STANDARDS AND INFORMATION DOCUMENTS

AES31-3-2008
(reaffirmed 2019)

AUDIO ES STANDARDS

AES standard on Network & file transfer of audio — Audio-file transfer and exchange Part 3: Simple project exchange

Users of this standard are encouraged to determine if they are using the latest printing incorporating all current amendments and editorial corrections. Information on the latest status, edition, and printing of a standard can be found at: http://www.aes.org/standards

AUDIO ENGINEERING SOCIETY, INC.
60 East 42nd Street, New York, New York 10165, US.

Documents
Document Sharing Hub

Full document details: AES 31-3 2019

The AES Standards Committee is the organization responsible for the standards program of the Audio Engineering Society. It publishes technical standards, information documents and technical reports. Working groups and task groups with a fully international membership are engaged in writing standards covering fields that include topics of specific relevance to professional audio. Membership of any AES standards working group is open to all individuals who are materially and directly affected by the documents that may be issued under the scope of that working group.

Complete information, including working group scopes and project status is available at http://www.aes.org/standards. Enquiries may be addressed to standards@aes.org

The AES Standards Committee is supported in part by those listed below who, as Standards Sustainers, make significant financial contribution to its operation.

Standards Sustainer Gold

NETFLIX, Focusrite, XFRM Inc. Transforming the World One Idea at a Time, Olson Sound Design

Standards Sustainer Silver

LECTROSONICS, 21ST CENTURY FOX, Audio precision, SENNHEISER, ADAMSON, THE TELOS ALLIANCE, BOSE, Eventide, fulcrum ACOUSTIC, COVELOZ

Standards Sustainer

AUDYSSEY, NTi AUDIO, MQA, audio-technica, LAWO, CLAIR, WEISS, METRIC HALO, HEA, U-SOUND, THAT Corporation, NBCUniversal, DYNAUDIO, WorldCast Systems, Fraunhofer IIS, Control 4, MERGING, RAVENNA, DSPECIALISTS, ARCHWAVE, DOLBY, HARMAN, KLIPPEL, NEUTRIK, dbx-tv, XPERI

This list is current as of 2017/7/25

Documents
Document Sharing Hub

AES31-3-2008
(reaffirmed 2013)

AES standard for network and file transfer of audio - Audio-file transfer and exchange - Part 3: Simple project interchange

Published by
Audio Engineering Society, Inc.
Copyright ©2008 by the Audio Engineering Society

Abstract

This standard provides a convention for expressing edit data in text form in a manner that enables simple and accurate computer parsing while retaining human readability. It also describes a method for expressing time-code information in character notation and simple automation for stereo & surround panning and audio gain.

An AES standard implies a consensus of those directly and materially affected by its scope and provisions and is intended as a guide to aid the manufacturer, the consumer, and the general public. The existence of an AES standard does not in any respect preclude anyone, whether or not he or she has approved the document, from manufacturing, marketing, purchasing, or using products, processes, or procedures not in agreement with the standard. Prior to approval, all parties were provided opportunities to comment or object to any provision. Approval does not assume any liability to any patent owner, nor does it assume any obligation whatever to parties adopting the standards document. This document is subject to periodic review and users are cautioned to obtain the latest edition. Recipients of this document are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

2014-01-27 printing

Documents
Document Sharing Hub

Full document details: AES 31-3 2019

Contents

0 Introduction **4**
0.1 Rationale for this standard 4
0.2 Conventions used in this standard 5
1 Scope **5**
2 Normative references **5**
3 Definitions **6**
4 Edit decision markup language (EDML) **6**
4.1 Description 6
4.2 Text specifications and character set 6
4.3 Keywords 7
4.4 EDML data types 8
4.5 EDML usage conventions 10
4.6 Required sections and reserved keywords 11
4.7 Column alignment example 12
5 Time-code character groups **13**
5.0 General 13
5.1 Time-code group 13
5.2 Sample group 16
6 Audio decision list (ADL) **18**
6.0 ADL version 18
6.1 List structure 18
6.2 Edit time markers 18
6.3 Sections 18
6.4 ADL event entries 26
7 Edit automation **34**
7.0 Introduction 34
7.1 Description 34
7.2 Fader section 34
7.3 Pan section 36
7.4 Mute section 39
7.5 Markers section 40
8 Reference lists **41**
9 Assignment of coding **42**
9.1 Resource locators, identifiers, and formats 42
9.2 Reserved locations 42
Annex A *(Normative)* Video frame rate to sampling-frequency ratios **43**
Annex B *(Informative)* Time-code character format **44**
Annex C *(Informative)* Section keyword summary **45**
Annex D *(Informative)* Informative references **46**
Annex E *(Informative)* Zone differences from UTC for some locations **47**
Annex F (Normative) Currently approved resource locator, identifier, and format references **48**
Annex G *(normative)* ADL Version **49**

