SMPTE Public Committee Draft

Extensible Time Label — Structure

Table of Contents

Page

Table of Figures Page

Table 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.

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; then formal languages; then figures; and then any other language forms.

Introduction

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

SMPTE ST 12-1 specifies timecode, 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 timecode. 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 timecode.

[Editors notes: The following paragraph will be replaced with the appropriate patent information during the SMPTE Headquarters publication process.]

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

This standard describes the structure of extensible time labels (TLX) and the relationship to media essence.

The following are out of scope for this document: Specifications for specific time label items, configurations of time labels suitable for a particular use, specifications for mechanisms by which a media unit is labeled, and the particular process by which a virtual label is reconstructed.

2 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.

IEEE Std 754™-2008, IEEE Standard for Floating-Point Arithmetic

IETF RFC 8259 The JavaScript Object Notation (JSON) Data Interchange Format

3 Terms and Definitions

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

3.1. essence

sound, picture, and data resources that make up media

EXAMPLES: audio, video, captions, subtitles, camera metadata

[SOURCE: adapted from *essence* in SMPTE ST 429-7:2006]

3.2. media unit

smallest temporal increment of access to an essence

EXAMPLES: A media unit could be a video frame or field, a single audio sample, a block of contiguous audio samples, a subtitle, or a caption.

[SOURCE: adapted from *editable unit* in SMPTE ST 429-7:2006]

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

3.3. media sequence

sequence

collection of one or more media units of the same type of essence in an indexed order

3.4. media event

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

EXAMPLE: a subtitle, caption, or instance of timed text

3.5. media unit instant

point on a timeline representing the position of a media unit

Note: For periodic signals complying with SMPTE ST 2059-1, the Media Unit Instant is the "Alignment Point" as required by that standard.

3.6. media unit interval

part of a timeline bound by a media unit instant and a next consecutive media unit instant

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

Note 2: The latter media unit need not exist, as when the former media unit is the last in a sequence, in which case the media unit interval describes the position that a next consecutive media unit could occupy, were it appended.

3.7. media unit duration

length of time during which a media unit is captured or presented

3.8. fixed-rate media

essence composed of media units constrained to have a constant media interval

Note: ST 12 timecode was designed to work for fixed rate media.

3.9. variable-rate media

essence composed of media units not constrained to have a constant media interval

Note: 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.

3.10. TLX suite

all parts of the SMPTE standards documents numbered 2120

3.11. TLX label

extensible time label

TLX

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

Note: The TLX label may be present in the media sequence or may be inferred from other TLX labels in the media sequence or external data (see section 6.7).

3.12. TLX item

named component of a TLX label comprising one or more TLX item attributes

3.13. TLX item attribute

attribute

metadata comprising a name component and value component

Note: The "name-value" representation used in JSON does not require a length as used in key-length-value (KLV) representations.

3.14. source

<time label> generator of time labels

<media> generator of media units

4 Data Types

Any time label shall be representable in JSON, as specified in IETF RFC 8259, though conversion to and from other representations is expected. Time labels shall be based on primitive and structured data types listed and constrained in Table 1 and Table 2. Additional data types may be derived from these primitive and structured types, but derived data types are outside the scope of this document.

Table 1 - Primitive data types

Data type	Definition	Constraints

number	a member of the set of real numbers	Value shall be such that it can be represented exactly by an IEEE 754 binary64 number. [See NOTE 2]
string	a sequence of zero or more Unicode characters	Encodings shall use UTF-8.
boolean	a truth value (i.e., one of true and 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 represented “unknown” or “not available”

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

NOTE 2. This constrains integers to the range [$-2^{53} + 1$, $2^{53} - 1$]. Integers, including their appearance in rational values, are expected to be the most common numeric values used in TLX.

Table 2 - Structured data types

Data type	Definition	Constraints
array	an ordered sequence 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 3. The types in Table 2 are based on the like-named structured data types specified in IETF RFC 8259.

5 Media Organization

5.1. Media Units

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

An essence can be a sequence of media units, as is common for audio and video essences; or a collection of one or more media events (e.g., a timed-text file representing a series of captions).

The granularity of a media unit can vary by application or implementation. For example, audio can be organized as a sequence of samples for a single channel, as a block of samples for a single channel, as a sequence of multi-channel samples, or as a block of multi-channel audio. 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 need not be the same as the organization, e.g., a block of single-channel audio samples could be treated as representing a portion of a sequence of samples, rather than as one block of a sequence 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. Media Sequences and Events

A media sequence comprises at least a first media unit.

