable Index	1 (1st PosTable Entry)
Data Delta Entry	Slice	UInt8	Slice number in IndexEntry	0
Data Delta Entry	Element Delta	UInt32	Delta from start of slice to this Element	0
Picture Delta Entry	PosTableIndex	Int8	-ve - reordered	-1 (reordered Long GOP content)
Picture Delta Entry	Slice	UInt8	Slice number in IndexEntry	0
Picture Delta Entry	Element Delta	UInt32	Delta from start of slice to this Element	sizeof(KL) + sizeof(Data)
Sound Delta Entry	PosTableIndex	Int8	0= No reordering +ve = PosTable Index	2 (2nd PosTable Entry)
Sound Delta Entry	Slice	UInt8	Slice number in IndexEntry	1
Sound Delta Entry	Element Delta	UInt32	Delta from start of slice to this Element	0
Fill Delta Entry	PosTableIndex	Int8	0= No reordering +ve = PosTable Index	0
Fill Delta Entry	Slice	UInt8	Slice number in IndexEntry	2
Fill Delta Entry	Element Delta	UInt32	Delta from start of slice to this Element	0

Table B.8 – Frame Wrapped Index Entry Array for Frame Wrapped VBE Example

	N	Field Name	Type	Meaning	Use
	1	NIE	UInt32	Number of Index Entries	Equal to number of frames
	1	Length	UInt32	Length of each Index Array Entry	Varies depending on the number of slices and Position offsets
One Index Entry for every frame	NIE	Temporal Offset	Int8	Offset in Edit Units from Display Order to Coded Order	Set according to picture type
		Key Offset	Int8	Offset in Edit Units to previous Key Frame. The value is zero if this is a Key Frame.	Set according to picture type
		Flags	EditUnitFlag	Flags for this Edit Unit	Set according to picture type
		Stream Offset	UInt64	Offset in bytes from the first KLV element in this Edit Unit within the Essence Container Stream	Offset from the first byte of the Key of the KLV for the first frame to the first byte of the Key of the KLV for the Data Element in this frame
		SliceOffset	NSL x UInt32	The offset in bytes from the Stream Offset to the start of this slice.	Optional depending on the complexity of the VBR items. In this case there are 3 slices and NSL is set to 2
		PosTable	NPE *Rational	The fractional position offset from the start of the Content Package to the synchronized sample in the Content Package	This ought to be calculated for each wrapped element to ensure precise synchronization is maintained. There are 2 elements requiring offsets in this example so NPE=2

Table B.8 shows the descriptions of the various elements required in the Index Entry. Table B.9 shows entries for the first 6 frames of a long GOP sequence. The following example values have been used in creating the table:

- The GOP display sequence for frames 0-5 is $B_0I_1B_2P_3B_4P_5$. This is the indexed order of the frames when looking up the temporal offset;
- The GOP transmission order for frames 0-5 is $I_1B_0P_3B_2P_5B_4$. This is the stored order of the frames and the stored order of the Index Table data;
- The GOP is closed (i.e. the first B frame contains predictions only from the I frame);
- All lengths include the length of the key and length fields (16 bytes + 4 bytes);
- The Data Element length is fixed at 700 bytes and temporally offset by -0.25 Edit Units;
- The I frames are 48000 bytes, P frames are 9000 bytes and B frames 1000 bytes;
- In the 6 Content Packages, there are 8 Sound Elements;
- The number of Sound Elements are multiplexed in the Content Packages as follows: (1)(1)(2)(1)(1)(2);
- Each Sound Elements is 1000 bytes;
- Each Fill Element is 300 bytes.