Documents
Document Sharing Hub
Full document details: AES 31-3 2019

Foreword

[This foreword is not a part of *AES standard for network and file transfer of audio — Audio-file transfer and exchange — Part 3: Simple project interchange*, AES31-3-1999.]

In 1997-03, the SC-06-01 Working Group on Audio-File Transfer and Exchange of the SC-06 Subcommittee on Network and File Transfer of Audio replaced SC-02-08 to consider the topic of interchange of audio material in the form of computer file data.

This document is part of a proposed multi-part standard for a simple format for audio project interchange defined in project AES-X67. It describes a generalized means to interchange edit data in the form of human-readable text. It describes a means to communicate timing information to sample accuracy in the form of a text string. It describes a technique for communicating edit information between different types of computer platform using plain text.

The remaining projects at the time of issue of this printing, and their intended parts, are:

Project AES-X69: AES31-1 *Media for Interchange;*
Project AES-X66: AES31-2 *Audio File Format;*
Project AES-X68: AES31-4 *Object Oriented Project Interchange.*

Following the presentation of an earlier version of AES31-3 at the AES 105th Convention in San Francisco, CA, US, the draft was substantially revised by Task Group SC-06-01-C at Oxford, England, UK, in 1999-02. It was completed and issued with a call for comment following the working group meeting in conjunction with the AES 106th Convention in Munich, Germany, 1999-05.

At the time of public approval of the standard, the membership of the task group was A. Bell, D. Brenan, C. Broad, K. Brown, J. Bull, G. Dimino, J. Emmett, B. Fattorini, A. Faust, B. Harris, U. Henry, E. Mcdermid, O. Morgan, J. Nunn, M. Porter, I. Rodrigues, C. Sleight, and M. Yonge.

Mark Yonge, Chair
Brooks Harris, Vice-Chair
SC-06-01, 1999-06

Foreword to second edition

This revision of AES31-3 incorporates previous corrigenda, and includes a number of editorial updates and corrections based on experiences of practical implementation. It has also been extended to include support for higher sampling frequencies and simple gain and surround pan automation.

At the time of revision, the writing group included: J. Bull, R. Caine, A. Faust, C. Garcha, B. Harris, U. Henry, J. Palmer, M. Yonge.

Mark Yonge
Chair, SC-02-08

Documents
Document Sharing Hub
Full document details: AES 31-3 2019

AES standard for network and file transfer of audio - Audio-file transfer and exchange - Part 3: Simple project interchange

0 Introduction

0.1 Rationale for this standard

Time-code character format (TCF) is a modified version of the familiar ASCII time-code display. It supports sample-accurate audio editing while integrating easily with systems and processes requiring conventional time codes. It can carry additional information describing time-base derivation. This generalized specification describes features that may not be applicable in all implementations.

Audio sampling resolution provides sample-accurate positions using a video frame count plus audio sample count scheme. The sample count is appended to the time code together with a sampling frequency indicator. Supported sampling frequencies include the primary audio rates together with the pull-up and pull-down rates required for sampling frequency conversions. Sampling frequency is expressed relative to true seconds.

In addition, TCF provides a means for communicating information about the timing relationships between different source media.

Society of Motion Picture and Television Engineers (SMPTE), European Broadcasting Union (EBU), and film time codes indicate how the HH:MM:SS:FF portion of the time code will be interpreted (30 frames per time base for SMPTE, 25 for EBU, 24 for film).

