EBU – Recommendation R133

Transport of subtitles using MXF in an IT-Based Television Production Environment

Source: SP-HIPS

Status: Recommendation

Geneva
May 2012

Page intentionally left blank. This document is paginated for two sided printing

Conformance Notation

This document contains both normative text and informative text.

All text is normative except for that in the Introduction, any section explicitly labelled as 'Informative' or individual paragraphs which start with 'Note:'.

Normative text describes indispensable or mandatory elements. It contains the conformance keywords 'shall', 'should' or 'may', defined as follows:

'Shall' and 'shall not':	Indicate requirements to be followed strictly and from which no deviation is permitted in order to conform to the document.
'Should' and 'should not':	Indicate that, among several possibilities, one is recommended as particularly suitable, without mentioning or excluding others. OR indicate that a certain course of action is preferred but not necessarily required. OR indicate that (in the negative form) a certain possibility or course of action is deprecated but not prohibited.
'May' and 'need not':	Indicate a course of action permissible within the limits of the document.

Default identifies mandatory (in phrases containing "shall") or recommended (in phrases containing "should") presets that can, optionally, be overwritten by user action or supplemented with other options in advanced applications. Mandatory defaults must be supported. The support of recommended defaults is preferred, but not necessarily required.

Informative text is potentially helpful to the user, but it is not indispensable and it does not affect the normative text. Informative text does not contain any conformance keywords.

A conformant implementation is one which 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.

Page intentionally left blank. This document is paginated for two sided printing

Contents

Page intentionally left blank. This document is paginated for two sided printing

Transport of Subtitles using MXF in an IT-Based Television Production Environment

EBU Committee	*First Issued*	*Revised*	*Re-issued*
TC	2012		

Keywords: MXF, subtitles, data transport, IT-based production.

1. Scope

This recommendation defines the transport of subtitles[1] in IT-based television production environments. Since MXF (Material Exchange Format) is the de facto standard when exchanging audio and video essence in file-based production workflows, this recommendation focuses on the use of MXF and subtitles. The entire life time cycle of subtitles – from generation to playout and archive – is considered.

The EBU has collected user requirements and has been investigating implementations that are already in use today. Harmonisation is needed to obtain predictable and reliable results when interchanging MXF files and when integrating new products into TV production environments. This is especially important for the creation of subtitles, which is often taking place in external production houses or at home working places on using a big variety of different platforms and applications. Hence, reliable exchange mechanisms are needed.

The introduction of HDTV has created new expectations such as displaying subtitles in different and more user-friendly ways. This recommendation takes this into account as well.

The document covers both the option to transport audio-visual essence and subtitles as two or more separated files (MXF external) and the option to transport all data in one common MXF file (MXF internal) for different user cases.

The first part of this document describes the different options when following MXF internal and MXF external approaches. Based on user requirements, the second part of this document recommends the use of some options and discourages others. Due to the great variety of different user requirements it is not the goal of this document to recommend one single option only.

[1] For simplicity, only the term "subtitles" is used in this document. The term should be interpreted as referring both to subtitles and to captions, equally.

2. Introduction

In the past subtitles have always been created, edited, exchanged and archived in many different ways. Sometimes subtitles have been closely linked to the video, in the extreme as burned-in subtitles in the video, sometimes they have been loosely coupled to the audio/video essence and have been stored on external storage media.

This is not different today; there are requirements and corresponding use cases where the audio/video essence and the subtitles are closely linked together and are stored in one common file. On the other hand there are different use cases and workflows where subtitles are handled separately and as a consequence, are not stored in the same file as the audio/video essence.

3. Definition of terms

- If the subtitles are stored in the same MXF file(s) as the audio/video essence, this is defined as "MXF internal".
- If the subtitles are stored in a separate file (besides the MXF file(s)), this is defined as "MXF external".

In the foreseeable future both methods will coexist in daily business.

