SMPTE ST 2120-1:2026-03

SMPTE STANDARD

Extensible Time Label — Structure

Table of Contents   Page

List of Figures — Page

List of Tables — Page

Foreword

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

SMPTE Engineering Documents are drafted in accordance with the rules given in its Standards Operations Manual. This SMPTE Engineering Document was prepared by Technology Committee 32NF Network/Facilities Infrastructure.

Introduction

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

SMPTE ST 12-1 specifies time code and is one of the oldest SMPTE Standards. It was first adopted in 1975 and has been widely used for many systems outside SMPTE's normal area of purview. It was developed for analog television recording systems and thus dealt only with interlaced television systems operating with frame rates up to 30 frames per second. It has, however, been flexible enough in design to be used in digital television systems, both standard and high definition. Still, various limitations are known that cannot be addressed within the limits of its design.

This standard introduces the extensible time label (TLX), describes its characteristics, and specifies its architecture. Other documents in this suite expand upon this specification, allowing for different configurations suitable for varied use cases.

TLX can provide a label for every media unit, and each label can be globally unique. The design of TLX overcomes the limitations most commonly encountered when using time code. For systems and workflows that could benefit from TLX, migration and/or interoperation is supported, as TLX can provide a usable bridge to those systems and workflows that continue to rely on time code.

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

1 Scope

The TLX Suite is a multipart standard. This document is one part of the TLX Suite. The parts collectively describe a system for defining labels for elements of media content, representing the labels in various forms, associating the labels with specific elements of the content, specifying profiles of the labels for particular purposes, conveying individual or grouped labels over various connection means, and storing individual or grouped labels in various formats.

This document part describes the architecture and data model of extensible time labels (TLX), the relationship to content, one specific representation in the JSON format, and general considerations for the design of applications of TLX Labels.

2 Conformance

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 clause explicitly labeled as "Informative" or individual paragraphs that start with "Note:"

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

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

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

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

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

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

3 Normative References

The following standard contains provisions that, through reference in this text, constitute provisions of this standard. Dated references require that the specific edition cited shall be used as the reference. Undated citations refer to the edition of the referenced document (including any amendments) current at the date of publication of this document. All standards are subject to revision, and users of this engineering document are encouraged to investigate the possibility of applying the most recent edition of any undated reference.

IETF RFC 8259 *The JavaScript Object Notation (JSON) Data Interchange Format.*
https://datatracker.ietf.org/doc/html/rfc8259

ISO 8601-1:2019 *Date and time — Representations for information interchange — Part 1: Basic rules.*
https://www.iso.org/standard/70907.html

4 Terms and Definitions

For the purposes of this document, the following terms and definitions apply:

4.1
TLX Data Model
data model

abstract model of data representing a TLX Label, association of a TLX Label with a Media Unit, and aggregation of a TLX Label in a TLX Label Sequence

Note 1 to entry: The TLX Data Model is specified in Clause 6.

4.2
TLX Representation
representation

standardized format in which to communicate an instance of a TLX Data Model

4.3
TLX Processor
processor

subsystem that uses an instance of the TLX Data Model

4.4
TLX Encoder
encoder

subsystem that writes an instance of the TLX Data Model using a specific TLX Representation

Note 1 to entry: See also the definition of <time label> source defined in 4.15 source

4.5
TLX Decoder
decoder

subsystem that reads an instance of the TLX Data Model in a specific TLX Representation

4.6
TLX Application
application

subsystem that incorporates at least one TLX Processor, TLX Encoder, or TLX Decoder

4.7
TLX System

at least two interoperating TLX Applications

4.8
TLX Profile

set of required and prohibited items and other constraints such that a TLX is suited to a stated purpose

4.9
TLX Suite

all parts of the SMPTE standards documents numbered 2120

4.10
TLX Label Sequence

ordered collection of one or more TLX Labels

4.11
TLX Label
extensible time label
TLX

data structure, containing metadata associated with one media unit, comprising one or more TLX Items

Note 1 to entry: The TLX Label can be present in the media sequence or might be inferred from other TLX Labels in the media sequence or external data (see Clause 8.2.3).

4.12
TLX Item

named component of a TLX Label comprising one or more TLX Item Attributes

4.13
TLX Item Attribute
TLX Attribute
attribute

metadata comprising a name component and a value component

Note 1 to entry: The “name-value” representation used in JSON does not require a length as used in key-length-value (KLV) representations.

4.14
essence

sound, picture, and data resources that constitute a media sequence

EXAMPLES audio, video, captions, subtitles, dynamic metadata

Note 1 to entry: See Clause 5.2.

[SOURCE: SMPTE ST 377-1:2019, Clause 4.1 — Adapted from definition of essence; Note 1 to entry has been added.]

4.15
source

<time label> generator of time labels

4.16
source

<media> generator of media units

4.17
media unit

smallest temporal increment of access to essence

EXAMPLES a video frame or field, a single audio sample, a block of contiguous audio samples, a subtitle, or a caption

Note 1 to entry: The granularity of the media unit, e.g., samples vs. blocks or frames vs. fields, is determined by an application or implementation.

Note 2 to entry: Temporal increment is not necessarily fixed for media units in a media sequence (refer to 4.26, 4.27, and 4.28).

Note 3 to entry: Media unit is similar to the MXF editable unit as defined in SMPTE ST 377-1:2019.

[SOURCE: SMPTE ST 429-7:2006, Clause 4 — Adapted from definition of editable unit; Examples and Notes 1, 2, and 3 to entry have been added.]