Time base indicates the actual speed (true hertz) at which the time code and the picture material associated with it is resolved. This has two values, 1 Hz and 1/1,001 Hz.

Video fields and time-code mode indicate a) the video field (field one or two) for National Television Systems Committee (NTSC) or Phase Alternating Line (PAL), and b) the frame-code mode (drop frame, non-drop frame) if the time code is NTSC.

Film framing indicates how film frames are transferred to form the video field sequence.

These enhancements provide an underlying mechanism for a broad range of edit data exchange formats capable of tracking film/video/audio phase relationships and conversions throughout the production process. TCF's similarity to conventional time code means TCF can interchange with conventional time-code formats with little difficulty.

2014-01-27 printing
AVA Documents
Document Sharing Hub
Full document details: AES 31-3 2019

0.2 Conventions used in this standard

0.2.1 Decimal points

According to IEC directives, the comma is used in all text to indicate the decimal point. However, in the specified coding, including the examples shown, the full stop is used as in IEC programming language standards.

0.2.2 Data representation

In this standard, all coding and data representations are printed in an equally spaced font.

0.2.3 Non-printing ASCII characters

Non-printing characters are delimited by angle brackets, as in `<CR>` for carriage return.

1 Scope

This standard provides a convention for expressing edit data in text form in a manner that enables simple and accurate computer parsing while retaining human readability. It also describes a method for expressing time-code information in character notation. It supports common professional audio sampling frequencies, video frame rates, and film framing. This document addresses the core need of the AES31 series of standards in providing a simple but extensible system for passing audio material between systems.

2 Normative references

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

ISO/IEC 646:1991, *Information technology — ISO 7-bit coded character set for information interchange*. International Standards Organisation, Geneva, Switzerland.

ISO/IEC 11578, - *Information technology - Open Systems Interconnection - Remote Procedure Call (RPC) [Annex A: Universal Unique Identifier];* International Standards Organisation, Geneva, Switzerland.

ISO 8601, *Data elements and interchange formats - Information interchange - Representation of dates and times*; International Standards Organisation, Geneva, Switzerland.

AES31-2, *AES standard on network and file transfer of audio – Audio-file transfer and exchange – File format for transferring digital audio data between systems of different type and manufacture,* Audio Engineering Society, New York, NY., US.

2014-01-27 printing
Documents
Document Sharing Hub
Full document details: AES 31-3 2019

3 Definitions

3.1
audio decision list
ADL
interchange list format that also forms part of a document conforming to AES31-3

3.2
time base
nominal second having one of two values, 1,000 s and 1,001 s

NOTE In a system running at a frame rate of 30 frames per second, exactly one second of real time elapses during the scanning of 30 frames. An example of such a system is an 1125/60 high-definition system. In an NTSC television system running at a vertical field rate of 60/1,001 field/s (approximately 59,94 Hz), one second of NTSC time elapses during the scanning of 60 television fields or 30 television frames, as described in SMPTE 12M, sub-clause 4.1. Because of the difference in vertical scanning rates, the relationship between real time and NTSC time is that 1 s NTSC is equivalent to 1,001 s real time.

3.3
NTSC time
time base of 1,001 s

3.4
nondrop frame
uncompensated mode frame numbered 0 through 29 successively with no omissions

3.5
drop frame
frame compensated using NTSC time counting mode. See 5.1

3.6
ASCII
7-bit coded character set, per ISO 646, which describes the character encoding used by this standard

3.7
broadcast wave format
BWF
audio data file format (see AES31-2)

4 Edit decision markup language (EDML)

4.1 Description

EDML is a very simple markup language designed to accommodate the requirements of edit data exchange. A primary design objective is to maximize platform and transmission compatibility and to provide simple and explicit parsing. It comprises a set of rules for creating ASCII edit data exchange formats and components.

4.2 Text specifications and character set

EDML stores and displays data in 7-bit ASCII text form. To help assure compatibility across platforms and transmission links EDML is restricted to a limited character set. Active characters represent the information contents of EDML. White space delimits fields (columns) within records (rows). Carriage return followed by line feed `<CR><LF>` and form feed `<FF>` may be used in their usual manner to accommodate display formatting but are ignored by the EDML parser.