Table B.9 – Index Entry Samples for Frame Wrapped VBE Example

	N	Field Name	Type	Value	Note
		NIE	UInt32	6	6 Entries in this Index Table Segment
		Length	UInt32	35	Sizeof(IndexEntry) including the SliceOffsets & PosTable
	Index Entry[0] contains Index data for I_1, and a Temporal offset from B_0 to Index Entry[1]				
0	Index Entry[0] – "I_1"	Temporal Offset	Int8	1	B_0 index data is stored in Index Entry[1]
		Key Frame Offset	Int8	0	The Key frame is IndexEntry[0]
		Flags	EditUnitFlag	00h	I frame – no reference
		Stream Offset	UInt64	0	Offset of the 1st Stored Content Package in the Stream – I_1
		SliceOffset[0]	UInt32	48700	sizeof(data) + sizeof(I_1 frame)
		SliceOffset[1]	UInt32	49700	sizeof(data) + sizeof(I_1 frame) + sizeof(sound)
		PosTable[0]	Rational	-1/4	Temporal offset from start of data to start of video
		PosTable[1]	Rational	-1/3	Temporal offset from start of sound to start of video
	Index Entry[1] contains Index data for B_0, and a Temporal offset from I_1 to Index Entry[0]				
1	Index Entry[1] – "B_0"	Temporal Offset	Int8	-1	I_1 index data is stored in Index Entry[0]
		Key Frame Offset	Int8	1	The Key frame is IndexEntry[0] and 1-0=1
		Flags	EditUnitFlag	D0h	Closed GOP B frame – backward reference, sequence_header & random access
		Stream Offset	UInt64	50 000	Offset of the 2nd Stored Content Package in the Stream – B_0
		SliceOffset[0]	UInt32	1700	sizeof(data) + sizeof(B_0 frame)
		SliceOffset[1]	UInt32	2700	sizeof(data) + sizeof(B_0 frame) + sizeof(sound)
		PosTable[0]	Rational	0	Temporal offset from start of data to start of video
		PosTable[1]	Rational	0	Temporal offset from start of sound to start of video
	Index Entry[2] contains Index data for P_3, and a Temporal offset from B_2 to Index Entry[3]				
2	Index Entry[2] – "P_3"	Temporal Offset	Int8	1	B_2 index data is stored in Index Entry[3]
		Key Frame Offset	Int8	-2	The Key frame is IndexEntry[0] and 0-2= -2
		Flags	EditUnitFlag	40h	P frame – forward reference
		Stream Offset	UInt64	53 000	Offset of the 3rd Stored Content Package in the Stream – P_3
		SliceOffset[0]	UInt32	9700	sizeof(data) + sizeof(P_3 frame)
		SliceOffset[1]	UInt32	11700	sizeof(data) + sizeof(P_3 frame) + 2*sizeof(sound)
		PosTable[0]	Rational	-1/4	Temporal offset from start of data to start of video
		PosTable[1]	Rational	0	Temporal offset from start of sound to start of video
	Index Entry[3] contains Index data for B_2, and a Temporal offset from P_3 to Index Entry[2]				
3	Index Entry[3] – "B_2"	Temporal Offset	Int8	-1	P_3 index data is stored in Index Entry[2]
		Key Frame Offset	Int8	-3	The Key frame is IndexEntry[0] and 0-3= -3
		Flags	EditUnitFlag	30h	B frame – bidirectional reference
		Stream Offset	UInt64	65 000	Offset of the 4th Stored Content Package in the Stream – B_2
		SliceOffset[0]	UInt32	1700	sizeof(data) + sizeof(B_2 frame)
		SliceOffset[1]	UInt32	2700	sizeof(data) + sizeof(B_2 frame) + sizeof(sound)
		PosTable[0]	Rational	-1/4	Temporal offset from start of data to start of video
		PosTable[1]	Rational	-2/3	Temporal offset from start of sound to start of video

	N	Field Name	Type	Value	Note
	Index Entry[4] contains Index data for P_5, and a Temporal offset from B_4 to Index Entry[5]				
4	Index Entry[4] – "P_5"	Temporal Offset	Int8	1	B_4 index data is stored in Index Entry[5]
		Key Frame Offset	Int8	-4	The Key frame is IndexEntry[0] and 0-4= -4
		Flags	EditUnitFlag	40h	P frame – forward reference
		Stream Offset	UInt64	68 000	Offset of the 5th Stored Content Package in the Stream – P_5
		SliceOffset[0]	UInt32	9700	sizeof(data) + sizeof(P_5 frame)
		SliceOffset[1]	UInt32	10700	sizeof(data) + sizeof(P_5 frame) + sizeof(sound)
		PosTable[0]	Rational	-1/4	Temporal offset from start of data to start of video
		PosTable[1]	Rational	-2/3	Temporal offset from start of sound to start of video
	Index Entry[5] contains Index data for B_4, and a Temporal offset from P_5 to Index Entry[4]				
5	IE[5] – "B_4"	Temporal Offset	Int8	-1	P_5 index data is stored in Index Entry[4]
		Key Frame Offset	Int8	-5	The Key frame is IndexEntry[0] and 0-5= -5
		Flags	EditUnitFlag	30h	B frame – bidirectional reference
		Stream Offset	UInt64	79 000	Offset of the 6th Stored Content Package in the Stream – B_4
		SliceOffset[0]	UInt32	1700	sizeof(data) + sizeof(B_4 frame)
		SliceOffset[1]	UInt32	3700	sizeof(data) + sizeof(B_4 frame) + 2*sizeof(sound)
		PosTable[0]	Rational	-1/4	Temporal offset from start of data to start of video
		PosTable[1]	Rational	-1/3	Temporal offset from start of sound to start of video