4.18
media sequence

continuous ordered collection of one or more media units

Note 1 to entry: See Clause 5.2

4.19
media event

one media unit, as the sole member of a media sequence

EXAMPLES a subtitle, caption, or instance of timed text, a stand-alone video frame or block of audio

4.20
media unit instant

point on a time axis representing the position of a media unit

4.21
media unit time point

mark attributed to a media unit instant by means of a specified time scale

Note 1 to entry: Such marks are identified by a corresponding duration relative to the origin of the specified time scale.

Note 2 to entry: For periodic signals complying with SMPTE ST 2059-1, the media unit time point might be the "Alignment Point" as required by that standard.

4.22
media unit interval

part of the time axis limited by two consecutive media unit instants

Note 1 to entry: The former media unit instant is part of the interval; the latter media unit instant is not.

Note 2 to entry: The latter media unit might not exist, as when the former media unit is the last in a media sequence, in which case the media unit interval describes the media unit instant where a next consecutive media unit would occur.

[SOURCE: ISO 8601-1:2019 — Adapted from definition of time interval; Notes 1 and 2 to entry have been added.]

4.23
media unit duration

measurement of a media unit interval by means of a specified time scale

4.24
adjunct interval

interval, other than a media unit interval, corresponding to a media unit

EXAMPLE an interval during which a media unit is captured

4.25
adjunct duration

measurement of an adjunct interval by means of a specified time scale

4.26
fixed-rate media

essence composed of media units constrained to have a constant media unit duration

4.27
variable-rate media

essence composed of media units not constrained to have a constant media unit duration

Note 1 to entry: Media having a rate that is not standard, but is constrained to have a constant media interval throughout a media sequence, is not considered variable-rate media.

4.28
regular-cadence media

essence composed of media units whose media unit duration changes according to a predictable pattern.

EXAMPLE a block of contiguous audio samples, such as a block of 1602 or 1601 samples following the pattern 1602, 1601, 1602, 1601, 1602 samples for an average of 1601.6 samples

4.29
virtual

repetitive TLX Labels, TLX Items, or TLX Attributes that can be omitted

4.30
sparse

TLX Label, TLX Item, or TLX Attribute that is not virtual

Note 1 to entry: A label that has not been omitted from the bitstream.

4.31
inferred

TLX Label, TLX Item, or TLX Attribute that is reconstructed based on sparse labels or external data

5 TLX Architecture

5.1 Architecture Overview

The TLX Architecture defines the TLX Data Model, the relation of the data model to content and media units, representations of the data model, and systems of applications composed of TLX Encoders, TLX Decoders, and TLX Processors.

These parts of the TLX Architecture are described in Clauses 5.2, 5.3, 6, 7, and 8.

5.2 Content Structure and Media Organization

5.2.1 Packaged Media (Informative)

An abstract data model for how packaged content is structured is described in the EBU/SMPTE Task Force for Harmonized Standards for the Exchange of Programme Material as Bitstreams Final Report [TFHS Final Report, Section 4.3.3]. An overview of that data model is shown in Figure 1.