2014-01-27 printing
AVA Documents
Document Sharing Hub
Full document details: AES 31-3 2019

Active character set:

```
abcdefghijklmnopqrstuvwxyz

ABCDEFGHIJKLMNOPQRSTUVWXYZ

0123456789

!"#$%&'()*+,-./:;<=>?@[]^_`{|}~
```

Delimiters:

`<space>`, `<tab>`, carriage return `<CR>`, linefeed `<LF>`, formfeed `<FF>`, `Esc` `<ESC>`

4.3 Keywords

EDML uses special keywords to identify elements.

4.3.1 Sections

A computer system is required to construct and parse text lists. To simplify these tasks and to support modular software development, the list is partitioned into discrete sections, each with a specific purpose. Each section has an explicitly defined start and end using a conventional tag technique.

`<A_SECTION>` identifies the start of section A_SECTION

`</A_SECTION>` identifies the end of section A_SECTION

Sections may be nested to form hierarchical structures. Parsing applications that do not recognize a section tag should seek forward to the end of that section in order to skip it and move on to the next. The sequence of sections at any level of a hierarchy is arbitrary.

4.3.2 Entries

Within each section, each EDML compliant format defines keywords, which begin and delimit entries and define the contents of the entries.

The text defining entries and their contents shall not be case dependent.

Entry keywords are composed of a specified keyword surrounded by parentheses.

```
EXAMPLE:
(KEY_WORD)
```

An entry may comprise one or more data values, or fields, corresponding with the data types defined in 4.4.

```
EXAMPLE 1
(SINGLE_VALUE) 0001

EXAMPLE 2
(RECORD_FIELDS) 0001 00:00:00:00/0000 "Some String"
```

An underscore character (ASCII $5F_{16}$) shall be used in place of any field where valid data are unavailable.

EDML reserves certain entry keywords. See 4.6.

AVA Documents
Document Sharing Hub
Full document details: AES 31-3 2019

4.3.3 Delimiters

Statements, entries, and events within the list are delimited and identified by specific keywords.

Fields within entries shall be delimited by white space. White space is defined as one or more of `<space>`, `<tab>`, `<linefeed>`, `<formfeed>`, or `<carriage return>` in any combination.

White space shall be ignored immediately following a keyword.

Line terminators within entries for print formatting shall comprise the two white space characters `<CR><LF>`.

Embedded entities, such as URIs, shall use their own internally-defined delimiters which may differ from this specification.

4.3.4 Comments

Comments are general purpose and distinct from entries using remark keywords (`Rem`) and are not expected to be machine readable. The parser shall be prepared to ignore them.

Comments may be placed in the list using a leading slash-asterisk, `/*` and a terminating asterisk-slash `*/`.

Comments can be placed anywhere in an EDML except within other entries. Comments shall be restricted to the characters defined as part of the EDML character set.

```
/* you can put a comment anywhere except within a record */

/*
you can let it wrap
            if you want, but it shall be terminated with a star-slash */
```

4.4 EDML data types

4.4.1 Integer numbers

Negative numbers shall be preceded by a minus sign (ASCII $2D_{16}$). Positive numbers may be preceded by an optional plus sign (ASCII $2B_{16}$).

Leading zeros may be used for visual formatting.

4.4.2 Real numbers

The EDML compliant format specifies the range of a real number in its field specification.

The decimal point may float.

Negative numbers shall be preceded by a minus sign (ASCII $2D_{16}$). Positive numbers may be preceded by an optional plus sign (ASCII $2B_{16}$).

EXAMPLE 1
`1.12345` provides for a number with six significant figures

EXAMPLE 2
`123.9876` provides for a number with seven significant figures

Leading zeros may be used for visual formatting.

AVA Documents
Document Sharing Hub
Full document details: AES 31-3 2019

4.4.3 Strings

Literal strings shall be enclosed in double quotes.

```
"String"
```

Unless specified otherwise, string lengths are limited to 255 printable characters that comply with 4.2.