A media event is a special case of media sequence, composed of exactly one media unit.

In the more general case, a media sequence can further comprise additional media units of the same kind as the first (e.g., the media units are all video frames). These additional media units follow the first consecutively, forming a continuous, ordered set.

5.3. Organizing Principles

To exist, media requires organizing principles to arrange media units. Organizing principles allow 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. Timecode, 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 sequences for use in the creation of a new 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 sequence 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 is out of scope for this document of the TLX suite.

6 Time Labels

6.1. Time Label Overview

A time label contains metadata associated with a media unit.

The association of a time label with a media unit may be explicit. When this is the case, the time label is represented as data that is either embedded in or associated with the media unit, or explicitly related to the media unit, e.g., in a database. Except as discussed in section 6.7 in the context of sparse and inferred labels, specification of mechanisms by which a media unit is labeled by a time label is outside the scope of this document.

Some applications might define a TLX header that precedes the TLX label. The definition of the TLX header is out of scope.

6.2. Time Label Structure

A TLX label shall have the data type of object (see Table 2). Each TLX label shall comprise one or more TLX items as shown in Figure 1. This structure allows TLX labels to be extended by adding new kinds of TLX items.

A TLX item shall have the data type of object (see Table 2). Each TLX item shall have a name that is unique within its TLX label. Each TLX item shall comprise one or more TLX item attributes. TLX items can be extended by adding new TLX item attributes.

A TLX item attribute shall be a name-value pair. The type of the value shall be specified, along with any other constraints on the value. Each TLX item attribute shall be specified as being required or optional within its TLX item. Where optional, a default value for a TLX item attribute value may be specified. The name of an attribute shall be unique within its TLX item. To avoid a likelihood of confusion by human readers, the TLX item attribute name should be unique across all time labels.

Specification of specific time label items, and the detailed structure for each, is out of scope for this document in the TLX suite. However, any list of time label items and the detailed structures for each shall 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.

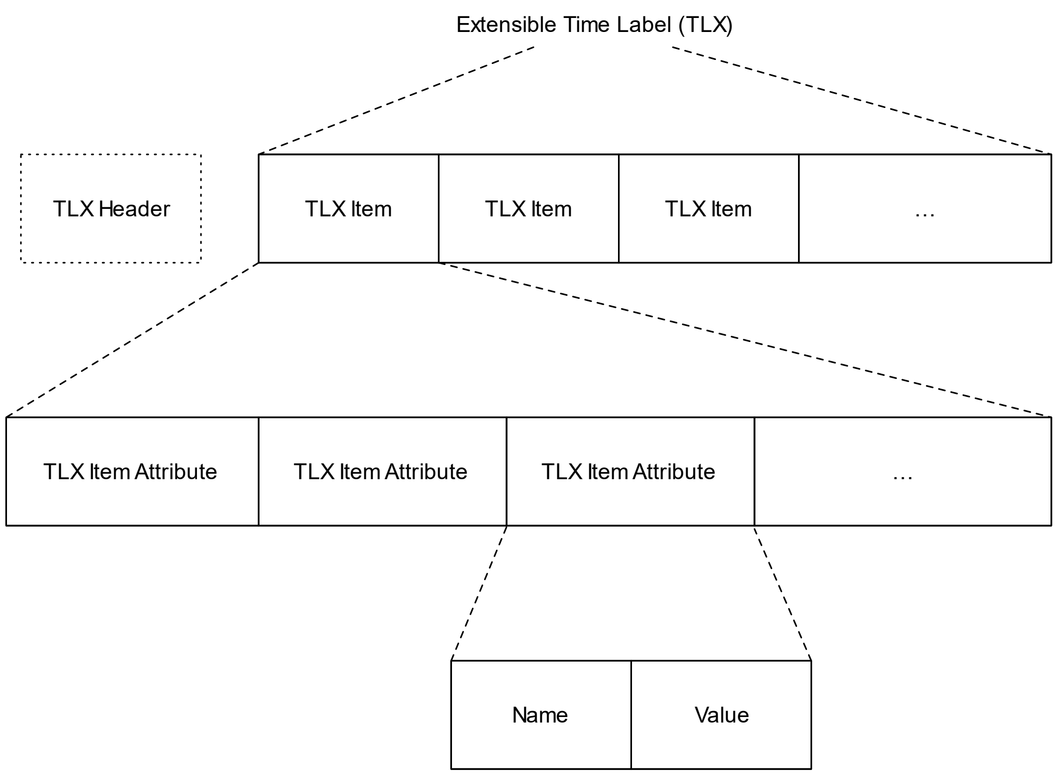

Figure 1. Structure of TLX labels.