- The term "captions" describes the use of subtitling for the deaf or hard of hearing. Captions include indications of the speakers and relevant sound effects.
- The term "subtitles" describes the use of subtitles for the use of translating purposes.
- The term "closed" describes optionally displayed captions/subtitles.
- The term "open" describes captions/subtitles that are "burnt-in" the video material.
- The term "Subtitle Instance" describes the collection of all subtitle fragments and related subtitle specific information (typically stored in the Subtitle Instance header) that "belong" to a programme. Every language/purpose/version combination shall be stored in a separated Subtitle Instance.

4. Current situation

The current specifications of MXF and their normative references (see Bibliography) provide for a number of different ways of carrying subtitle information inside or outside MXF files.

When following the MXF internal approach, six different options are available to enable the storage of subtitles inside one MXF file:

A.1	Subtitles as an EBU-TT instance inside a Generic Stream Partition
A.2	Subtitles as an STL instance inside a Generic Stream Partition
B.1	Subtitles in ANC packets inside Generic Container data elements
B.2	Subtitles in VBI lines inside Generic Container data elements
C.1	Subtitles in VBI lines inside Generic Container picture elements
C.2	Subtitles in dedicated areas of the compressed bitstream inside Generic Container picture elements

When following the MXF external approach, subtitles are stored in binary or XML-based file formats, such as:

D	Subtitles inside an external STL file (EBU Subtitling data exchange format – EBU Tech 3264)
E	Subtitles inside an external EBU-TT file
F	Subtitles inside other (proprietary) file formats

Figure 1 provides an overview of these possible options.

