

SMPTE STANDARD

Material Exchange Format — Mapping AES3 and Broadcast Wave Audio into the MXF Generic Container



Page 1 of 41 pages

Table of Contents	Page
Foreword	3
1 Scope	4
2 Conformance Notation	4
3 Normative References	4
4 Glossary of Acronyms, Terms and Data Types	5
5 Introduction (Informative)	6
6 Packing of Audio Samples	6
6.1 Sample Validity	7
6.2 Division of Audio Essence into Frame-Wrapped Content.....	7
6.3 Division of Audio Essence into Clip-Wrapped Content Packages.....	8
6.4 Custom Wrapped Audio Elements.....	9
6.5 KLV Coding of Digital Audio Data Sound Elements	9
6.5.1 Wave sound element key.....	9
6.5.2 Essence element count — Byte 14.....	10
6.5.3 Essence element type — Byte 15.....	10
6.5.4 Essence element number — Byte 16	10
6.5.5 Wave sound element length	10
6.5.6 Wave sound element value.....	10
6.6 Use of KAG	11
6.7 Optional Mapping of Material Package Tracks to Audio Channel IDs.....	11
7 Carriage of BWF Metadata Chunks	11
7.1 General	11
7.2 <fmt> Chunk.....	11
7.2.1 Sound essence compression.....	13
7.3 Broadcast Extension <bext> Chunk.....	13
7.4 Other Chunks	15
7.4.1 Peak envelope <levl> chunk	15

7.4.2	Quality <qlty> chunk	15
7.4.3	Other chunks	15
7.4.4	Unknown chunks while MXF encoding	16
7.4.5	Use of a physical descriptor (Informative)	16
8	Carriage of SMPTE 339M Time Stamp, V-Sync and User Data	16
9	Carriage of AES3 Channel Status Data and User Data	17
9.1	General	17
9.2	Block Start Offset	17
9.3	Channel Status Data	17
9.3.1	Definition	17
9.3.2	Channel status data — Minimum implementation	17
9.3.3	Channel status data — Standard implementation	17
9.3.4	Channel status data — Enhanced implementation	17
9.3.5	Optional mapping of channel status data into MXF	18
9.4	User Data	19
10	SMPTE Label for Essence Container Identification	19
Annex A	Definition of the Essence Descriptor Sets for the Essence Data Defined by this Standard (Normative)	20
A.1	Wave Audio Essence Descriptor	20
A.2	AES3 Audio Essence Descriptor	22
A.3	Wave Audio Physical Descriptor	23
A.4	Unknown Chunk Set	24
Annex B	Example of Mapping Channels and Descriptors (Informative)	25
Annex C	Comparison with Other Standards (Informative)	26
C.1	Comparison with SMPTE 302M and SMPTE 331M	26
C.2	Comparison with SMPTE 337M / SMPTE 338M / SMPTE 339M and Related Standards	26
Annex D	Terminology Comparison with AES31-2 (Normative)	27
D.1	Comparison of Data Type Definitions	27
D.2	Comparison of <fmt> Chunk Fields	27
D.3	Comparison of <bext> Chunk Fields	28
Annex E	Simple Audio Channel Mapping (Normative)	30
E.1	Optional Mapping of Subchannels of Audio Elements to Material Package	30
E.2	Optional Combination of Source Package Tracks	30
Annex F	Mapping Multiple SMPTE 339M Time Stamps (Normative)	32
F.1	Association of SMPTE 337M data_dstream_number to Time Code Tracks	32
Annex G	Peak Envelope Chunk (Normative)	33
G.1	Generation of the Peak Envelope Data	33
G.2	Definition of the Peak Envelope Chunk	33

Annex H Quality Chunk (Normative).....	36
H.1 Syntax of Quality Chunk.....	36
H.2 Definition of Quality Chunk.....	36
H.3 Example of Quality Chunk.....	39
Annex I Bibliography (Informative).....	41

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 382M was prepared by Technology Committee W25.

1 Scope

This standard defines the mapping of digital audio data, ancillary data and metadata from the broadcast wave format (BWF) and from AES3 digital audio data into sound essence elements, which can be included in an MXF generic container. The data carried in wave audio essence elements may be uncompressed PCM audio data, compressed data or raw data as in BWF, AES3 or SMPTE 337M carried in a single AES3 stream. Labels for the identification of uncompressed audio samples within this mapping are provided by this standard. Labels for compressed audio types can be added to the Labels Register defined by SMPTE 400M.

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.