6.3. Media Unit Timing

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

A media unit instant is the position of a media unit on the timeline as specified by other standards or industry practice.

The length of time between consecutive media unit instants is the media unit interval. The length of time during which the media unit was captured or the length of time during which the media unit is presented is the media unit duration.

The media unit is a conceptual model that supports multiple use cases:

1. The media unit duration is typically equal to the media unit interval.
2. The media unit duration may be zero: This allows media units to be modeled as delta functions.
3. The media unit duration can be greater than zero but less than the media unit interval: This allows the model to represent cases where a media unit occupies a portion of the media interval.

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

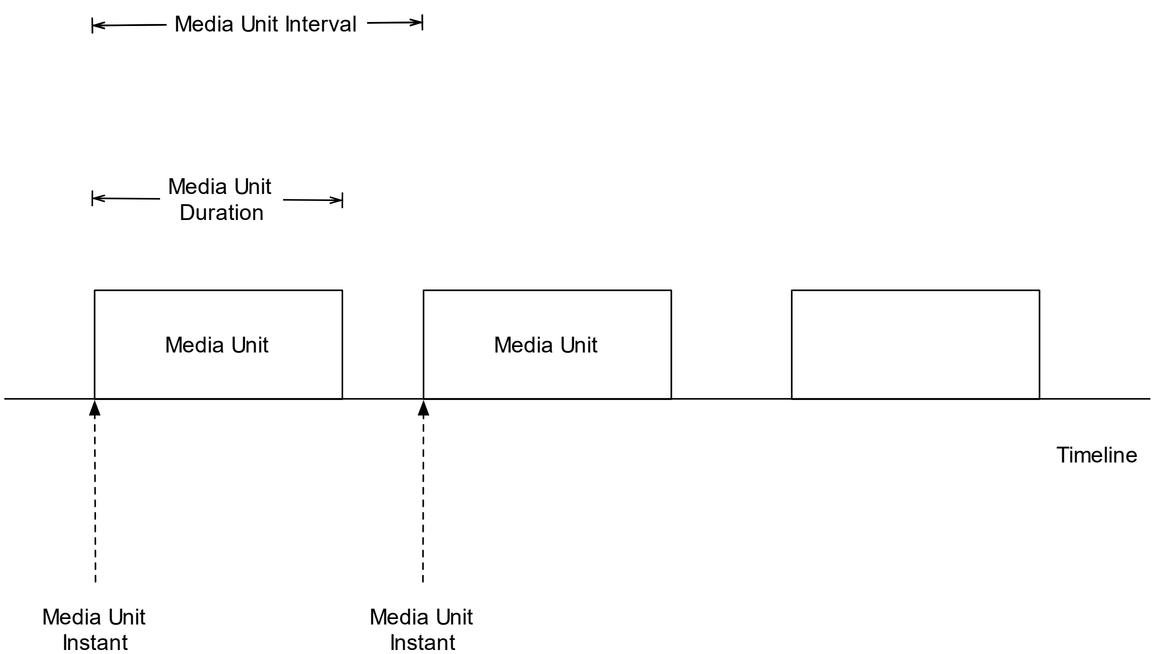

Figure 2. Media unit timeline.

Example: Each media unit might contain 1000 audio samples at a sample rate of 48,000 samples per second. The media unit duration equals the media unit interval of 1/48 seconds. The last sample in each media unit starts at 1/48000 seconds before the media unit instant of the next media unit.

6.4. Labeling and Cardinality

Each TLX label shall be associated as labeling 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 consistent with other TLX labels associated with the media unit.

6.5. Sources

A time label may 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. For example, a camera can label images as it captures them.

A time label may 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 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 out of scope for this standard.

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

6.6. 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 were unrecognized.

6.7. Sparse and Inferred Labels

Sometimes, a label associated with a media unit of a sequence will provide metadata that is redundant with respect to another label associated with a consecutive media unit of the 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 sequence. As an optimization, an application may omit redundant attributes and items, and may 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 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”.

By way of example, the labeling of audio essence is one expected use of sparse labeling. Labeling each media unit of an audio sequence, 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 sequence, and the labels for the remaining media units could all be virtual. This would be quite appropriate if the sequence 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.

Alternatively, labels could be provided at the start and then periodically throughout the audio sequence. 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 sequence 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.

