SMPTE ST 2075:2013

SMPTE STANDARD

Mapping EBU TECH 3264 (STL) into the MXF Generic Stream Container

Table of Contents

Page

Foreword ... 2
Intellectual Property ... 2
1 Scope ... 3
2 Conformance Notation ... 3
3 Normative References ... 3
4 Definition of Acronyms, Terms and Data Types ... 4
5 General ... 4
5.1 Header Metadata ... 4
6 Essence Descriptors ... 6
6.1 Event Text Descriptor ... 7
6.2 STL Descriptor ... 7
6.3 STL SubDescriptor ... 8
7 Generic Stream Partition Pack ... 9
7.1 Generic Stream Partition Pack Key ... 9
7.2 Generic Stream Partition Pack Length ... 9
7.3 Generic Stream Partition Pack Value ... 9
8 Generic Stream Data Element Coding ... 10
8.1 Generic Stream Data Element Key ... 10
8.2 Generic Stream Data Element Length ... 10
8.3 Generic Stream Data Element Value ... 10
9 Essence Container Data ... 11
10 STL Instances Outside the MXF File (External Essence) ... 11
Annex A Bibliography (Informative) ... 12
Annex B Use cases (Informative) ... 13
B.1 Use Case 1 (Alignment Signals are not present inside the MXF File) ... 13
B.2 Use Case 2 (Alignment Signals are present inside the MXF File) ... 13
B.3 Values for Start Timecode and STL Reference Point Timecode ... 14

Approved
June 6, 2013

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 Operations Manual.

SMPTE ST 2075 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.

1 Scope

This document provides the specification on how to carry binary EBU TECH 3264 (STL) files into the MXF Generic Stream Container using the MXF Generic Stream Partition defined in SMPTE ST 410. This document specifies the Generic Stream Partition Pack, Generic Stream Data Element KLV coding and Essence Descriptors for STL files.

The document also defines how to link to STL essence which is external to an MXF file.

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. 2007) are functionally identical.

The following standards contain provisions that, 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.

RFC 5646, Tags for Identifying Languages

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 410:2008, Material Exchange Format — Generic Stream Partition

4 Definition of Acronyms, Terms and Data Types

For the purposes of this document, the terms and definitions given in SMPTE ST 377-1 and SMPTE ST 410 and the following apply.

GSI: General Subtitle Information (according to EBU TECH 3264).

STL: EBU “standard subtitle data exchange format” (according to EBU TECH 3264).

STL Generic Stream Partition: A Generic Stream Partition that carries EBU STL data.

TTI: Text and Timing Information (according to EBU TECH 3264).

5 General

EBU STL files store the subtitle data as binary data. For the mapping of this binary data into an MXF file, one or more Generic Stream Partitions that are defined in SMPTE ST 410 shall be used. One complete EBU STL file shall be mapped into one Generic Stream Partition as a single binary stream. For the mapping into MXF, one EBU STL file shall not be divided.

If multiple STL files are being mapped into one MXF file, each STL file shall be stored in an individual Generic Stream Partition. Each Generic Stream Partition shall be associated with a single Stream ID value.

5.1 Header Metadata

Following the provisions in SMPTE ST 410, in MXF files that store both audio-visual essence and STL data internally, there shall be one Top-Level File Package be present for each Essence Container and each Generic Stream. Hence, the Operational Pattern of such an MXF file shall be OP 1b or higher.

For MXF files that store either the audio-visual essence or the STL data outside the MXF file but still link to this external essence (see Section 10), there shall be one Top-Level File Package be present for each Essence Container and each Generic Stream. Hence, the Operational Pattern of such an MXF file shall be OP 1b or higher as well.

For MXF files that store only STL essence in one MXF file and do not link to any other external essence, the Operational Pattern of such an MXF file shall be OP 1a.

Note: Figure 1 illustrates an example of an MXF file layout with one internal Essence Container (in Body Partition 1) and one internal STL essence (in Generic Stream Partition 1). The order of Partitions is not defined, therefore, Generic Stream Partitions can exist before and/or after Body Partitions.