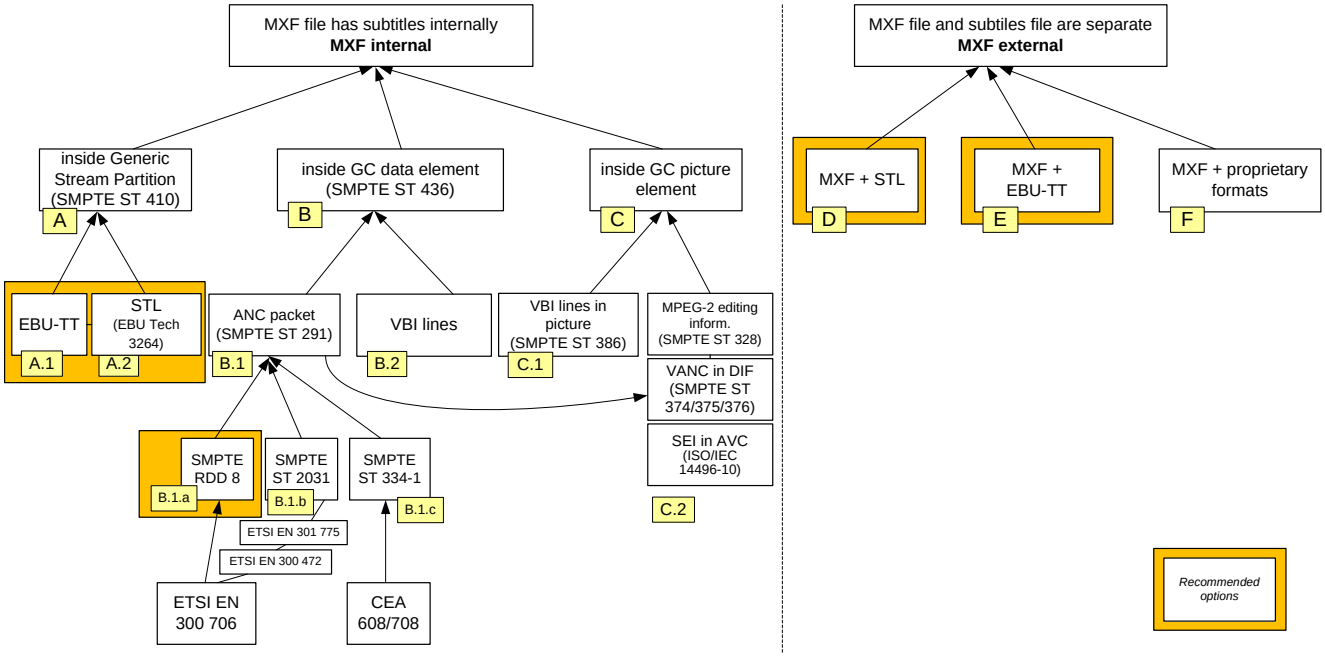


Figure 1: Overview of MXF internal and MXF external options

5. Recommendations

5.1 Recommendations for MXF internal usage

The EBU recommends the use of EBU-TT inside MXF Generic Stream Partitions (option A.1). This allows the use of a versatile, extensible XML-based subtitle format.

The EBU recommends the use of ANC packets in MXF Generic Container data elements according to SMPTE RDD-8 (option B.1.a).

Note: It would have been desirable to recommend a single option. However, existing implementations in the market have shown that both options are suitable for production use. Table 1 provides an overview of the advantages and disadvantages of each option.

A — Inside Generic Stream Partitions

Mapping of caption and subtitle data into the Generic Stream Partition according to SMPTE ST 410M:

A.1 — EBU-TT in MXF Generic Stream Partition

EBU-TT instances are mapped inside the Generic Stream Partition according to SMPTE ST 410. One Generic Stream Partition shall include one Subtitle Instance.

A.2 — STL in MXF Generic Stream Partition

Binary coded STL-files generated according to EBU Tech 3264 are mapped inside the Generic Stream Partition according to SMPTE ST 410. One Generic Stream Partition shall include one Subtitle Instance.

The EBU recommends the use of option A.1. However it is recognised that many STL legacy subtitle files still exist and are used for production today. If no conversions of these STL files into EBU-TT instances are desired or reasonable, the use of STL files in MXF (option A.2) is also acceptable during a period of transition.

All Generic Stream Partitions that contain subtitles shall be placed at the end of the MXF file. If no Body Partitions are present, the Generic Stream Partitions shall be placed between the Header Partition and the Footer Partition, otherwise they shall be placed between the last Body Partition and the Footer Partition. A RIP shall be present at the end of the file and this RIP shall reference all Generic Stream Partitions.

Note: Depending on the use case there are advantages and disadvantages to place Generic Stream Partition at the beginning of the file or at the end. In order to minimize the options and in order to meet the Basic User Requirement defined in Annex D, the EBU recommends storing Subtitle Instances (as STL or EBU-TT) at the end of the MXF file.

If there is more than one Subtitle Instance present in one MXF file, every instance shall be placed in a separate Generic Stream Partition. This applies to both EBU-TT (see A.1) and STL (see A.2) mappings. In order to separate the different Subtitle Instances, the language and purpose (e.g. "captions" or "subtitles") shall be indicated. This shall take place within the MXF Header Metadata by using a mechanism that is currently undergoing standardisation in SMPTE (in the 31FS Ad-Hoc Group "mapping EBU-t3264 STL to MXF").

B — Inside the Generic Data Element (SMPTE 436)

Mapping of caption and subtitle data inside the Generic Container (GC) Data Element:

B.1 — ANC packets in MXF Generic Container data elements

There are three different ways of mapping subtitling data into the Ancillary Data Packet according to SMPTE ST 291:

B.1.a — SMPTE RDD 8

System-B Teletext Subtitles as specified in ETSI EN 300 706 are mapped into a VANC Data Packet that is specified in SMPTE RDD 8 (see **Annex B**).