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 out of scope for this document.

Annex A Example Fictitious Time Label, Item Specification, Schema, and Instance (Informative)

A.1 General

The normative structure of the extensible time label (TLX) is specified in section 6.2. This annex provides an example of a fictitious time label in section A.5, which is consistent with the example specification of a fictitious TLX item “TLXfoo”, provided by section A.3. 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 section A.4. Such schemas can support validation of time label instances.

A.2 Appearance of Conformance Language in Section A.3.2

When specifying an actual TLX Item, conformance terms are expected to appear in the Attributes subsection (e.g., A.3.2 for the fictitious item “TLXfoo”) that follows the Description section. For this informative example, the conformance term “*shall*” is necessary to properly illustrate how to specify the normative association between a TLX item name and its corresponding attributes while not rendering the example false and misleading. The conformance term “*shall*” is presented in section A.3.2 in italics to acknowledge that conformance terms are not permitted in informative portions of a standard. Despite the presence of conformance language here, this example is merely informative, whereas in actual item definitions, such conformance language, not typically in italics, would be actively normative.

A.3 TLXfoo (a Fictitious TLX Item)

A.3.1 Description

TLXfoo is a fictitious time label item that provides an example of item definitions presented as the entirety of A.3. The name of an item is provided as the name of a document section, as with A.3, though for non-fictitious items, such sections would be normative, not informative. This Description subsection A.3.1 identifies the purpose of the **TLFfoo** 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.3.2 Attributes

The attributes of **TLXfoo** *shall*[1] be as specified in Table 3.

Table 3 - TLXfoo attributes

Name	Type	Constraints	Required/Optional	Description
bar	string	length of the value is from 1 to 4 characters[2]	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 UTF8, some characters can require a single byte, while other characters can require more than one byte.

These 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 subsections could be provided to specify or describe other aspects of the item (e.g., processing).

A.4 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.3, 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-ficticious",
  "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
```

[1] See section A.2.

[2] Take care in specifying the constraints of strings. Strings in JSON are Unicode, and in TLX are constrained to be UTF8 encoded. Thus, the length of a string in characters can be different from its length in bytes.


```
      }
    }
" :  ],
  "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.5 Example fictitious time label instance as JSON [Informative]

The following JSON instance is compliant with the example fictitious schema of A.4.

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

Annex B Binary Representation of Time Labels (Informative)

B.1 UBJSON

One binary representation for JSON data can be provided using The Universal Binary JSON Specification, available at ubjson.org. Accordingly, UBJSON can be used for a binary representation of time labels.

UBJSON has multiple advantages: It provides bidirectional 1:1 transforms between standard JSON and the UBJSON binary representation. It offers more efficient processing and storage of data than the equivalent JSON form. It is implemented by a number of libraries supporting many popular programming languages.

UBJSON has an underlying type, length, data architecture, where types are represented by single-character, single-byte markers.

Applications using can choose JSON, UBJSON, or a different representation for time labels.

B.1.1 Example Instance as UBJSON

The example fictitious time label instance presented in A.5 translates to UBJSON as follows.

Section B.1.1.1 shows the UBJSON representation in block-notation, where square brackets enclose each single-character marker (type), length (as a string of numerals), and data (as a string, which might be numeric). UBJSON relies on a structural knowledge of JSON, making representational optimizations when possible. Section B.1.1.2 presents the binary UBJSON representation, which is a sequence of octets, each of which is shown here in hexadecimal.

B.1.1.1 Translation (using block-notation)

```
[{]
    [i][6][TLXfoo][{]
        [i][3][bar][S][i][6][три]
        [i][3][baz][i][5]
    [}]
[}]
```

B.1.1.2 Binary

```
7b 69 06 54 4c 58 66 6f 6f 7b 69 03 62 61 72 53
69 06 d1 82 d1 80 d0 b8 69 03 62 61 7a 69 05 7d
7d
```

Bibliography (Informative)

SMPTE ST 12-1:2014 Time and Control Code

SMPTE ST 258:2004 SMPTE Standard – For Television – Transfer of Edit Decision Lists

SMPTE ST 377-1:2019 Material Exchange Format (MXF) – File Format Specification

SMPTE ST 428-7:2010 Digital Cinema Distribution Master – Subtitle

SMPTE ST 429-7:2006 D-Cinema Packaging – Composition Playlist

SMPTE ST 2059-1:2015 Generation and Alignment of Interface Signals to the SMPTE Epoch

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>