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

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

SMPTE 337M-2000, Television — Format for Non-PCM Audio and Data in an AES3 Serial Digital Audio Interface

SMPTE 338M-2000, Television — Format for Non-PCM Audio and Data in AES3 — Data Types

SMPTE 339M-2004, Television — Format for Non-PCM Audio and Data in AES3 — Generic Data Types

SMPTE 377M-2004, Television — Material Exchange Format (MXF) — File Format Specification

SMPTE 379M-2004, Television — Material Exchange Format (MXF) — MXF Generic Container

SMPTE 400M-2004, Television — SMPTE Labels Structure

AES3-2003, Serial Transmission Format for Two-Channel Linearly Represented Digital Audio Data

AES31-2-2006, AES Standard on Network and File Transfer of Audio — Audio-File Transfer and Exchange — File Format for Transferring Digital Audio Data Between Systems of Different Type and Manufacture

ITU-R BR.1352-2:2002, Broadcast Wave Format (BWF), Annex 1, Annex 1 Appendix 1 and 2, and Annex 3

4 Glossary of Acronyms, Terms and Data Types

Explanations and definitions of the acronyms and terms can be found in the MXF file format specification document. A supplementary glossary of acronyms and terms is defined in SMPTE 379M. They are not repeated here to avoid any divergence of meaning.

The following acronyms are used in this standard:

- AES Audio Engineering Society
- BEXT The Broadcast Extension Chunk
- BWF¹ Broadcast Wave Format
- BWFF Broadcast Wave Format file as defined in AES31-2
- CRCC Cyclic Redundancy Check Code
- LEVL The Peak Envelope (level) RIFF Chunk defined by Annex G
- PCM Pulse Code Modulation
- RIFF Resource Interchange File Format — a generic file format consisting of a series of chunks, each composed of a 32-bit chunk identifier, a 32-bit length field, and a number of data bytes.
- QLTY The Capturing Report (quality) RIFF Chunk defined by Annex H

The following additional terms apply to AES3. Refer to AES3 for the precise definition.

Subframe: fixed structure used to carry the information described in 4.1.1 and 4.1.2 of AES3.

Frame: sequence of two successive and associated subframes as described in 4.1.2 of AES3.

Block: a group of 192 consecutive frames. The start of a block is designated by a special subframe preamble.

¹ The abbreviations BWFF and BWF co-exist in this standard due to the fact that references cited by this standard were developed at different times. The abbreviations have the same technical meaning.

The following additional data types apply to this standard:

char[] A data type comprising either a single ISO 7-bit character or a string of ISO 7-bit characters with the string length determined by the number in the brackets. Where brackets are specified as simply “[]”, the string length is variable and terminated by a zero value.

In some geographic areas, strings of type **char[]** may contain UTF-8 or Shift-JIS characters.

5 Introduction (Informative)

This standard defines the mapping of digital audio data and ancillary data and metadata from broadcast wave format (BWF) files, and AES3 data streams into KLV-encoded sets for inclusion in MXF files.

In common with other body specifications, the principal mapping is of the essence, in this case samples of digital audio data, into KLV packets. These are called wave audio essence elements.

This standard employs the same bit-by-bit packing of sample data as is used by BWF files. The sample data is packed on a frame-by-frame or clip-by-clip or 5-frame basis excluding all ancillary data. This is described in section 6.

This standard defines wave audio essence elements which may be used as MXF essence containers in their own right or as elements in content packages within an MXF generic container, depending upon application requirements. The division of audio essence into elements is discussed in section 6.2.

Data carried in wave audio essence elements may be uncompressed PCM audio data or compressed data, as defined in AES31-2, AES3 or SMPTE 337M carried in a single AES3 stream. The specific format is indicated by the Sound Essence Coding property, which is described in section 7.2.1.

BWF defines metadata which applies to the entire clip. This standard maps this metadata into the MXF header metadata. This is described in section 7.

This standard also describes the mapping of constant and time-varying ancillary data and metadata from AES3 subframes and U and C bits into appropriate MXF data structures. This is described in section 9.

NOTE – Data can be lost if applications do not support the carriage of the broadcast wave metadata or AES metadata parameters that are not mapped to the MXF structural metadata. It is recommended that implementers and users ensure that descriptive metadata mappings or other mechanisms are available in applications which require this level of data integrity. This particularly applies to parameters such as unmapped BWFF chunk parameters as described in section 7.3.

6 Packaging of Audio Samples

This standard defines audio essence elements containing one, two or N channels.