B.1.b — SMPTE ST 2031

System-B Teletext Subtitles as specified in ETSI EN 300 706 are mapped into a SMPTE ST 2031 Ancillary Data Packet in accordance to ETSI EN 301 775 (VBI carriage in DVB) and ETSI EN 300 472. The EBU discourages the use of option B.1.b since this option is suitable only for MPEG-2-TS data and it provides an unnecessary complex and nested solution.

B.1.c — SMPTE ST 334-1

Closed Captions as specified in CEA 608 and CEA 708 are mapped into a SMPTE ST 334 Ancillary Data Packet. The EBU discourages the use of option B.1.c since this option is not applicable in European installations.

The EBU recommends the use of B.1.a. The EBU nevertheless recognizes that B.1.b and B.1.c are currently also in use in other regions of the world.

Note: SMPTE ST 436 defines the sequence order of the ANC elements and their corresponding picture elements.

B.2 — VBI lines in MXF Generic Container data elements

The EBU discourages the use of option B.2 since this option is only applicable for SD systems. Due to the different scanning rasters of HD systems, a 1:1 mapping of the VBI lines from SD to HD systems is not possible.

C — Inside the Generic Picture Element

C.1 — VBI lines in the Picture Element

The EBU discourages the use of option C.1 since this option is only possible where VBI lines are part of the encoded picture element (e.g. D10). In all other relevant SD and HD formats that are mapped into MXF files, this method is not applicable.

C.2 — Ancillary data in the Picture Element

Most of the video compression systems that are in use in the professional video domain support the transport of ancillary data by defining specific areas inside the compressed video bitstream, such as SEI (Supplement Enhancement Information) in the case of AVC.

The EBU discourages the use of option C.2 since the subtitle data is stored deep inside the compressed video bitstream. Post production processes such as editing, updating or deleting subtitles are very complex to perform.

Note: Table 1 provides a general and simplified overview of the advantages and disadvantages of the three recommended MXF internal methods. For this purpose the methods A.1 and A.2 are being considered equally. For real-life installations application-specific considerations have also to be taken into account.

Table 1: Overview between different MXF internal methods

Use Case	EBU-TT or STL in MXF Generic Stream Partitions (method A.1 and A.2)	ANC packets inside the MXF Generic Container Data Element (method B.1)
Partial restore	-	+
Streaming	-	+
Adding subtitles to growing file	-	+
Editing	-	+
“Low latency” playback (MXF stream)	-	+
“Low latency” playback (MXF file)	+	+
Easy to add/remove subtitles	+	-

Easy to add additional Subtitle Instance (e.g. more languages)	+	-
Late binding (i.e. Subtitles and audio-visual essence can be produced separately)	+	-
Migrating from MXF external to MXF internal	+	-
Migrating from MXF internal to MXF external	+	-
Describing the Subtitle Instance (e.g. the language)	+	-

For all use cases that process, edit or re-use parts of a Subtitle Instance, method B.1 is more suitable. In contrast for all use cases that process, edit or re-use complete Subtitle Instances the methods A.1 and A.2 are more suitable.

5.2 Recommendations for MXF external usage

The EBU recommends the use of EBU-TT files (option E) as a versatile, extensible XML-based Subtitle format.

Note: During a period of transition the use of STL files (option D) is also acceptable.

In this part of the recommendation the following types of external subtitle files are considered (refer to Figure 1 for the overview):

D	STL format. More information can be found in **Appendix B**
E	EBU-TT format. More information on EBU-TT can be found in **Appendix B**.
F	Other (proprietary) formats that exist but that are not part of this recommendation.

In the MXF external usage there is the need to have a linking mechanism between MXF file and the subtitle file.

Such a linking mechanism shall be based on a common unique identifier, e.g. a unique file name or a programme identifier which typically might be provided by a central media asset management system.