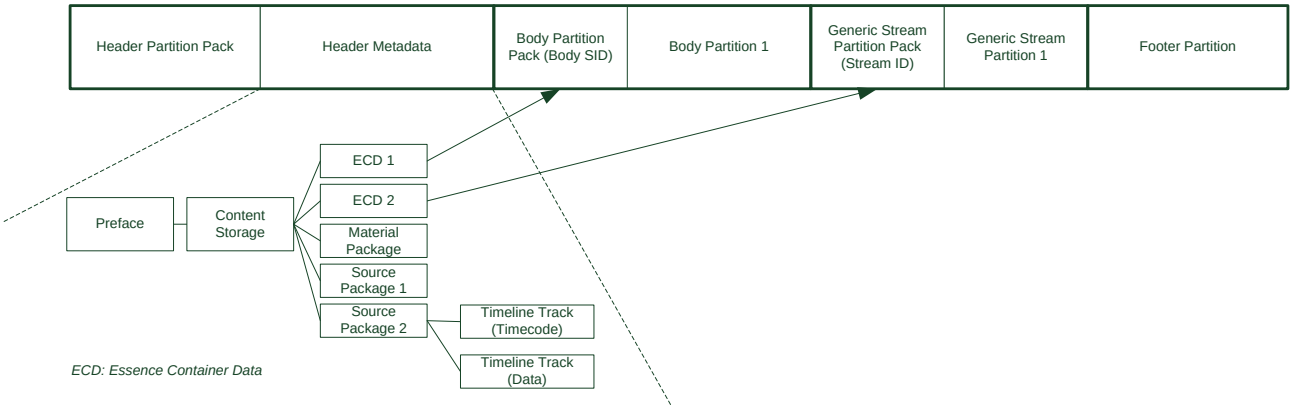

Figure 1 – Example of an MXF file layout

5.1.1 Tracks

Each Top-Level Source Package that links to one STL Generic Stream Partition (via the Essence Container Data set) shall link to exactly one Timeline Track (Data). The Track Number of this track shall be the last four bytes of the Generic Stream Data Element key which is defined in Section 7.1. This Timeline Track (Data) shall be described by a STL Essence Descriptor which is defined in Section 8.

Each Top-Level Source Package that links to one STL Generic Stream Partition (via the Essence Container Data set) shall also link to exactly one Timeline Track (Timecode) present. This track provides the time reference for the STL essence.

The value of the property Origin of this Timeline Track (Timecode) shall be set to zero. The value of the property Edit Rate of this track shall be set to the value of the “Disk Format Code” (DFC) inside the GSI block of the linked STL file and shall follow the rules given in Table 1.

Table 1 – STL Disc Format Code and MXF Edit Rate values

STL Disc Format Code (DFC)	Edit Rate value (numerator/denominator)
STL25.01	25/1
STL30.01	30/1 (in case of actual 30 frames per second essence) 30000/1001 (in case of 29.97 frames per second essence)

5.1.2 Timecode Component

The Timeline Track (Timecode) which has been defined in Section 5.1.1 shall link (via a Sequence set) to exactly one Timecode Component. Hence, there cannot be any timecode discontinuities. The value of the property Start Timecode in this Timecode Component shall define the start time code of this Top-Level Source Package which links to a Generic Stream Partition that carries STL data.

Source Package Timecode track Start Timecode and STL Reference Point Timecode values shall be defined as a frame count and shall be of type Position.

Note: There is no requirement for the two values to be the same.