In a multiple-channel element, samples are interleaved in order of channel number. The top-level file package shall have one track per mapped wave audio essence stream.

A material package may have multiple tracks of type sound. Source Clips in the material package may reference a given channel or channels within the top-level file package by setting an optional Channel ID property to the channel number in the target track.

When AES3 subframes are being mapped, the first subframe of a block (i.e., that identified by a Z preamble) is first in every sample pair.

All ancillary data, including the VUCP bits, are mapped to a wave audio essence element as part of the packing of audio samples. Section 6.1 considers the validity bit, while section 9 describes the handling of the user, channel status and parity data.

Each sample shall be carried in the smallest number of bytes necessary to completely contain the data, stored as a little-endian integer (least significant byte first). Data shall be aligned to the most significant bit, and unused bits shall be set to zero.

For example, 20 bit samples are carried in 3 bytes. The least significant 4 bits of the first (least significant) byte are set to zero. A decoder can determine the number of bytes from the essence descriptor using the formula given in section 7.2.

NOTE – AES31-2 provides additional details of the mapping of audio samples into data bytes. SMPTE 320M and SMPTE 323M provide definitions of commonly used assignments of multiple channels. ITU-R BR.1352 also has details of the mapping of audio samples into data bytes.

MXF encoders should set the optional Channel Assignment property of the Wave Audio Essence Descriptor (see Table A.1) to the appropriate Label from the SMPTE Labels Register (i.e., the register defined by SMPTE 400M) in order to describe specific channel assignments.

This standard does not define the format of MXF essence containers or essence descriptors for big-endian sound essence.

6.1 Sample Validity

When formatted according to AES3, audio subframes include a channel validity (V) bit, which is set to one when the sample data is not suitable for conversion to an analog audio signal. When the validity bit is not present, applications shall process audio data as though the validity bit were set to zero.

This data is not available in the packed wave audio essence elements data format, and all applications shall process audio data as though the validity bit were set to zero.

NOTE – SMPTE 314M section 4.6.2.1.3 describes an audio error code in DV-derived audio sample data. This error code is the most-negative value of the sample (8000h), and is used to indicate invalid samples. It is recommended that these samples be overwritten with valid data in the data stream before packing into wave audio essence elements. This applies only to DV-derived audio; for AES3-derived audio, 8000h is valid sample data. DV-derived audio sample data can be determined from the Sound Essence Coding parameter which can be extended using the Labels Register (defined by SMPTE 400M) to include DV-derived audio types.

6.2 Division of Audio Essence into Frame-Wrapped Content

Audio essence may be divided into elements of approximately equal duration, for synchronizing with video essence within content packages of an MXF generic container. “Approximately equal duration” is defined as a constant number of samples ± 5 samples.

The division into elements can take a variety of forms which are driven by various system design issues. This section describes four example divisions drawn from current practice:

- Audio elements each as long as the corresponding picture frame.
- Audio elements where the duration of the elements in each content package matches the duration of the synchronized video within that content package.
- Audio elements where each has the same duration as some fixed number of video frames. Generally, the duration (or number of frames) is chosen to reduce or eliminate the variation in the size of each group, or is chosen to be convenient for buffering considerations.
- Audio elements divided into content packages where the duration of those elements is convenient for buffering, such as one or two seconds.

If the audio is locked to video and if the audio sample rate (i.e., the sampling rate for linear audio or the rate of audio access units for compressed audio) is an integer multiple of the video frame rate, then the audio element shall have the same size in each content package.

Example: In a multiplexed, frame-wrapped essence container that carries video with a frame rate of 25 frames per second, and locked linear audio with a sample rate of 48 kHz, each audio element contains exactly 1920 samples.

If the audio sample rate is not an integer multiple of the video frame rate, then the number of samples in each element shall vary such that a correct aggregate rate is maintained.

Example: In a multiplexed, frame-wrapped essence container that carries video with a frame rate of $30 \times 1000 / 1001$ (≈ 29.97) frames per second, and locked linear audio with a sample rate of 48 kHz, the aggregate ratio of audio samples per video frame is about 1601.6 samples per video frame. This can be approximated with a 5-frame pattern, for example 1602, 1601, 1602, 1601, 1602, or 1600, 1602, 1602, 1602, 1602. However, other sequences are also possible.

The ordinal number of the first frame of audio essence within a five-frame sequence may optionally be indicated in the Wave Audio Essence Descriptor :: Sequence Offset property (see annex A.1), to provide guidance in processing.