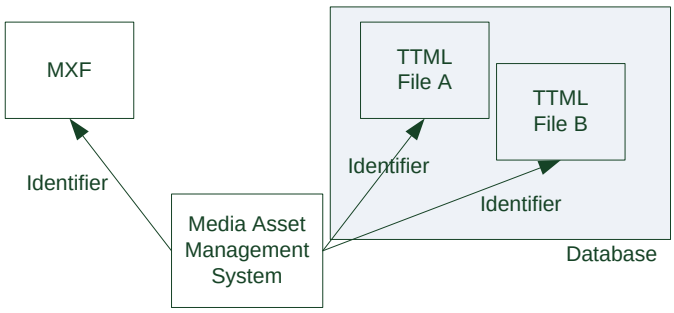


Figure 2: Linking mechanism of MXF and EBU-TT via a central Media Asset Management System

The EBU encourages the use of option E. EBU-TT files are stored separately from the MXF-file (e.g. in a central repository such as a database). Linking shall be provided from a central media asset management system to both the MXF files and to the EBU-TT file(s). This mechanism also provides the possibility to use several EBU-TT files for one MXF file. Each complete Subtitle

Instance shall be placed in a separate EBU-TT file (e.g. different languages, captions and subtitles). There shall not be any direct references between the MXF files and the EBU-TT files as double referencing shall be avoided. Figure 2 shows the linking mechanism for MXF with several EBU-TT files.

The EBU discourages the use of option D and option F. However it is recognised that many legacy STL subtitle files still exist and are used for production today. If no conversions of these STL files into EBU-TT instances are desired or reasonable, the use of MXF files and corresponding external STL files (option D) is also acceptable during a transition period.

6. Bibliography

CEA 608-E	Line 21 Data Services
CEA-708-D	Digital Television (DTV) Closed Captioning
EBU Tech 3264	Specification of the EBU Subtitling data exchange format
EBU Tech 3350	EBU-TT Part 1 Subtitling format definition
ETSI EN 300 706 V.1.2.1 (2003-04)	Enhanced Teletext Specification
ETSI EN 300 743 V.1.3.1 (2006-11)	Digital Video Broadcasting (DVB): Subtitling Systems
ETSI EN 300 775 V.1.2.1 (2003-05)	Digital Video Broadcasting (DVB); Specification for the carriage of Vertical Blanking Information (VBI) data in DVB bitstreams
SMPTE ST 291-2011	Television — Ancillary Data Packet and Space Formatting
SMPTE ST 328-2000	Television — MPEG-2 Video Elementary Stream Editing Information
SMPTE ST 334-1-2007	Television — Vertical Ancillary Data Mapping of Caption Data and Other Related Data
SMPTE ST 336-2007	Television — Data Encoding Protocol Using Key-Length-Value Mapping of Vertical Ancillary Data Packets and Extended Video Line Data into Video DIF Blocks of DV-Based 50 Mbit/s DIF Stream Format
SMPTE ST 374-2003	Television — Mapping of Vertical Ancillary Data Packets (VANC) into VAUX DIF Blocks of DV-Based 100 Mbit/s DIF Stream Format
SMPTE ST 375-2003	Television — Mapping of Vertical Ancillary Data Packets (VANC) into VAUX DIF Blocks of DV-Based 25 Mbit/s or 50 Mbit/s Streams and Extended Video Line Data into VAUX DIF Blocks of DV-Based 25 Mbit/s Stream
SMPTE ST 376-2003	Material Exchange Format (MXF) — File Format Specification
SMPTE ST 377-1	Television — Material Exchange Format (MXF) — Mapping Type D-10 Essence Data to the MXF Generic Container
SMPTE ST 386-2004	Television — Material Exchange Format (MXF) — Generic Stream Partition
SMPTE ST 410-2008	Television — MXF Mappings for VBI Lines and Ancillary Data Packets Carriage of DVB/SCTE VBI Data in VANC Storage and Distribution of Teletext Subtitles and VBI Data for High-Definition Television

Page intentionally left blank. This document is paginated for two sided printing

Annex A: More information on STL and EBU-TT

STL

STL is a binary file format specified in EBU Tech 3264 (1991) and has been the de-facto standard for subtitle exchange for many years. Its use is widespread and it can still be found in many archives and subtitle databases. Reference to the audio-visual material timing is achieved through timecode values. The file structure of STL files is simple and does not provide many options (limited support for colour of characters and background, and for character size and very basic formatting such as underline, italics, boxing). This helps for reliable exchange, however the feature set of STL is very limited.