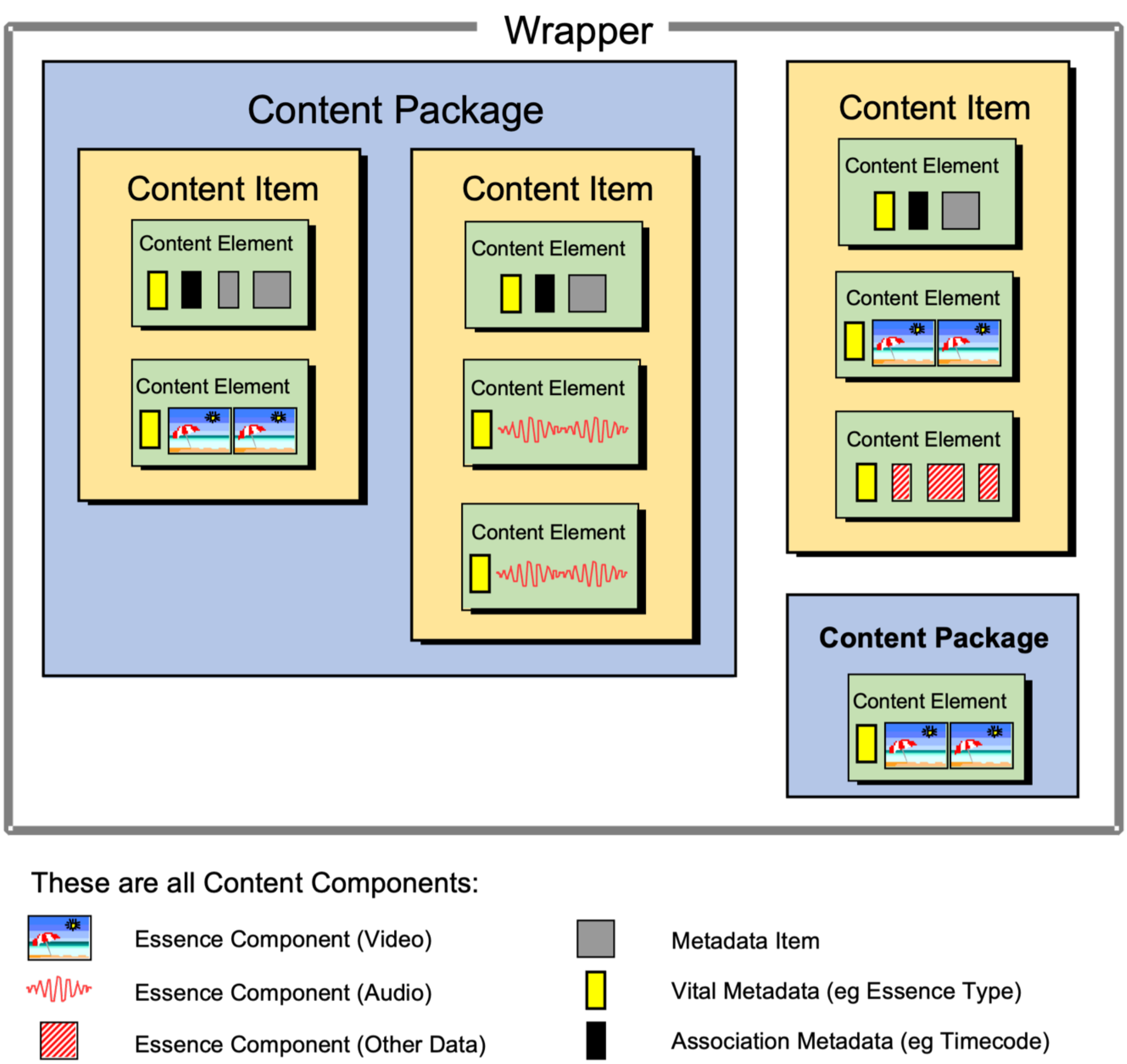

Figure 1 — Abstract Data Model for Content.

5.2.2 Media Units (Informative)

Media comprises one or more essences, each having a type (e.g., “video” or “audio”). An essence component (a Content Element, e.g., a video clip) comprises one or more media units of the same kind (e.g., video frames). Where media comprises more than one essence, they are synchronized.

The granularity of a media unit can vary by application or implementation. For example, an audio clip can be organized as a series of samples for a single channel, as a block of samples for a single channel, as a series of multi-channel samples, or as a block of multi-channel samples. The duration of the media unit is defined according to the granularity. For example, a series of audio samples can have a fixed sample rate of 1/48000 seconds. An audio block might span a particular fixed duration because the blocks have a uniform number of samples. An audio block might span a variable duration if blocks vary in their number of samples. Depending on the application, the granularity can be different from the organization, e.g., a block of single-channel audio samples could be treated as representing a portion of a series of samples, rather than as one block of a series of blocks.

The duration of media units in an essence can be constrained to a fixed value or allowed to vary. Video is commonly seen as a fixed rate media, with frames having a uniform, usually standardized, spacing in time. However, in some real-time protocols, video rates are permitted to dynamically adapt to environmental conditions, such that the spacing in time of video frames can vary.

5.2.3 Media Sequences and Events

A media sequence shall comprise at least a first media unit.

A media event is a special case of media sequence. A media event shall comprise exactly one media unit.

In the more general case, a media sequence may further consist of additional media units of the same kind as the first. (For example, the media units of a video clip are all video frames). These additional media units shall follow the first consecutively, forming a continuous, ordered set.

5.2.4 Organizing Principles (Informative)

To exist, media requires organizing principles to arrange media units. Organizing principles allow media sequences to be kept or accessed in the appropriate order and define synchronization among component essences.

In some cases, standards provide an organizing principle, as does the Media Exchange Format (MXF) specified in SMPTE ST 377-1. Time code, as specified in SMPTE ST 12-1 and related documents, has been used in many contexts as an organizing principle, for example, by supporting synchronization of audio and video essence tracks, or identifying specific portions of media sequences for use in the creation of a new media sequence, such as specified by the edit decision lists in SMPTE ST 258.

In other cases, an organizing principle is established by a workflow or policies, for example, when a video clip is represented by a folder containing sequentially numbered image files.

In some cases, time labels can be a component of an organizing principle. In other cases, time labels can be a complete organizing principle. Configurations of time labels suitable for use in or as an organizing principle are not specified in this document.

5.3 Media Unit Timing

5.3.1 Media Unit Instant, Intervals, and Durations (Informative)

The relationship between media units in a media sequence is diagrammed in Figure 2.

Conceptually, a media unit instant is the position of a media unit on the time axis as specified by other standards or industry practice.

The measure of time between consecutive media unit instants is the media unit duration.

The time during which the media unit is captured, during which the media unit is to be presented, or other associated intervals, can be represented by an adjunct interval for such purpose.

The measure of an adjunct interval is the adjunct duration and can be zero.

If an adjunct interval is not specified, assumptions to be made about a specific TLX Item can be provided by the document specifying that TLX Item.

Figure 2 shows a media unit with one adjunct interval representing one aspect of media unit timing that, for this example, is entirely within the media unit interval.