NOTE 1 – In the case where the audio sample rate is not an integer multiple of the video frame rate, then there will be a variable number of samples per video frame which could require the audio samples to be indexed as variable rate elements. This can be avoided by adding fill KLV items to ensure constant bytes per element.

In the case of there being no interleaved picture content, then one frame, for the purposes of frame wrapping, shall be equivalent to the definition of the edit unit in the top-level file package track, which is linked to the sound element keys via the mechanism defined in SMPTE 379M.

NOTE 2 – It is recommended, however, that clip wrapping be used when there is no interleaved picture content.

The number of samples in each content package is calculated from the length field of the surrounding KLV packet, divided by the value of the Block Align property of the Wave Audio Essence Descriptor.

Two content packages, each containing two sound elements, are shown in Figure 1.

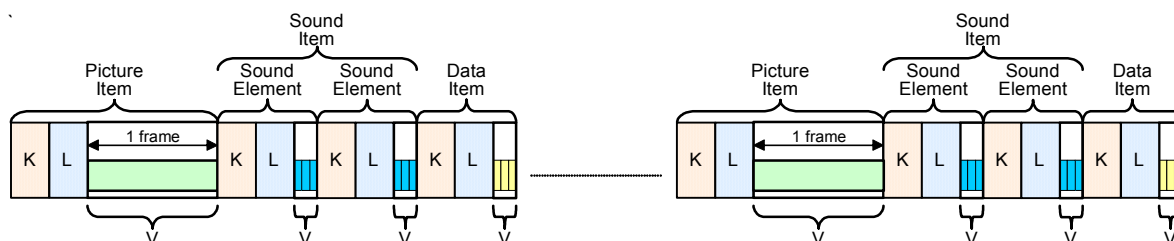


Figure 1 – Representation of frame wrapping with other elements present

The generic container key for the sound item is given in section 6.5.1.

6.3 Division of Audio Essence into Clip-Wrapped Content Packages

An alternative to frame wrapping is to encode all the audio data in a single element. This is defined as clip-wrapped mapping in SMPTE 379M.

This wrapping may also be useful in sound-only applications.

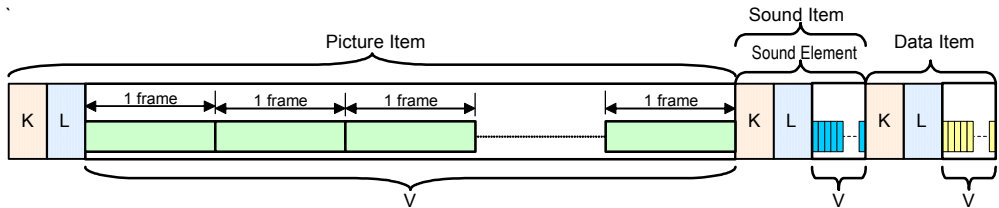


Figure 2 – Simple representation of clip wrapping with other elements

6.4 Custom-Wrapped Audio Elements

When audio elements are created that are neither frame wrapped nor clip wrapped, the file shall use KLV Key values that indicate custom wrapping.

When the audio wrapping is slaved to some external criteria where the “approximately constant” number of samples per element cannot be guaranteed, the file shall use key values that indicate custom wrapping.

Example: The case where a logical pattern of video frames occurs. When interleaving audio with MPEG-2 video, the size of the GOP (group of pictures) need not be constant, and it may be desirable for the arrangement of audio elements within a content package to correspond to each GOP. For example, Figure 3 shows the case where the first GOP has a single frame, while the second GOP has only three frames. The length of audio elements could be chosen in lengths approximately equal to these logical frame groups.

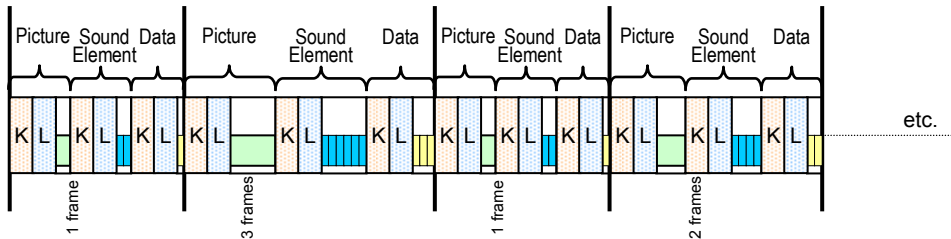


Figure 3 – Simple representation of custom wrapping with other elements

6.5 KLV Coding of Digital Audio Data Sound Elements

6.5.1 Wave sound element key

The values of the first 12 bytes of the essence element key are defined in SMPTE 379M, MXF generic container format. The values of the last four bytes of the essence element key are given in Table 1.