Any double-quote character (ASCII 22_{16}) within a string shall be preceded by an escape character (ASCII $1B_{16}$). For the purposes of documenting this standard, the escape character is shown as `<ESC>`.

```
"She said<ESC>"Hello<ESC>""
```

NOTE because strings imported from other applications may be of arbitrary size, may include double-quote characters that require an additional escape character, and may include non-printing characters, the total string length could exceed the 255 printable characters specified here. The prudent implementor will take care to accommodate the total number of characters in the string, or implement a suitable truncation policy.

4.4.4 Variable-length binary data

Binary data shall be UU encoded for protected transmission as ASCII symbols.

The EDML variable-length binary data type uses a special keyword construction beginning with a defined keyword followed by a data length, in bytes (including UU header), as an EDML number (decimal integer) followed by a single white space followed immediately by the binary data, including any header data required by the encoder.

EXAMPLE
```
(KEYWORD)46 ;/[2=MT!G*,H,'\!@C5/H^&25FGNV'8SA4K58>G#N7H+9
```

To keep the binary data length exact, no white space shall be inserted between the keyword and the data length and precisely one white space character shall be inserted between the data length and the first byte of binary data. This is a stricter requirement than is assumed elsewhere in EDML.

4.4.5 Boolean

The EDML Boolean data type consists of two words, `TRUE` and `FALSE`, with no surrounding punctuation.

EXAMPLE 1
TRUE

EXAMPLE 2
FALSE

4.4.6 UUID

The format of the Universal Unique Identifier (UUID) shall comply with ISO/IEC 11578, annex A, using the string representation. Each field of the UUID is treated as an integer and has its value presented as a zero-filled hexadecimal digit string with the most significant digit first. The hexadecimal values a to f inclusive shall be output as lower case characters, and shall be case insensitive on input.

```
00000000-0000-0000-0000-000000000000
```

This example shows the only special case known as a “null UUID” which is clearly not unique. The following is an example of the string representation of a UUID:

```
2fac1234-31f8-11b4-a222-08002b34c003
```


AVA Documents
Document Sharing Hub
Full document details: AES 31-3 2019

4.5 EDML usage conventions

4.5.1 Version

Version shall be represented as a string with up to 15 characters including point separators having the format `MJ.MI(.BG)(.ST)(.RV)`, where:

`MJ`	Major version number	Required
`MI`	Minor release number	Required
`BG`	Bug revision (point-release) number (where appropriate)	Optional
`ST`	Stage: developmental, alpha, beta, final (where appropriate)	Optional
`RV`	Revision of non-released code (where appropriate)	Optional

EXAMPLE 1
`01.01.00.00.00`

EXAMPLE 2
`01.01`

If data are unavailable, an underscore character shall indicate an empty field.

4.5.2 Date, time of day, and zone

Coordinated Universal Time (UTC) is the time scale maintained by the International Bureau of Weights and Measures (BIPM), with assistance from the International Earth Rotation Service (IERS), which forms the basis of a coordinated dissemination of standard frequencies and time signals. Its representation is described in ISO 8601.

4.5.2.1 Date

The date shall be represented as the year, month, and day of the Gregorian calendar. Hyphens (`-`) shall be used as separators within the date expression. All components of the date are required. In the case of unavailable data, the default value shall be the origin of the modified Julian date (MJD), namely, 1858-11-17.

Format: `CCYY-MM-DD`

`CCYY` = 4 characters for century and year — required — shall contain a value 0000 - 9999
`-` = 1 character — required
`MM` = 2 characters for month — required — shall contain a value 1 to 12
`-` = 1 character — required
`DD` = 2 characters for day of month — required — shall contain a value 1 to 31

4.5.2.2 Time of day

The time of day may be appended to the date. If given, it shall be represented as hours, minutes, and seconds. The character `T` shall be used as time designator to indicate the start of the representation of the time of day and shall immediately follow the date. Colons (`:`) shall be used as separators within the time of day expression.

If time is given, all components of the time component shall be present.