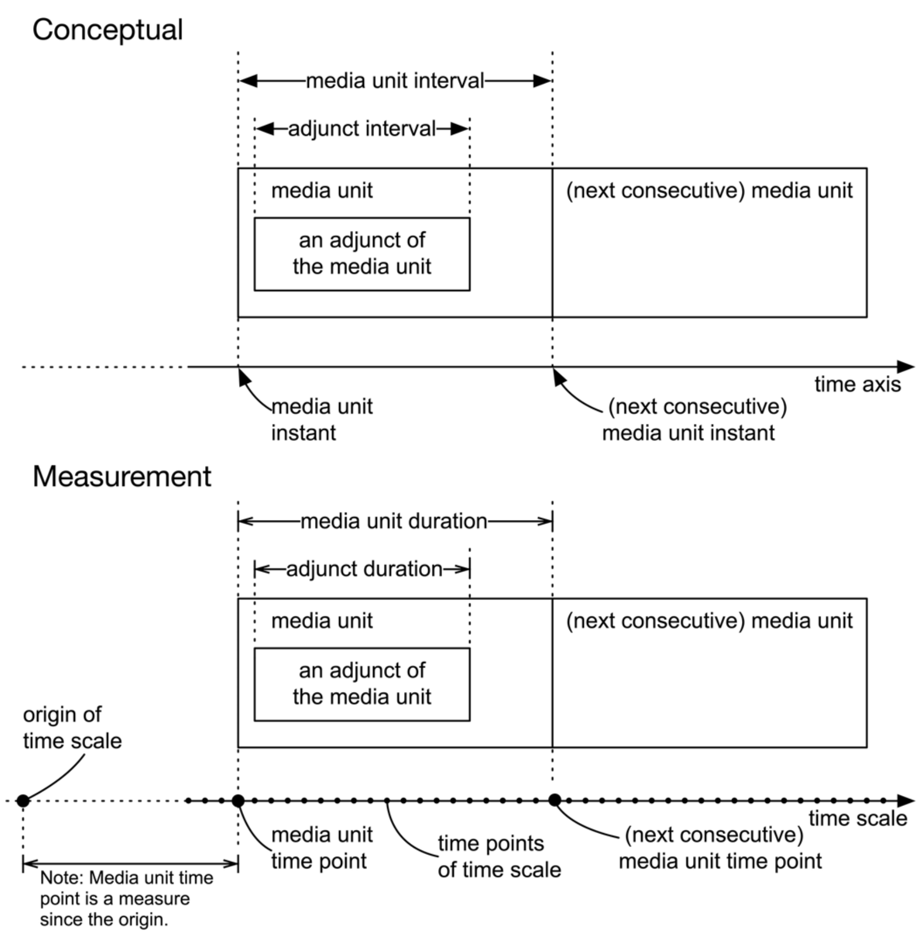

Figure 2 — Media Unit Timing Diagram.

5.3.2 Multi-sample Media Units

Where the media unit comprises more than one sample, then the media unit instant should correspond to the first sample in the media unit.

EXAMPLE
Each media unit might contain 1920 audio samples at a sample rate of 48,000 samples per second. The media unit duration equals the media unit interval of 1/25 seconds. The last sample in each media unit starts at 1/48000 seconds before the media unit instant of the next media unit.

5.3.3 Labeling and Cardinality

Each TLX Label shall label exactly one media unit. Each media unit may be labeled by zero, one, or more TLX Labels. Each TLX Label should be self-consistent, but need not be identical to other TLX Labels associated with the media unit.

6 TLX Data Model

6.1 Data Model Overview

The TLX Data Model defines the component parts, aggregation, and associations of a TLX Label.

6.2 TLX Naming Constraints

The names of TLX Labels, TLX Items, and TLX Item Attributes shall be a string composed only of the characters A-Z, a-z, 0-9, and underscore (_), and shall begin with an alpha character (A-Z, a-z) or an underscore (_).

NOTE This is the same as in the specification of “symbol” in ST 335, in the clause titled “Symbol”.

6.3 TLX Label Sequence

A TLX Label Sequence shall be a contiguous, ordered collection of one or more TLX Labels.

The ordering mechanism shall be specified. The ordering mechanism can be based on any attribute of the labels or items within the labels.

6.4 TLX Label

A TLX Label comprises a collection of TLX Items and each TLX Item comprises a collection of TLX Item Attributes. TLX Labels, TLX Items, and TLX Item Attributes are shown in Figure 3.

A TLX Label shall be a contiguous, unordered collection of one or more TLX Items.

The ordering of items within the collection shall not be specified, except by a TLX representation outside of which the ordering need not be preserved.

A TLX Label shall have the data type of object (see Table 2).

Each instance of a TLX Label shall comprise one or more TLX Items as shown in Figure 3. This capability provides extensibility: TLX Labels can be extended by introducing new kinds of TLX Items.

A TLX Label shall not contain more than one instance of any identically named TLX Item.

6.5 TLX Item

A TLX Item shall be named and shall be a contiguous and unordered collection of one or more TLX Item Attributes.

The ordering of attributes within the collection shall not be specified, except by a TLX representation outside of which the ordering need not be preserved.

The name of a TLX Item shall be specified. The name of a TLX Item shall conform to the TLX naming constraints as defined in Clause 6.2.

The name of a TLX Item need not be carried literally by a TLX representation.

A TLX Item shall have the data type of object (see Table 2).

Specification of a TLX Item shall include one or more TLX Item Attributes available for inclusion in an instance of the TLX Item. TLX Item Attributes need not have meaning outside of an instance of the TLX Item specifying them.

Each instance of a TLX Item shall be composed of one or more TLX Item Attributes, as shown in Figure 3. This provides extensibility, as the expressive capabilities of TLX Items can be extended by introducing new kinds of TLX Item Attributes.

A TLX Item shall not contain more than one instance of any identically named TLX Item Attribute.

Specification of a TLX Item may include additional constraints as to which TLX Item Attributes and combinations thereof are allowed in an instance of the TLX Item.