Table 1 – Key value for the wave sound element

Byte No.	Description	Value (hex)	Meaning
13	Item Type Identifier	16h	GC Sound Item
14	Essence Element Count	kkh	Count of Sound Elements in this Generic Container
15	Essence Element Type	01h 02h 03h 04h 0Bh 0Ch	Wave Frame-Wrapped Element Wave Clip-Wrapped Element AES Frame-Wrapped Element AES Clip-Wrapped Element Wave Custom-Wrapped Element AES Custom-Wrapped Element
16	Essence Element Number		The number (used as an index) of this Sound Element in this Generic Container

6.5.2 Essence element count — Byte 14

This is a count of the number of elements in the sound items in the generic container.

6.5.3 Essence element type — Byte 15

The values are given in table 1.

6.5.4 Essence element number — Byte 16

This is a number used as an index to identify this instance of the element type within the item. Each element within an item shall have a unique value between 0 and 7Fh as defined by SMPTE 379M, which shall remain constant throughout any instance of a generic container.

6.5.5 Wave sound element length

For frame wrapping, the length field of the KLV-coded element shall be 4 bytes BER long-form encoded (i.e., 83h.xx.yy.zz).

For clip wrapping, the length field of the KLV-coded element shall be 8 bytes BER long-form encoded (i.e., 87h.aa.bb.cc.dd.ee.ff.gg).

For custom wrapping, the length field of the KLV-coded element shall be any value within the range defined by SMPTE 377M.

NOTE – SMPTE 377M requires MXF decoders to support BER long-form encodings with other byte lengths.

6.5.6 Wave sound element value

In a clip-wrapped file, the element value shall be the complete wave audio data.

In a frame-wrapped file containing uncompressed audio samples, the start position of the first sample of wave audio data shall be the same as the start position of the synchronized picture frame.

When equal start position values cannot be achieved because of complex clocking relationships or because of audio encoding schemes that encode multiple audio samples in non-separable blocks of bits (i.e., audio frames), the start position of the video access unit should fall within the duration of the first sample or audio frame of the sound element.

NOTE – More information on mapping to achieve synchronization is given in SMPTE EG 41, the MXF engineering guideline.

6.6 Use of KAG

There are no specific KAG requirements for the wave mapping. MXF encoders and decoders shall comply with the KAG rules in the MXF format document and other SMPTE specifications applicable to the Essence Container and Operational Pattern.

6.7 Optional Mapping of Material Package Tracks to Audio Channel IDs

Annex E provides a simple mechanism for mapping one or two channels of audio from a Source Package. It provides an optional mechanism for an MXF decoder to “play” a subset of stored audio channels. A complete N to M audio channel mapping mechanism is outside the scope of this specification.

7 Carriage of BWF Metadata Chunks

This section applies only to implementations where BWF metadata is available.

7.1 General

BWF defines metadata which applies to the entire clip. This standard maps this metadata into the MXF header metadata in properties of the Wave Audio Essence Descriptor (see Annex A).

In BWF, clip-oriented metadata is usually stored at the head of the file in RIFF data structures (RIFF is defined in AES31-2). Defined chunks include the <fmt> chunk and the broadcast wave extension <bext> chunk. Other RIFF chunks are allowed.

A minimum implementation shall map all the RIFF chunks that are required by the AES31-2 specification.

A transparent implementation shall map all the RIFF chunks in a BWF file.

NOTE – Some metadata chunks can also be stored after the audio data. According to the normative references, the only chunk that is required to appear before the audio data is the <fmt> chunk.

7.2 <fmt> Chunk

The <fmt> chunk as defined by BWFF includes several parameters describing the audio essence, as follows:

```
typedef struct {
    UInt16 wFormatTag;
    UInt16 nChannels;
    UInt32 nSamplesPerSec;
    UInt32 nAvgBytesPerSec;
    UInt16 nBlockAlign;
    UInt16 nBitsPerSample;
} fmt-ck;
```

The precise definitions of these parameters and other coding rules are given in AES-31-2-2006 Annex A, and readers should refer to that standard. These are also listed in Annex D.

wFormatTag	Defines the audio waveform type of the audio stream.
nChannels	Specifies the number of channels in the audio stream, 1 for mono, 2 for stereo.
nSamplesPerSec	Specifies the frequency of the sample rate of the audio stream in samples/second (Hz). Examples are 48000, 44100, and 96000.

nAvgBytesPerSec Specifies the average data rate. Playback processors can estimate the buffer size by using this value.