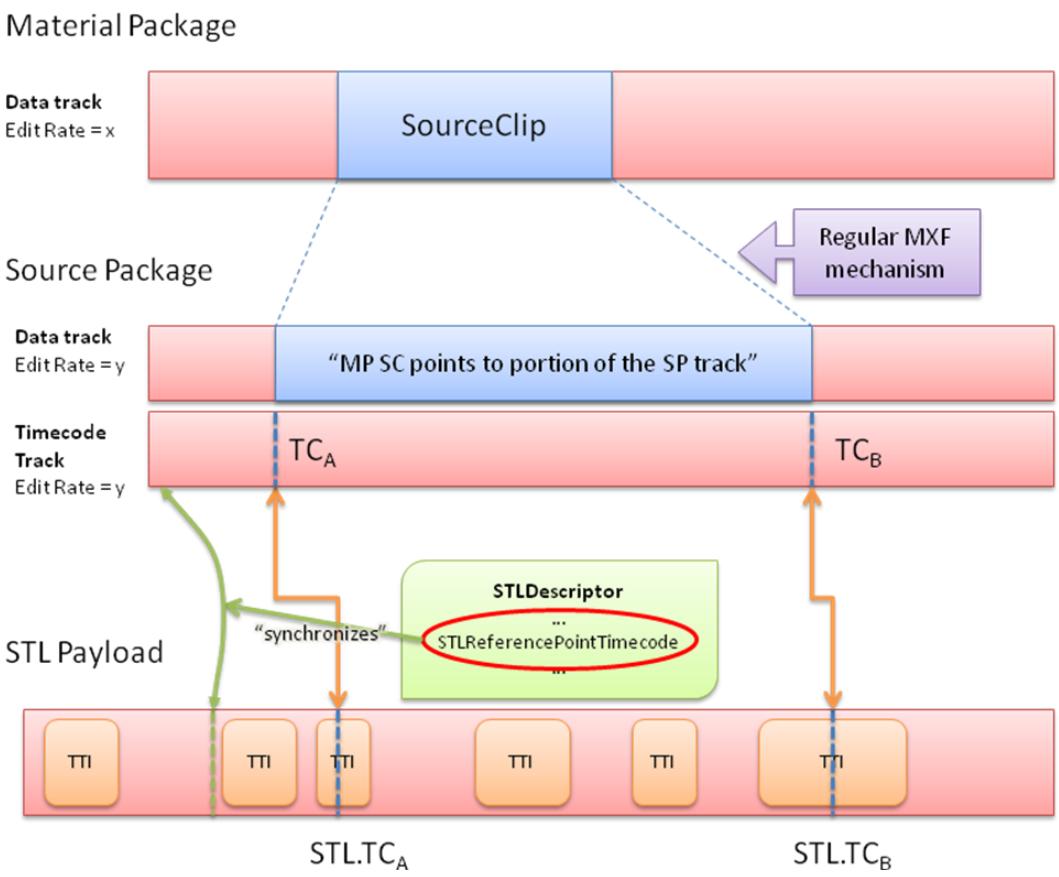

Figure 2 – MXF Header Metadata and STL relationship

6 Essence Descriptors

Essence Descriptors for STL files are called STL Essence Descriptor.

STL Essence Descriptor shall be a subclass of Generic Event Text Essence Descriptor.

6.1 Event Text Descriptor

Item Name	Type	Len	Local Tag	Item Designator	Req ?	Meaning	Default
Event Text Descriptor	Set Key	16		06.0E.2B.34. 02.7F.01.01. 0D.01.01.01. 01.01.6F.00	Req	Defines the Generic Event Text Essence Descriptor	
Length	BER Length	var			Req		
All items in SMPTE ST 377-1:2011 F.6 (definition of Generic Data Essence Descriptor)							
Event Text Kind	UL	16	Dyn	06.0E.2B.34. 01.01.01.0E. 03.02.01.08. 01.00.00.00	Req	Identifies the type (purpose) of the event text type, e.g. subtitles or captions. [**RP 224** specifies two ULs for this: “EBU-t3264 STL Subtitle Essence” and “EBU-t3264 STL Captions Essence”]	
Event Text Language Code	UTF-16 String	var	Dyn	06.0E.2B.34. 01.01.01.0D. 03.01.01.02. 02.15.00.00	Req	Specifies the language of the event text using RFC 5646 tags	

This shall be an abstract class.

6.2 STL Descriptor

Item Name	Type	Len	Local Tag	Item Designator	Req ?	Meaning	Default
STL Descriptor	Set Key	16		06.0E.2B.34. 02.7F.01.01. 0D.01.01.01. 01.01.70.00	Req	Defines the STL Essence Descriptor	
Length	BER Length	var			Req		
All items in 6.1 (definition of Generic Event Text Essence Descriptor)							
STL Reference Point Timecode	Position	16	Dyn	06.0E.2B.34. 01.01.01.0E. 07.02.01.02. 02.02.00.00	Req	Specifies the Reference Point Timecode of the STL essence. See Section 6.2.1.	

6.2.1 STL Reference Point Timecode

The STL Reference Point Timecode property provides the time reference mechanism into the STL data. The value of this property shall be the same as the “Time Code Start-of-Programme” (TCP) inside the STL GSI block.