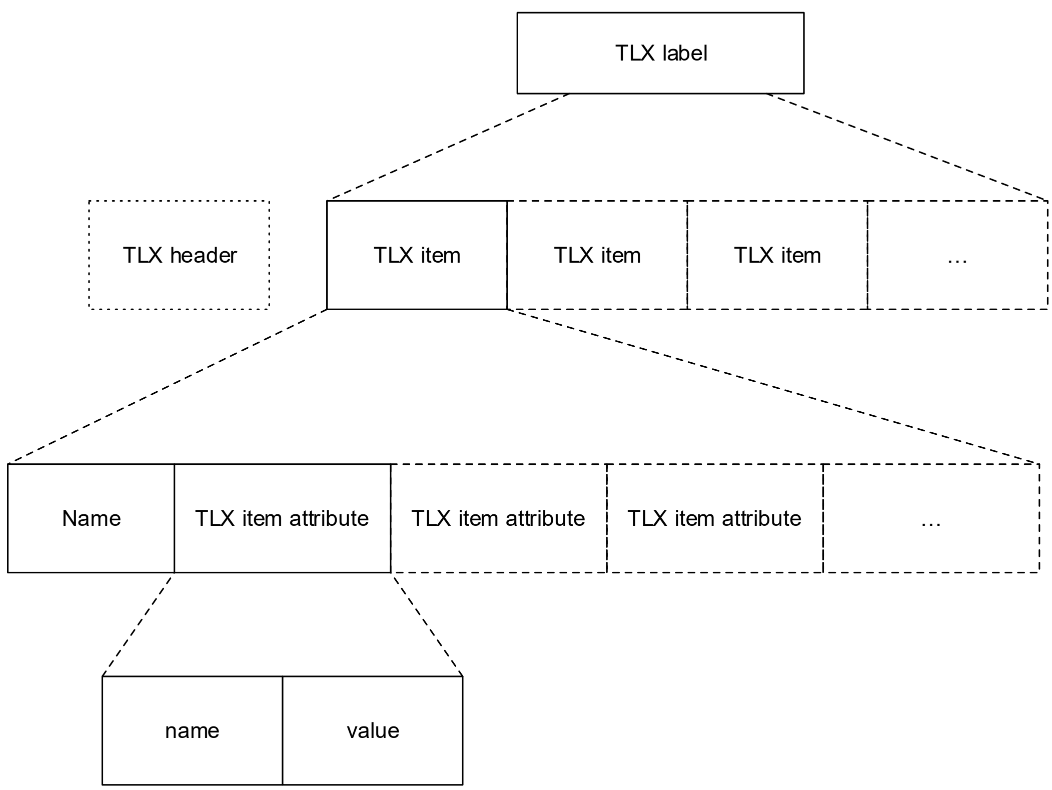

Figure 3 — Extensible Time Label (TLX) Structure.

6.6 TLX Item Attribute

A TLX Item Attribute shall be a name-value pair.

The name of a TLX Item Attribute shall be specified. The name of a TLX Item Attribute shall conform to the TLX naming constraints as defined in Clause 6.2. To avoid confusion, the name of a TLX Item Attribute shall be unique across all TLX Items.

TLX Item Attributes are strongly typed as follows:

- The type of value of a TLX Item Attribute shall be a single specified type from Table 1 or Table 2.
- Constraints on the value shall be specified sufficiently to bound the set of allowed values such that TLX representations can assign concrete encodings.
- These constraints should be specified sufficiently to allow TLX representations to assign finite encodings.

For a TLX Item Attribute whose value is of a structural type (from Table 2), each of the structural element values shall be of a single specified type from Table 1 or Table 2 and, for each of the structure element values, constraints on the value shall be specified sufficiently to bound the set of allowed values such that TLX representations can assign concrete encodings.

EXAMPLE 1
Constraints for a "number" type value could require the value to be "an integer in the range of [0, 100]", which could allow one TLX representation to encode the value as an unsigned byte, or another TLX representation to encode the value as a string of one to three digits (or perhaps a fixed-length string of three digits). Tighter constraints, where possible, can allow more efficient representations, as an integer in the range [0, 63] could be encoded by different TLX representations in six bits or two digits.

EXAMPLE 2
Constraints for an "array" type could specify that the array elements are numbers themselves having certain constraints and might specify a maximum, or an exact, array length.

6.7 Time Label Structures (Informative)

Specification of specific TLX Items, TLX Item Attributes, and the constraints for each, is not specified in this document. However, any such specification of these elements ought to allow for future extensibility.

A complete JSON example of a time label, based on an example specification of a fictitious metadata item, attributes, type definitions, and value constraints, is given in Annex A, along with an example JSON schema based on the specification.

7 TLX Representations

7.1 Representations Overview

A TLX representation is a standardized format in which to communicate an instance of the TLX data model.

Several TLX representations are envisaged, including JSON, XML and KLV. Each representation specifies a canonical form for TLX when conveyed in that format.

The JSON TLX representation is defined in Clause 7.3. Other representations are not specified in this document.

7.2 Encoder and Decoder Requirements

7.2.1 TLX Encoder

A TLX Encoder creates an instance of a TLX data model in a specific TLX Representation.

A TLX Encoder may retain sparse TLX Labels, TLX Items, or TLX Item Attributes, allowing redundant labels and/or elements to become virtual.

The behavior of a TLX Encoder is not specified in this document.

7.2.2 TLX Decoder

A TLX Decoder receives an instance of the TLX data model in a specific TLX representation.