For VBR audio, Index Tables must be used to further refine any value in Average Bytes Per Second.

nBlockAlign Specifies the block alignment of the data, in bytes. Playback processors must process a multiple of **nBlockAlign** bytes of data at a time, so that the value of **nBlockAlign** can be used for buffer alignment. Examples are: 2 (for 16-bit mono), 6 (for 20- and 24-bit dual channel).

For uncompressed PCM:

$$BlockAlign = ChannelCount \times \text{floor}\left(\frac{QuantizationBits + 7}{8}\right)$$

and the reverse is true:

$$ChannelCount = \frac{BlockAlign}{\text{floor}\left(\frac{QuantizationBits + 7}{8}\right)}$$

For compressed signals, the equations above are not valid. ChannelCount and BlockAlign are independent. ChannelCount must be specified equal to the number of channels in the decompressed signal.

For 337M, the number of AES channels (aka AES subframes) present in the essence data is

$$SubframeCount = \frac{BlockAlign}{\text{floor}\left(\frac{QuantizationBits + 7}{8}\right)}$$

Applications shall use this formula - there is no need to carry this as a separate property in the stream.

nBitsPerSample Specifies the number of bits per sample per channel data. Each channel is assumed to have the same sample resolution. If this field is not needed, it shall be set to zero when mapping it to an MXF property.

For MXF, the parameters below are carried as individual items in the Wave Audio Essence Descriptor set. The correspondence is as follows:

Table 2 – <fmt> Chunk to MXF mapping

Field in <fmt> chunk	MXF Set :: Property
wFormatTag	Wave Audio Essence Descriptor :: Sound Essence Coding
nChannels	Wave Audio Essence Descriptor :: Channel Count
nSamplesPerSec	Wave Audio Essence Descriptor :: Audio Sampling Rate
nAvgBytesPerSec	Wave Audio Essence Descriptor :: Average Bytes Per Second
nBlockAlign	Wave Audio Essence Descriptor :: Block Align
nBitsPerSample	Wave Audio Essence Descriptor :: Quantization Bits

The Wave Audio Essence Descriptor :: Sample Rate property is defined by 377M in the File Descriptor. Its value is most commonly set to the frame rate of the video when in an interleave of video and audio. When in an audio-only file, it should be set to the same value as Wave Audio Essence Descriptor :: Audio Sampling Rate, which is defined by 377M in the generic Sound Essence Descriptor.

Any <format-specific-fields> that are present in a BWF file but are not specified in AES31-2 are not carried in MXF.

7.2.1 Sound Essence Compression

The Sound Essence Coding property of the Wave Audio Essence Descriptor in MXF is a 16-byte SMPTE UL. The equivalent field in the <fmt> chunk is wFormatTag.

When the essence container element contains uncompressed (PCM) audio that employs the same bit-by-bit packing of sample data as is used by AES31-2, this optional property should be absent from the descriptor set.

For all other sound essence compressions (e.g., MPEG 1 level 1, DV or other coding formats specified in AES31-2, SMPTE 338M) an appropriate value from the Labels Register (defined by SMPTE 400M) shall be added.

If the Sound Essence Coding property is not present in the Descriptor, MXF decoders shall use the SMPTE RP 224 Item Designator value of 04.02.02.01.01.00.00.00.

NOTE – This corresponds to the case of uncompressed audio essence that employs the same bit-by-bit packing of sample data as is used by AES31-2.

7.3 Broadcast Extension <bext> Chunk

The <bext> chunk defined by AES31-2 carries the following metadata:

```
typedef struct {
    char Description[256];
    char Originator[32];
    char OriginatorReference[32];
    char OriginationDate[10];
    char OriginationTime[8];
    UInt32 TimeReferenceLow;
    UInt32 TimeReferenceHigh;
    UInt16 Version;
    UInt8 UMID[64];
    char Reserved[190];
    char CodingHistory[];
} bext-ck;
```

The precise definitions of these parameters and other coding rules are given in AES31-2, and readers should refer to that standard. These definitions are also listed in Annex D.

Description	is a free text description of the sound sequence. BWF defines the coding as an ISO 7-bit character string.
Originator	is the name of the originator as an ISO 7-bit character string.
OriginatorReference	is an unambiguous reference to the material provided by the originator as an ISO 7-bit character string. See also EBU R99.
OriginationDate	is the date of creation specified as an ISO 7-bit character string: "yyyy-mm-dd".