EBU-TT (EBU-Timed Text)

EBU Timed Text (EBU-TT) provides an easy-to-use method to interchange and archive subtitles in XML. It is based on the W3C Timed Text Markup Language (TTML) specification and can be seen as a constrained version of the W3C spec, aimed at providing a solution more tailored to broadcast operation.

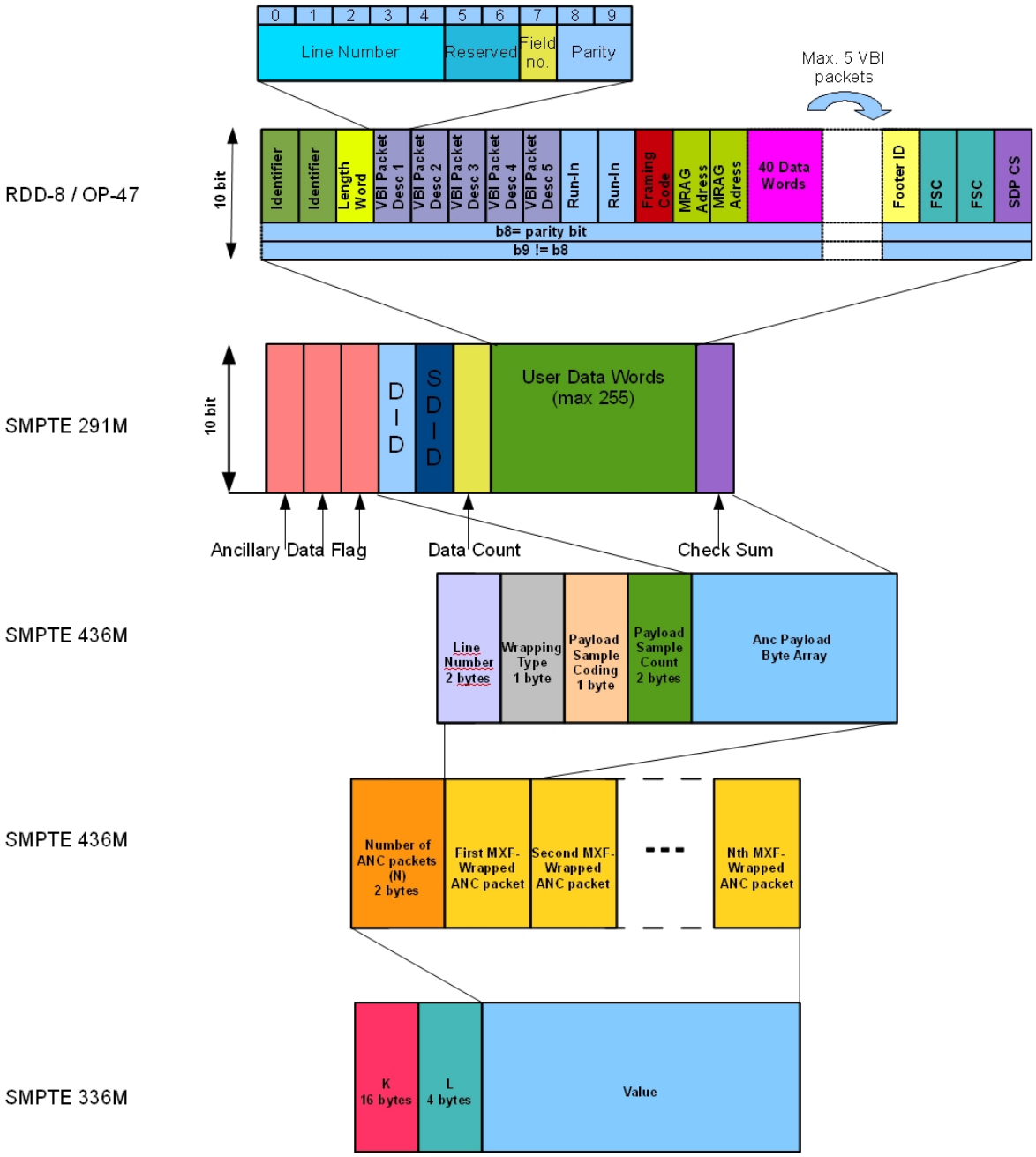
The purposes of EBU-TT include interchange among authoring systems, transcoding and exchanging among legacy formats and in general the support of file-based HDTV production.