A TLX Decoder may use sparse TLX Labels and/or elements to infer labels and/or elements that were made virtual by an encoder.

A TLX Decoder should not omit components of a TLX Label.

A TLX Decoder may omit some portions of the instance produced, as determined by a specific TLX application or TLX profile.

The behavior of a TLX Decoder is not specified in this document.

7.3 JSON Representation

7.3.1 JSON Representation General

JSON is defined by IETF RFC 8259. JSON is also defined by ISO 21778, and the provisions of the ISO specification are intended to be a subset of those of the RFC. In the event of a conflict, the RFC shall take precedence. ISO 21778 is listed as a bibliographic reference. A TLX Encoder applies the mapping rules specified in Clause 7.3.3 to an instance of the TLX Data Model, and a conformant JSON generator (IETF RFC 8259 Section 10) produces a JSON representation of the TLX Data Model.

A TLX Decoder applies a JSON Parser (RFC 8259 Section 9) to a JSON representation, and then applies the inverse of the mapping rules to produce a conformant instance of the TLX Data Model.

7.3.2 JSON Data Types

JSON data types are defined by RFC 8259 Sections 2 - 7, and listed in Table 1 and Table 2 of this document. In the event of a conflict, RFC 8259 shall take precedence.

Table 1 — Primitive Data Types

Data type	Definition	Constraints
number	a member of the set of real numbers	
string	a series of zero or more Unicode characters	Encodings shall use UTF-8.
Boolean	a truth value (i.e., one of true or false)	
null	a literal value of this type is a representation for a missing value	Except as elsewhere constrained, a value of type null is an allowed value for any other primitive, structured, or derived type. NOTE This could be used to represent "unknown" or "not available"

NOTE 1 The types listed in Table 1 are based on the like-named primitive data types specified in IETF RFC 8259.

Table 2 — Structured Data Types

Data type	Definition	Constraints
array	an ordered series of zero or more values	The values in an array should be of a single type. A value of null in an array is allowed.
object	an unordered collection of zero or more name/value pairs	A name shall be a string of non-zero length. Within an object, each name shall be unique.

NOTE 2 The types in Table 2 are based on the like-named, structured data types specified in IETF RFC 8259.

7.3.3 TLX JSON Mapping Rules

7.3.3.1 Mapping Rules for Objects

The following mapping rules apply:

1. A TLX Label Sequence may be mapped to a JSON Array (IETF RFC 8259, Section 4),
2. A TLX Label that is not virtual shall be mapped to a JSON Object (IETF RFC 8259, Section 4),
3. A TLX Item that is not virtual shall be mapped to a JSON Object (IETF RFC 8259, Section 4),
4. A TLX Attribute that is not virtual shall be mapped to a member of the corresponding TLX Item per JSON Object (IETF RFC 8259, Section 4).

NOTE Virtual labels, items, and attributes are not mapped.

7.3.3.2 Mapping Rules for Attributes

A TLX Attribute that is not virtual shall be mapped to a JSON member (IETF RFC 8259, Section 4). The attribute shall be mapped to a member of its TLX Item. The attribute name shall be mapped to the name string of that member. The attribute value shall be mapped to the value of that member as follows:

1. A scalar value shall be transformed into a JSON number value having no more precision than specified for the attribute value. A scalar value constrained to have an integer value shall be represented by a JSON number having no decimal point,

2. A string value shall be transformed into a JSON string value. As JSON is UTF-8 encoded, strings originating in other encodings can require conversion,

3. A Boolean value shall be transformed into the corresponding JSON literal values of true or false,

4. An ordered list of values shall be mapped exactly to a JSON Array (IETF RFC 8259, Section 5) and the array element values shall each be transformed as herein described for attribute values, recursively. The order of the list values shall be the same as the order of the corresponding array elements,

5. An unordered list of named values shall be mapped exactly to a JSON Object (IETF RFC 8259, Section 4), each named value being mapped to a corresponding member as herein described for attributes, recursively. The order of the named values need not be preserved by the order of the corresponding members,

The JSON literal value of null shall not be used.

7.3.3.3 Inverse Mapping Rules

The following inverse mapping rules apply:

1. A bare JSON array may be mapped to a TLX Label Sequence, providing each of its values is a JSON object representing a TLX Label.

2. A JSON object shall be mapped to a TLX Label, item or attribute, depending on context. A bare JSON object shall be mapped to a TLX Label, providing all of its members are JSON objects and at least one member is a recognized TLX Item. Members of a JSON object that is a TLX Item shall be mapped to a TLX Attribute of that item.

3. If the JSON member name is the name of a known TLX component, the TLX decoder shall first construct an empty target instance of that component, then populate the contained components by iteratively applying these inverse mapping rules to the JSON value.

4. If the JSON value is a JSON array, the TLX decoder shall create one TLX Attribute for each value in the JSON array by iteratively applying the inverse mapping rules listed in this clause to the individual values in the JSON array.

5. If the JSON value is a JSON number and the target is a TLX Attribute with a mapping trait that indicates that the value may be condensed, the TLX decoder shall parse JSON value to obtain an integer.

6. If the JSON value is a JSON string and the target is a TLX Attribute with an inverse mapping trait, the TLX decoder shall apply the inverse mapping trait to the JSON value.