Format: `Thh:mm:ss`

`T` = 1 character — required if time given
`hh` (hour) = 2 characters — shall contain a value 00 to 23 if time given
`:` = 1 character — required if time given
`mm` (minute) = 2 characters — shall contain a value 00 to 59 if time given
`:` = 1 character — required if time given
`ss` (second) = 2 characters — shall contain a value 00 to 59 if time given

AVA Documents
Document Sharing Hub
Full document details: AES 31-3 2019

4.5.2.3 Zone

The relation of local time of day to Coordinated Universal Time (UTC) may be indicated by the zone designator, as shown in annex E.

The local time of day may be expressed directly as UTC. In this case the date and time of day shall be followed immediately, without spaces, by the zone designator `Z`.

The local time of day may be expressed as the difference between local time and UTC. In this case the difference shall be expressed as positive hours and minutes (for example, with a leading plus sign (`+`)) if the local time is ahead of UTC and as negative hours and minutes (for example, with a leading minus sign (`-`)) if it is behind UTC. The colon shall be used as separator between hours and minutes.

Format: `Z` or `+hh:mm` or `-hh:mm`

If zone is expressed as UTC:
`Z` = 1 character — required if zone given

If zone is expressed as difference from UTC:
`+` or `-` = 1 character — required if zone given
`hh` = 2 characters — shall contain a value 00 to 23 if zone given
`:` = 1 character — required if zone given
`mm` = 2 characters — shall contain a value 00 to 59 if zone given

> NOTE It is strongly encouraged to use the complete representation, including the zone designator, so the date and local time of day can be normalized to UTC. The difference between local time and UTC is the preferred method.

EXAMPLE 1: Date, time of day, and zone (the date, time of day, and zone in Geneva, Greenwich, and New York at the UTC turn of the century):

`2000-01-01T00:00:00Z`	(UTC)
`2000-01-01T00:00:00+00:00`	(Greenwich)
`2000-01-01T01:00:00+01:00`	(Geneva)
`1999-12-31T19:00:00-05:00`	(New York — daylight saving time is not in effect)

EXAMPLE 2: Date and local time of day only:

```
1985-03-04T12:00:00
```

EXAMPLE 3: Date only:

`2001-03-01`	
`1858-11-17`	(Default)

4.5.3 Time code

Time-code strings shall conform to the rules of time-code character format (TCF) to provide sample-accurate position and synchronization information. See clause 5.

4.6 Required sections and reserved keywords

EDML reserves several keywords and requires the use of several of them.

AVA Documents
Document Sharing Hub
Full document details: AES 31-3 2019

4.6.1 List-format declaration

Edit data exchange list formats employing EDML shall be contained entirely within a section defined by specifically named keyword tags. This identifies the list-format type and indicates the beginning and end of the list-format instance.

```
EXAMPLE
<ADL>
...
</ADL>
```

4.6.2 Header sections

All interchange lists shall contain one or more header sections to define parameters relating to the `VERSION`, `PROJECT`, `SYSTEM`, and `SEQUENCE` of the particular list.

```
EXAMPLE
<ADL>
      <VERSION>
      </VERSION>
</ADL>
```

4.6.3 List contents

There shall be one or more sections containing entries to define the payload contents of the list as defined in that list format. See clause 6.

4.7 Column alignment example

It is anticipated that most EDML lists could be displayed or printed with columns aligned where possible for legibility, but this alignment is not a requirement and will be impossible in some situations.

For instance, the CUT event of ADL has eight data fields, and the following three examples are legitimate for the parser and have identical meaning:

EXAMPLE 1:

```
(E)0005 (C) 0010 _ 2 2 00:00:00.18/1601 01:00:00.00/0000 01:00:01.08/1601 E
```

EXAMPLE 2:

```
(E)0005 (C) 0010 _ 2 2
             00:00:00.18/1601 01:00:00.00/0000 01:00:01.08/1601 E
```

EXAMPLE 3:

```
(E)0005
(C)0010
_
2
2
00:00:00.18/1601
01:00:00.00/0000
01:00:01.08/1601 E
```

2014-01-27 printing
Documents
Document Sharing Hub
Full document details: AES 31-3 2019

5 Time-code character groups

5.0 General

Time-code character format (TCF) expresses sample counts as time-code values in two groups of characters, time-code group and sample group. The numerical relationship between various frame rates and audio sampling frequencies is shown in annex A.