6.3 STL SubDescriptor

STL Essence SubDescriptors shall be used in order to describe multiple languages that are stored in a single STL file which is mapped into a single MXF file.

Note: The intention of EBU STL was to store subtitle information for exactly one language inside one STL file. However, there exist EBU STL files that store more than one language into one file by using different lines for different languages, e.g. the first line of subtitles provides English Subtitles, and the second line provides French Subtitles. This work-around is possible for language combinations that use the same code page.

Item Name	Type	Len	Local Tag	Item Designator	Req ?	Meaning	Default
STL Sub-Descriptor	Set Key	16		06.0E.2B.34. 02.7F.01.01. 0D.01.01.01. 01.01.71.00	Req	Defines the STL Essence SubDescriptor	
Length	BER Length	var			Req		
All items in SMPTE ST 377-1:2011 B.3 (definition of MXF SubDescriptor)							
STL Line number	UInt 8	1	Dyn	06.0E.2B.34. 01.01.01.0E. 03.02.01.08. 02.00.00.00	Req	The line number on which a specific language is being visible.	
Event Text Language Code	UTF-16 String	var	Dyn	06.0E.2B.34. 01.01.01.0D. 03.01.01.02. 02.15.00.00	Req	Specifies the language of the event text using RFC 5646 tags	

The value of the element line number shall be a decimal, one-based counter and shall identify the line from top to down. Since the first STL line (line number = ”1”) is used for the primary language and is being described by the STL Essence Descriptor (see Section 6.2), the value of the element line number for the 1st STL Essence SubDescriptor shall be ‘2’.

Note: Figure 3 illustrates an example in which three different languages (English, French, and German) are stored in one STL instance. On the right side of Figure 3 the corresponding STL Essence Descriptor and two STL Essence SubDescriptors are shown.

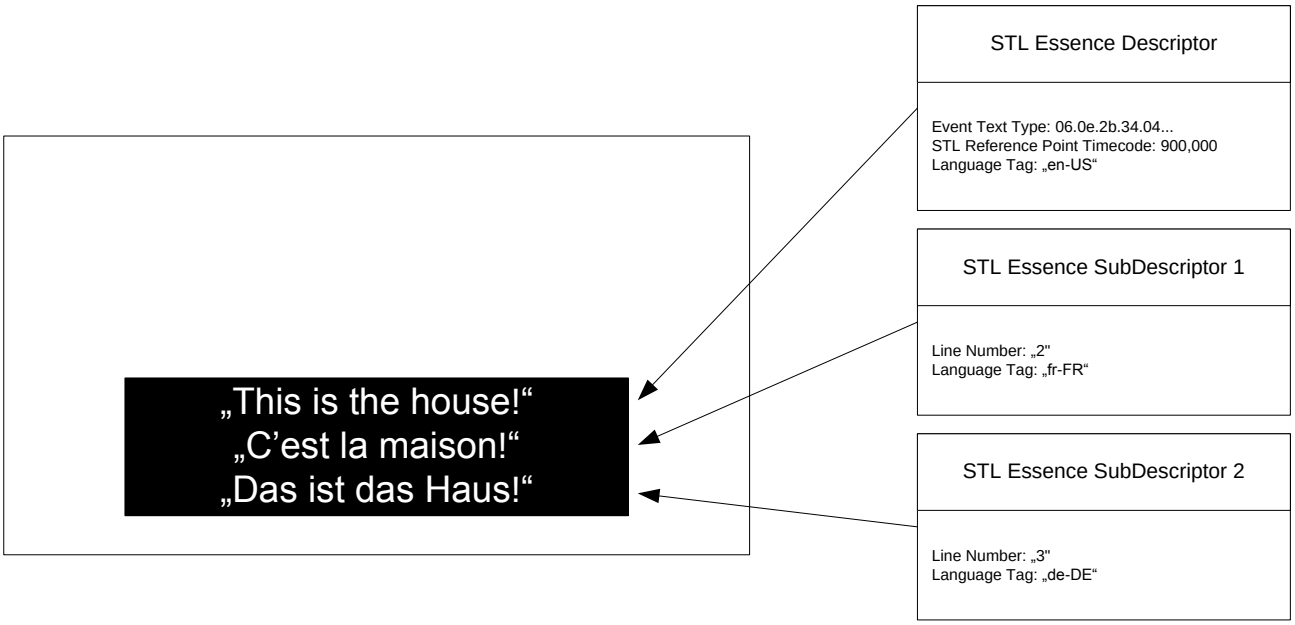

Figure 3 – Example of three different languages stored in one STL instance

7 Generic Stream Partition Pack

The Generic Stream Partition Pack shall comprise a Generic Stream Partition Pack Key, a Length and a Value as defined in SMPTE ST 410.