B.5 Index Table Example for Clip Wrapped VBE MPEG Essence with Extra Items in the Generic Container

This is the most complex case where the Index Table points to the first byte of the payload of each element in the Generic Container. The other Generic Container elements are indexed by correct use of the Delta Entries and Index Entries. The Delta Entry values are limited to a UInt32 number range and will, therefore, be incapable of specifying the offset from the Picture Item of frame 1 to the Sound Item of frame 1.

To overcome this, the Index Table Segment is sliced, even if the offset to the Sound Item is constant so that there is, in effect, a 33-bit variable offset from the MPEG frame to the corresponding Sound Item (i.e., a 32-bit variable Slice Offset + 32-bit fixed Delta Entry of, for example, 2^{32}-1). In clip wrapping where an Index Table is required, but 33-bit offsets are insufficient, the clip wrapped Generic Container need to be split into several smaller Generic Containers, each with a single clip.

Table B.10 – Clip Wrapped Index Table Segment Set for Clip Wrapped VBE Example

Item Name	Req ?	Meaning	Use
Index Table Segment	Req	An Index Table Segment set	See MXF Format Specification – SMPTE ST 377-1
Length	Req	Set Length	See MXF Format Specification – SMPTE ST 377-1
Instance ID	Req	Unique ID of this instance	See MXF Format Specification – SMPTE ST 377-1
Index Edit Rate	Req	Edit Rate copied from the Tracks of the Essence Container	See MXF Format Specification – SMPTE ST 377-1
Index Start position	Req	The first Edit Unit indexed by this Index Table Segment measured in Top-Level File Package Edit Units relative to the first Edit Unit in the Essence Container	This defines the temporal start point for this Index Table Segment
Index Duration	Req	Time Duration of this Index Table Segment measured in Edit Units of the referenced Package	Combined with the Start Position, this allows an application to determine if this Index Table Segment spans a given Position value.
Edit Unit Byte Count	D/Req	Defines the byte count of each and every Edit Unit. A value of 0 defines the byte count of Edit Units is only given in the Index Entry Array	0 unless the total length of all the Generic Container Elements is of constant size. In this example we assume the Data Elements are VBR so the value is 0.
IndexSID	D/Req	Stream Identifier (SID) of Index Table	See MXF Format Specification – SMPTE ST 377-1
BodySID	Req	Stream Identifier (SID) of the indexed Essence Container	See MXF Format Specification – SMPTE ST 377-1
Slice Count	D/Req	Number of Slices minus 1 (NSL)	3 (one slice per element)
PosTableCount	Opt	Number of PosTable Entries minus 1 NPE	In this example, there is only a single Content Package and the audio and video start synchronously. Hence NPE=0
Delta Entry Array	Opt	Map Elements onto Slices	See Table B.11
Index Entry Array	D/Req	Index from Edit Unit number to stream offset	See Table B.12

The Delta Entry Array needs to contain an entry for every indexed Element in the Generic Container. The order of the Elements in the Delta Entry array need to match the order of the Elements in the Generic Container. Table B.11 below gives an example for a Delta Entry Array based on Figure 6. Actual implementations would construct a Delta Entry Array according to the properties of the actual Essence in the file.

In clip wrapping mode, the Element Delta values do not include the lengths of the "KL" for each Element. The result is that each Element Delta points to the first byte of the value in the KLV which wraps an Element.