Sample count shall be zero-based such that midnight is represented by the value zero. Sample count then increments at the relevant sampling frequency. This convention is compatible with Broadcast Wave Format (BWF) file implementations.

The time-code group shall be required and provides frame and field accurate video indexing over an indicated 24-hour period. The optional sample group may be appended to the time-code group to provide audio sample resolution.

5.1 Time-code group

The time-code group identifies each video frame by a unique and complete address consisting of an hour, minute, second, and frame number. It shall be made up of four two-digit pairs representing hours (`HH`), minutes (`MM`), seconds (`SS`), and frames (`FF`), separated by indicators (`i`), in the form `HHiMMiSSiFF`. The hours, minutes, and seconds follow the ascending progression of a 24-hour clock beginning with 0 hours, 0 minutes, and 0 seconds to 23 hours, 59 minutes, and 59 seconds.

The frames shall be numbered successively from 0 to *n* as indicated by the frame count and time-base indicator (see 5.1.1). The count may be modified by the drop-frame count mode.

> NOTE Because the vertical field rate of an NTSC television signal is 60/1,001 fields per second (approximately 59,94 Hz) (see SMPTE 12M, sub-clause 4.2.2), straightforward counting at 30 frames per second will yield an error of approximately +108 frames (+3,6 s real time) in one hour of running time. To minimize the NTSC time error, the first two frame numbers (00 and 01) shall be omitted from the count at the start of each minute except minutes 00, 10, 20, 30, 40, and 50. When drop-frame compensation is applied to an NTSC television time code, the total error accumulated after one hour is reduced to 3,6 ms. The total error accumulated over a 24-hour period is 86 ms.

The indicators between the `HH MM SS FF` pairs specify frame count and time base; film framing; and video field and time-code type, respectively. They may be modified by the drop-frame counting mode.

The time-code group shall be required.

5.1.1 Frame count and time-base indicator

The first separator in the time-code group shall be the frame count and time-base indicator. It specifies the actual frame rate of the picture and the time code by combining two values, the frame count and the time base.

5.1.1.1 Frame count

The frame count specifies how the frames (`FF`) portion of the time-code group is incremented. The count may be modified if the drop-frame count mode is in effect. Table 1 shows the specified frame counts.

Table 1 — Frame count

Frames	Count
30	00 to 29
25	00 to 24
24	00 to 23

AVA Documents
Document Sharing Hub
Full document details: AES 31-3 2019

5.1.1.2 Frame-count division

The time base indicates the value of the nominal second into which the frame count is divided.

5.1.1.3 Coding of frame count and time-base indicator

The combination of frame count and time base shall be encoded into a single ASCII character as the frame count and time-base indicator. Table 2 shows the character for each condition followed by the ASCII character's decimal value in parentheses.

Table 2 — Frame count and time-base indicator coding

Frame count	Time base		
	Unknown	1,000	1,001
30	? (063)	| (124)	: (058)
25	! (033)	. (046)	/ (047)
24	# (035)	= (061)	- (045)

5.1.2 Film frame indicator

The second separator in the time-code group (character position 6) specifies the sequencing of film frames recorded into video.

5.1.2.1 Coding of film frame indicator

NTSC film framing (2–3 pull-down) shall be coded with the film frame in each of the video fields having the sequence `AaBbyCcDdz`. Table 3 shows the coding for each film frame in each video field in the NTSC 2–3 sequence.

Table 3 — NTSC 2–3 sequence

Video field	Video frame	Film	Indicator
1	1	1A	A (065)
2	1	1A	a (097)
3	2	2B	B (066)
4	2	2B	b (098)
5	3	2B	y (121)
6	3	3C	C (067)
7	4	3C	c (099)
8	4	4D	D (068)
9	5	4D	d (100)
10	5	4D	z (122)
One-to-one			Z (090)
Unknown			X (088)
Not applicable			Repeat frame count and time-base indicator.

2014-01-27 printing
AVA Documents
Document Sharing Hub
Full document details: AES 31-3 2019