7.1 Generic Stream Partition Pack Key

The Key of the Generic Stream Partition Pack shall be as defined in SMPTE ST 410.

7.2 Generic Stream Partition Pack Length

The length field shall be as defined in SMPTE ST 410.

7.3 Generic Stream Partition Pack Value

The value of each item within the Generic Stream Partition Pack shall be as defined in SMPTE ST 410.

The property Essence Containers shall contain the Universal Label for the STL Essence which is defined in Table 2.

Table 2 – Value of EssenceContainers batch

Item Name	Type	Len	Value
EssenceContainers	Batch of UL (Essence Container)	16	06.0E.2B.34.04.01.01.0A 0D.01.03.01.03.01.00.00

8 Generic Stream Data Element Coding

The Generic Stream Partition shall contain a single KLV triplet which wraps the entire content of one complete EBU TECH 3264 (STL) file. This section provides the specification on Generic Stream Data Element coding for each STL file

8.1 Generic Stream Data Element Key

The default Generic Stream Data Element key defined in SMPTE ST 410 shall be used.

Byte 12 “Data signaling of the Generic Stream Data Element key signals the data arrangement applied to the underlying data and the values shall be as given in Table 3.

Table 3 – Value of Generic Stream Data Element Key Byte 12

Bit	Value	Meaning
0	1	Marker bit to prevent termination of key
1	0	The KLV is not a part of the data and should be removed before processing
3, 2	10	The Generic Stream is Big-Endian or is a byte stream
7-4	0000	Reserved – set to 0

Byte 13 “Wrapping signaling ”of the Generic Stream Data Element key signals the wrapping strategy applied to the underlying data and the values shall be as given in Table 4.

Table 4 – Value of Generic Stream Data Element Key Byte 13

Bit	Value	Meaning
0	1	Marker bit to prevent termination of key
1	0	The first byte of a KLV triplet has no special importance.
2	0	The Generic Stream is not divided into Access Units.
3	0	The Generic Stream Data is not Frame Wrapped.
7-4	0000	Reserved – set to 0

8.2 Generic Stream Data Element Length

The length field shall be as specified in SMPTE ST 410.

8.3 Generic Stream Data Element Value

The value shall be the entire document of one complete EBU TECH 3264 (STL) file.

9 Essence Container Data

Each EBU STL file shall be described by one Essence Container Data Metadata Set as defined in SMPTE ST 377-1, Annex A.5

The BodySID property of the Essence Container Data object shall be the stream identifier of the Generic Stream that contains the STL data.

10 STL Instances Outside the MXF File (External Essence)

The STL essence may be located external to the MXF file. In this case, the Locators property inside the STL Essence Descriptor shall be present and shall link to exactly one Network Locator Set.

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. 2009) are functionally identical.

EBU TECH 3264, Specification of the EBU Subtitling Data Exchange Format (February 1991)

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

SMPTE RP 210, Metadata Element Dictionary

SMPTE RP 224, SMPTE Labels Register

Annex B Use Cases (Informative)

Figure B.1 illustrates a real-life example where a 25 Hz tape-based recording has been ingested in an MXF file. The original time code on the tape was starting at 09:58:00:00 and was running to 10:30:00:00. From timecode 09:58:00:00 to timecode 09:59:59:24 there were alignment signals present on the tape. During the ingest process a new timecode for the Top Level File Package has been created and the start value for this has been set to 02:00:00:00.

There is a corresponding STL file present which holds three TTI blocks. The “TTI 1” block falls into the period where the alignment signals are present. This TTI block typically displays metadata, e.g. who has created the subtitles.