Table B.11 – Clip Wrapped Delta Entry Array for Clip Wrapped VBE Example

	Field Name	Type	Meaning	Use
	NDE	UInt32	Number of Delta Entries	4
	Length	UInt32	Length of each Delta Entry	6
Picture Delta Entry	PosTableIndex	Int8	-ve - reordered	-1
	Slice	UInt8	Slice number in IndexEntry	0
	Element Delta	UInt32	Delta from start of slice to this Element	0
Sound Delta Entry	PosTableIndex	Int8	0= No reordering	0
	Slice	UInt8	Slice number in IndexEntry	1
	Element Delta	UInt32	Delta from start of slice to this Element	0
Sound Delta Entry	PosTableIndex	Int8	0= No reordering	0
	Slice	UInt8	Slice number in IndexEntry	2
	Element Delta	UInt32	Delta from start of slice to this Element	0
Data Delta Entry	PosTableIndex	Int8	0= No reordering	0
	Slice	UInt8	Slice number in IndexEntry	3
	Element Delta	UInt32	Delta from start of slice to this Element	0

Table B.12 – Clip Wrapped Index Entry Array for Clip Wrapped VBE Example

	N	Field Name	Type	Meaning	Use
	1	NIE	UInt32	Number of Index Entries	Equal to number of frames
	1	Length	UInt32	Length of each index array entry	23 in this example
One Index Entry for every frame	NIE	Temporal Offset	Int8	Offset in Edit Units from Display Order to Coded Order	Set according to picture type
		Key Frame Offset	Int8	Offset in Edit Units to previous Key Frame. The value is zero if this is a Key frame.	Set according to picture type
		Flags	EditUnitFlag	Flags for this Edit Unit	Set according to picture type
		Stream Offset	UInt64	Offset in bytes from the first KLV element in this Edit Unit within the Essence Container Stream	Offset from the first byte of the value of the MPEG KLV for the first frame to the first byte of the stored data for this MPEG frame in Figure 6
		SliceOffset	UInt32	The offset in bytes from the Stream Offset to the start of slice 1	Offset from the first byte of the stored data for this MPEG frame to the first byte of the corresponding audio sample in the first Sound Element in Figure 6
		SliceOffset	UInt32	The offset in bytes from the Stream Offset to the start of slice 2	Offset from the first byte of the stored data for this MPEG frame to the first byte of the corresponding audio sample in the second Sound Element in Figure 6

N	Field Name	Type	Meaning	Use
	SliceOffset	UInt32	The offset in bytes from the Stream Offset to the start of slice 3	Offset from the first byte of the stored data for this MPEG frame to the first byte of the corresponding audio sample in the Data Element in Figure 6
	PosTable	NPE *Rational	The fractional position offset from the start of the Content Package to the synchronized sample in the Content Package	In this example, there is only a single Content Package and the audio and video start synchronously. NPE is zero so there is no data here

B.6 Indexing of Fill Items

Indexing of fill is performed in the same way as any other variable length Element (whose length can be zero). Inspection of the Element key when the Index Entry is non-zero will reveal a fill key.

Annex C Identifying MPEG Picture Type and Finding MPEG Content on the Timeline
(Informative)

This section is intended to explain a mechanism by which the picture type of an MPEG picture can be identified. There are two relevant cases: in-stream for frame wrapping, and Index Table based for random access.

C.1 Identification of MPEG Picture Type

C.1.1 Identifying MPEG Picture Type Using Frame Wrapping Without an Index Table

In this document, frame wrapping of an MPEG AU is described. The value of the KLV triplet will be the start of an AU. For a video ES, this will be a sequence_start_code, a GOP_start_code, or a picture_start_code. There is a simple algorithm which will allow the picture type to be determined as described in Table C.1.

Table C.1 – MPEG picture type determination

Sequence header	This needs to be skipped, along with its sequence_header_extensions and look for picture_start_code
Group of Pictures Header	This needs to be skipped (59 bits) and look for picture_start_code
Picture header	This contains the picture_coding_type (3 bits offset 42 bits from start of header)

C.1.2 Identifying MPEG Picture Type Using an Index Table

When MPEG Essence is in an external file, or random access to long GOP material is required, it is often useful to be able to know the MPEG picture type so that sufficient frames can be fetched to ensure an MPEG key frame and all the key frames are available for decoding any frame within the MPEG GOP sequence. The Index Tables provide enough information for this to be calculated. Picture types are easily identified from bit 1 and bit 0 of the Flags Property which is described in Table 13 of Section 11 within this document.