7. Otherwise, the TLX decoder shall interpret the JSON value as a textual string.

7.3.4 JSON Schema (Informative)

Several initiatives are under way to develop a JSON Schema language that could be used to describe the syntax and semantics of generalized JSON documents. It is possible that the JSON representation of TLX could be expressed in such a language.

The most notable initiative is that underway at https://json-schema.org

However, the proponents of JSON Schema are not presently pursuing due process standardization of their work, thus it is usable only as a bibliographic reference.

8 TLX Applications

8.1 Applications Overview (Informative)

This clause discusses various topics relevant to TLX Applications and Systems.

A TLX Application is a system that incorporates at least one TLX Processor, TLX Encoder or TLX Decoder.

A TLX System includes at least two interoperating TLX Applications.

A TLX Profile is a set of required and prohibited items and other constraints such that a TLX is suited to a stated purpose.

One example TLX Application is a TLX Label source comprising a single TLX Encoder. Another example TLX Application is a reader comprising a TLX Decoder to read TLX Labels and a TLX Processor to perform some function (e.g., set a flag), based on values therein. A simple TLX System could be the two of these applications configured so the time label source supplies TLX Labels to the reader.

8.2 Application Considerations

8.2.1 Sources (Informative)

A time label can be generated in conjunction with the generation of a media unit, such that the same device or process is both the media unit source and the time label source.

EXAMPLE
A camera can label images as it captures them.

A time label can be generated for an extant media unit by a “labeler” – a device or process operating as a time label source. Such a labeler could be employed, for example, downstream from a camera that generates images, but does not generate labels; or when retrieving or refreshing archived media.

In some workflows, a media sequence with TLX Labels can be given additional TLX Labels. The correspondence between labels can be recorded in a database. The design or implementation of such a database is not specified in this document.

A media source could simultaneously generate more than one media unit, or more than one media sequence. Such simultaneously generated media units or media sequences can be related or not. Configurations of time labels suitable for distinguishing between media units or media sequences simultaneously generated and/or labeled by the same media/label source, and/or indicating when they are related, are not specified in this document.

8.2.2 Extensibility

An application could encounter a time label having items and/or attributes that it does not recognize. This is expected to occur eventually, because the time label structure is designed to be extensible.

An application shall operate in the presence of unrecognized time label items and unrecognized attributes.

An application should preserve unrecognized time label items and unrecognized attributes. This allows downstream applications to use metadata that might not be used or recognized by an intervening application.

When an application has no need of certain items or attributes, even though they are well-known, the application can treat the unneeded items or attributes as if they are unrecognized.

8.2.3 Sparse and Inferred Labels (Informative)

Sometimes, a label associated with a media unit of a media sequence provides metadata that is redundant with respect to another label associated with a consecutive media unit of the media sequence: There can be metadata in items that are unchanging, or items that change predictably among labels associated with the consecutive media units of a media sequence. As an optimization, an application can omit redundant attributes and items, and can omit labels containing only redundant items, and rely on the constant or predictable nature of the metadata to restore the original label for any media unit of the media sequence. When repetitive labels, items, or attributes are omitted, they are said to be “virtual”, and the remaining labels are said to be “sparse”. Labels, items, or attributes that are reconstructed based on sparse labels or external data are said to be “inferred”.

EXAMPLE 1
The labeling of audio essence is one expected use of sparse labeling. Labeling each media unit of an audio clip, with each sample being a media unit and appearing at a rate of 48000 per second, would be significantly inefficient. However, a single label could be provided for the first media unit of the clip, and the labels for the remaining media units could all be virtual. This would be quite appropriate if the clip were contained in a file. An editor would address the first media unit of the file by that sparse label and could address any other media unit in the file by a virtual label.

EXAMPLE 2
Labels could be provided at the start and then periodically throughout the audio clip. For instance, once per second or at the beginning of each audio block would be more appropriate for streams of media. To begin such an audio clip at a media unit having a virtual label, a switch could reconstruct an inferred label for the new start point. Thus, downstream receivers would be able to reconstruct any of the labels following, such that an inferred label can be available for any media unit.

EXAMPLE 3
Sparse labeling can also be employed where other data associated with the media units, besides or in addition to metadata in the sparse labels, is sufficient to inform the reconstruction of virtual labels, items, or attributes. One expected use for this technique is the sparse labeling of individual captions within a timed text file (for example SMPTE ST 428-7), where a single label corresponds to the first caption, or alternatively, to the start of the timed text file. The timed text file provides other data regarding individual captions, which can be used to reconstruct an inferred label for any caption in the file.

The methods for inferring TLX Labels, TLX Items, and TLX Item Attributes are not specified in this document.

8.2.4 Historical Data (Informative)

A TLX Label can contain the history of TLX information associated with that media unit. Mechanisms for associating historical information with a TLX Label are not specified in this document.

Annex A
(Informative)

Example Fictitious Time Label, Item Specification, Schema, and Instance

A.1 General

The normative structure of the extensible time label (TLX) is specified in Clause 6. This annex provides an example of a fictitious time label in Clause A.4, which is consistent with the example specification of a fictitious TLX Item “TLXfoo”, provided by Clause A.2. An example schema for the example time label, drafted in accordance with the IETF Internet-Draft current as of this writing and indicated in the first line therein, is presented in Clause A.3. Such schemas can support validation of time label instances.