EBU-TT offers a more creative set of tools than legacy formats. Subtitles can be placed all over the picture, metadata can be included and different screen sizes are supported.

As many broadcasters still use STL or have archived STL files, EBU-TT ensures backwards compatibility with its predecessor.

Page intentionally left blank. This document is paginated for two sided printing

Annex B: Mapping System-B Teletext Subtitles inside GC data elements



SMPTE RDD 8 (Registered Disclosure Document) deals with the storage and distribution of System-B-Teletext data into the vertical ancillary data space for High-Definition Television. Each Data Packet shall comply with the data format defined in SMPTE ST 291 for a 10-bit type 2 ancillary space packet. For further information on the data structure of a SMPTE RDD 8 Data Packet please refer to SMPTE RDD 8.

For further information on the MXF Frame Wrapping mechanism please refer to SMPTE ST 436-2006 Table 6.

Page intentionally left blank. This document is paginated for two sided printing

Annex C: Use cases

This Annex provides an overview of the use cases that have currently (in 2012) been identified. Since many use cases and workflows are rapidly changing in this domain, this overview cannot be exhaustive.

Where not otherwise noted, the following considerations apply for both MXF internal and MXF external methods.

1. Production

a. Creation of subtitle

b. Editing of subtitles

c. Editing of audio/video essence (including the need of "re-synchronising" the subtitles)

d. Linking between external Subtitle Instance(s) and an MXF file(s)

e. Format conversion (e.g. from STL into EBU-TT and vice versa)

2. Exchange and Delivery

a. Insertion of subtitles into an MXF file (MXF external into MXF internal)

b. Extraction of subtitles from an MXF file (MXF internal into MXF external)

3. Playout

a. Synchronous playout of audiovisual content and subtitles

4. Storage and Archiving

a. Complete restore of archived MXF files

b. Partial restore of archived MXF files

Page intentionally left blank. This document is paginated for two sided printing

Annex D: User requirements

Based on the use cases given in Annex C, the following overview lists basic and advanced user requirements for the use of subtitles and MXF:

Basic user requirements (BUR)

Req. ID	Description
BUR-1	Partial restore of the audio/video essence and subtitles should be possible
BUR-2	Subtitle Instances should be easy to add or remove from the audio/video essence
BUR-3	It should be possible to indicate the language of the subtitles
BUR-4	It should be possible to indicate the purpose of the subtitles (e.g. “captions”, “subtitles”)
BUR-5	It should be possible to support two or more different languages together with one MXF file
BUR-6	It should be possible to have a list of the related “internal or external” subtitles instances including where/how they are stored

Advanced user requirements (AUR)

Req. ID	Description
AUR-1	Streaming of the audio/video essence and subtitles should be possible. This includes the support of playing out the audio/video essence and subtitles while receiving an incomplete MXF file. Note: Complete download of subtitles before initiating the playout of the essence is an acceptable implementation
AUR-2	It should be possible to add subtitles while receiving an incomplete MXF file.