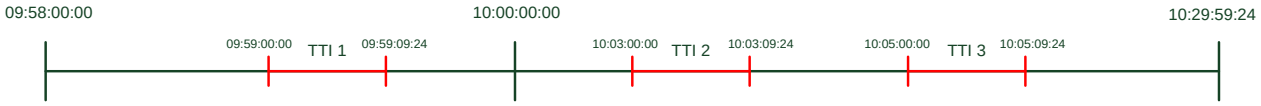

Figure B.1 – STL file example

B.1 Use Case 1 (Alignment Signals are not present inside the MXF File)

In this case the alignment signals have not been ingested, hence, the 1st frame inside the MXF file is the 1st frame of the program, i.e. the frame at time code 10:00:00:00 on the tape.

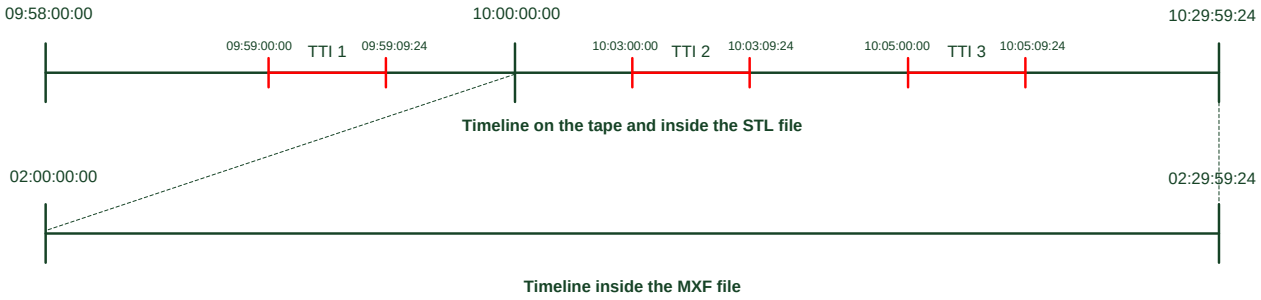

Figure B.2 – STL and MXF file – Use case 1

B.2 Use Case 2 (Alignment Signals are present inside the MXF File)

In this case the alignment signals have been ingested into the MXF file, hence, the 1st frame inside the MXF file is the 1st frame of the alignment signal, i.e. the frame at time code 09:58:00:00 on the tape.

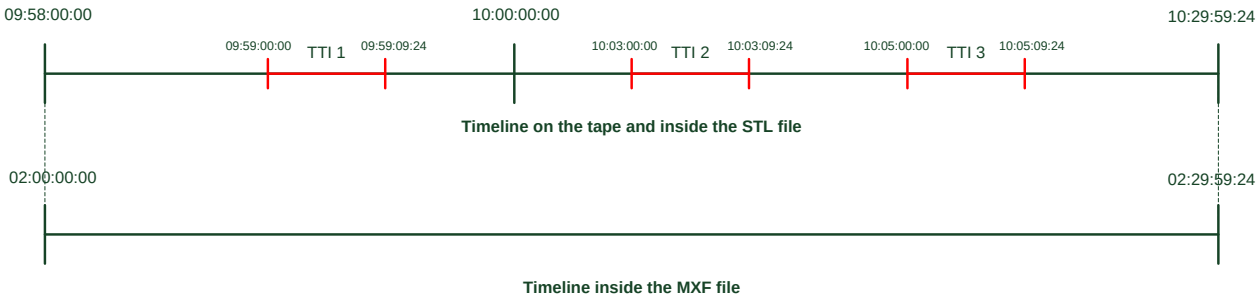

Figure B.3 – STL and MXF file – Use case 2

B.3 Values for Start Timecode and ST Reference Point Timecode

Table B.1 provides the values for the following properties:

- Start Timecoce inside Timecode Component set which is linked by the Top-Level Source Package
- STL Reference Point Timecode inside the STL Essence Descriptor

Table B.1 – Values for properties Timecode Component and STL Reference Point Timecode

	STL file		**MXF – Timecode Component**	**MXF – STL Essence Descriptor**
	STL “Timecode: Start-of-Progamme” (TCP)	STL “Timecode: first in-cue” (TCF)	Value of property Start Timecode	Value of property STL Reference Point Timecode
Use case 1	10:00:00:00	10:03:00:00	02:00:00:00	900,000 (= 10:00:00:00)
Use case 2	10:00:00:00	10:03:00:00	02:00:00:00	897,000 (= 09:58:00:00)