A.2 TLXfoo (a Fictitious TLX Item)

A.2.1 Description

TLXfoo is a fictitious time label item that provides an example of item definitions presented as the entirety of Clause A.2. The name of an item is provided as the name of a document clause, as with Clause A.2, though for non-fictitious items, such clauses would be normative, not informative. This clause A.2.1 identifies the purpose of the **TLXfoo** item and each of its fictitious attributes. Both the **bar** and **baz** attributes are examples for how to specify attributes. The **bar** attribute further illustrates the need for care with respect to string lengths.

A.2.2 Attributes

The attributes of **TLXfoo** would be as specified in Table A.1.

NOTE 1 In actual use, the verb would be a conformance term.

Table A.1 — TLXfoo Attributes

Name	Type	Constraints	Required/Optional	Description
bar	string	length of the value is from 1 to 4 characters	required	brief description of **bar** goes here
baz	number	value is in the interval [0,5]	optional; default = 1	brief description of **baz** goes here

While **bar** is constrained to be a length of four characters, that length represents a count of Unicode characters. Since the string type in TLX is constrained to be encoded in UTF-8, some characters can require a single byte, while other characters can require more than one byte.

NOTE 2 Take care in specifying the constraints of strings. Strings in JSON are Unicode, and in TLX are constrained to be UTF-8 encoded. Thus, the length of a string in characters can be different from its length in bytes.

The paragraphs following the attributes table represent further descriptions of the attributes or notes concerning the attributes or the item. Preferably, the attributes appear in these further descriptions in the same order as in the table. These further descriptions are not strictly required. Other subclauses could be provided to specify or describe other aspects of the item (e.g., processing).

A.3 Example Time Label Schema – JSON

JSON Schema, a JSON-based format for describing the structure (i.e., schema) of a JSON document, is documented by the December 8, 2020, IETF Internet-Draft: “JSON Schema: A Media Type for Describing JSON Documents”. Based on the example specification of the fictitious TLXfoo item in A.2, the following is an illustrative time label schema expressed in JSON Schema:

```
{
  "$schema": "http://json-schema.org/draft-07/schema",
  "$id": "http://smpte-ra.org/2120/1/2021/smpte-tlx-items-fictitious",
  "title": "TLX",
  "type": "object",
  "description": "This is an example schema for time labels based on a fictitious item,
TLXfoo.",
  "examples": [
    {
      "TLXfoo": {
        "bar": "चार",
        "baz": 5
      }
    }
      ],
  "properties": {
    "TLXfoo": {
      "$id": "#/properties/TLXfoo",
      "title": "TLXfoo",
      "type": "object",
      "description": "A fictitious TLX Item for illustration.",
      "examples": [
        {
          "bar": "दो",
          "baz": 5
```



```
        },
        {
          "bar": "two"
        }
      ],
      "properties": {
        "bar": {
          "$id": "#/properties/TLXfoo/properties/bar",
          "title": "bar",
          "type": "string",
          "description": "A fictitious string attribute, to illustrate character and byte counts can differ.",
          "minLength": 1,
          "maxLength": 4,
          "examples": [
            "four",
            "दो",
            "три"
          ]
        },
        "baz": {
          "$id": "#/properties/TLXfoo/properties/baz",
          "title": "baz",
          "type": "number",
          "description": "A fictitious number attribute, in the range [0,5].",
          "default": 1,
          "minimum": 0,
          "maximum": 5,
          "examples": [
```



```
          0,
          5
        ]
      }
    },
    "additionalProperties": true,
    "required": [ "bar" ]
  }
},
"additionalProperties": true,
"required": [ ],
"minProperties": 1
}
```

A.4 Example fictitious time label instance as JSON

The following JSON instance is compliant with the example fictitious schema of A.3.

```
{
     "TLXfoo": {
          "bar": "три",
          "baz": 5
     }
}
```

Bibliography (Informative)

EBU/SMPTE Task Force for Harmonized Standards for the Exchange of Programme Material as Bitstreams *"Final Report: Analyses and Results"*, August 1998, https://tech.ebu.ch/docs/techreview/ebu-smpte-tf-bitstreams.pdf

ISO 21788:2017 *Information Technology – The JSON data interchange system*, https://www.iso.org/standard/71616.html

SMPTE ST 12-1:2014 *Time and Control Code*. https://doi.org/10.5594/SMPTE.ST12-1.2014

SMPTE ST 258:2004 *SMPTE Standard – For Television – Transfer of Edit Decision Lists*. https://doi.org/10.5594/SMPTE.ST258.2004

SMPTE ST 335 *Metadata Element Dictionary Structure*. https://doi.org/10.5594/SMPTE.ST335

SMPTE ST 377-1:2019 *Material Exchange Format (MXF) – File Format Specification*. https://doi.org/10.5594/SMPTE.ST377-1.2019

SMPTE ST 428-7:2014 *Digital Cinema Distribution Master – Subtitle*. https://doi.org/10.5594/SMPTE.ST428-7.2014

SMPTE ST 429-7:2006 *D-Cinema Packaging – Composition Playlist*. https://doi.org/10.5594/SMPTE.ST429-7.2006

SMPTE ST 2059-1:2021 *Generation and Alignment of Interface Signals to the SMPTE Epoch*. https://doi.org/10.5594/SMPTE.ST2059-1.2021

The JSON Schema Organization, https://json-schema.org

Universal Binary JSON Specification, https://www.ubjson.org

Wright, A., H. Andrews; B. Hutton, G. Dennis, *"JSON Schema: A Media Type for Describing JSON Documents"*, IETF, December 8, 2020, https://datatracker.ietf.org/doc/html/draft-bhutton-json-schema-00

