

STANDARDS AND INFORMATION DOCUMENTS

Call for comment on DRAFT AES standard for network and file transport of audio - XML Implementation of Audio Decision Lists

This document was developed by a writing group of the Audio Engineering Society Standards Committee (AESSC) and has been prepared for comment according to AES policies and procedures. It has been brought to the attention of International Electrotechnical Commission Technical Committee 100. Existing international standards relating to the subject of this document were used and referenced throughout its development.

Address comments by E-mail to standards@aes.org, or by mail to the AESSC Secretariat, Audio Engineering Society, 697 Third Avenue, Suite 405, New York, NY 10017, US. **Only comments so addressed will be considered.** E-mail is preferred. **Comments that suggest changes must include proposed wording.** Comments shall be restricted to this document only. Send comments to other documents separately. 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.

This document will be approved by the AES after any adverse comment received within **six weeks** of the publication of this call on <http://www.aes.org/standards/comments/>, **2024-05-24**, has been resolved. Any person receiving this call first through the *JAES* distribution may inform the Secretariat immediately of an intention to comment within a month of this distribution.

Because this document is a draft and is subject to change, no portion of it shall be quoted in any publication without the written permission of the AES, and all published references to it must include a prominent warning that the draft will be changed and must not be used as a standard.

[This page intentionally blank]

DRAFT

AES standard for network and file transport of audio - XML Implementation of Audio Decision Lists

Abstract

This document provides a syntax mapping for AES31-3 Edit Decision Markup Language (EDML) to XML Schema Language. This facilitates the expansion of the Audio Decision List format to include non-ASCII characters and updates the format facilitating its implementation using standard XML parsers and tools. It also supports multi-byte character sets for human-readable metadata in all territories worldwide. This document includes both an XML schema definition and an XSLT implementation capable of transforming a conforming XML instance document back to EDML.

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. Attention is drawn to the possibility that some of the elements of this AES standard or information document may be the subject of patent rights. AES shall not be held responsible for identifying any or all such patents. Approval does not assume any liability to any patent owner, nor does it assume any obligation whatever to parties adopting the standards document. 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. This document is subject to periodic review and users are cautioned to obtain the latest edition.

Audio Engineering Society Inc. 697 Third Avenue, Suite 405, New York, NY 10017, US

www.aes.org/standards standards@aes.org

Contents

0 Intro	4
0.1 Introduction	4
0.2 Documentation conventions.....	4
1 Scope	4
2 Normative references	4
3 Terms, definitions and abbreviations	5
4 EDML mapping to AES31-3 Extensible Markup Language (XML) schema	5
4.1 Description	5
4.1 Schema	5
4.3 Stylesheet.....	5
5 ADL elements mappings	5
5.1 General.....	5
5.2 ADL root	5
5.3 Header sections	6
5.4 Source index	8
5.5 ADL events.....	9
5.6 Edit automation	12
5.7 Reference lists	14
Annex A (Normative): Schema	15
Annex B (normative): XSLT Stylesheet	23
Annex C (Informative) - Informative references	36

Foreword

This foreword is not part of the AES31-4-2015 *AES standard for network and file transport of audio - XML Implementation of Audio Decision Lists*.

AES31-3 was published in 1999 to provide a long-term alternative to proliferating proprietary formats. It provided a convention for expressing edit data in text form in a manner that enabled simple and accurate computer parsing while retaining human readability. It also described a method for expressing time-code information in character notation and simple automation for stereo & surround panning and audio gain. These edit documents were known as Audio Decision Lists (ADL) and used an Edit Decision Markup Language (EDML).

The subsequent growth of XML offers a similar markup facility but with better availability of software tools for faster implementations. XML also offers support for multi-byte characters in human-readable metadata instead of the plain ASCII of EDML, opening implementation to a world-wide user community.

This document was developed from a proposal written by David Ackerman with the assistance of Bruce Gordon at Harvard Library. It was developed in AESSC working group SC-07-01 under project AES-X214.

Chris Chambers
Co-chair, working group SC-07-01 on Audio Metadata
2015-12-16

Foreword to 2022 edition

The SMPTE Universal Label identified in Annex F.2 was requested of SMPTE on 2019-08-15 as a Leaf in the metadata dictionary for 'Audio edit decision list (ADL) defined in AES31-3'.

At the same time the AES requested a Node in the metadata dictionary for 'File exchange metadata defined in AES standards'. This means that further file-exchange labels can be generated more easily in the future.

Tormod Vervaegan, Kaylie Ackerman
Co-Chairs, working group SC-03-07 on Audio Metadata
2022-10-14

Note on normative language

In AES standards documents, sentences containing the word “shall” are requirements for compliance with the document. Sentences containing the verb “should” are strong suggestions (recommendations). Sentences giving permission use the verb “may”. Sentences expressing a possibility use the verb “can”.

DRAFT

AES standard for network and file transport of audio - XML Implementation of Audio Decision Lists

0 Intro

0.1 Introduction

This standard sets out a mapping to express EDML as XML as defined by a new XML Schema that closely mirrors the original structure of an Audio Decision List documents as defined in AES31-3. EDML was developed prior to wide adoption of XML as a standard data carrier. As a result it has features that are XML-like as well as features that are idiosyncratic. Perhaps it's greatest limitation is the restriction that requires all data to be represented in ASCII. This limitation makes it difficult to use in regions that rely on multi-byte character encodings to carry project metadata. By updating the standard to use XML this limitation is overcome. Doing so also opens up a set of programming tools for working with the resulting instance documents that may provide easier/faster implementation and better document validation.

Backwards compatibility to EDML may be maintained through XSLT. An example XSLT document is included for reference.

0.2 Patents

Attention is drawn to the possibility that some of the elements of this AES document may be the subject of patent rights not identified herein. The AES shall not be held responsible for identifying any or all such patent rights.

0.3 Documentation conventions

A Courier typeface may be used to identify computer listing examples to distinguish them from regular text.

1 Scope

This document specifies the mapping between EDML elements, described in AES31-3, and their XML counterparts described in the included schema. The complete schema is included along with an XSLT document which provides a reference transform back to EDML for backwards compatibility.

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

AES31-3-2021: *AES standard for network and file transfer of audio - Audio-file transfer and exchange - Part 3: Simple project interchange* Audio Engineering Society, New York, NY., US.

XML Schema Definition Language (XSD) 1.1 Part 1: Structures. World Wide Web Consortium (W3C), 2012-04-05

XML Schema Definition Language (XSD) 1.1 Part 2: Datatypes, World Wide Web Consortium (W3C), 2012-04-05

3 Terms, definitions and abbreviations

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

3.1

Audio Decision List

ADL

interchange list format that also forms part of a document conforming to AES31-3.

3.2

EDML

Edit Decision Markup Language

simple markup language designed to accommodate the requirements of edit data exchange. See AES31-3.

4 EDML mapping to AES31-3 Extensible Markup Language (XML) schema

4.1 Description

In the mappings set out in clause 5, a strict one-to-one mapping between AES31-4-XML and EDML has been maintained, with one addition. As noted in the schema definition, some XML elements have an ID attribute added for potential use in applications working with the instance documents. As required in XML Schema language, these IDs must be unique.

When transforming back to EDML, XML ID attributes shall be discarded.

4.1 Schema

The schema listed in annex A shall be used to produce an ADL-XML document.

This schema is also available at

<http://www.aes.org/standards/schemas/aes31-4-151112-schema.xsl>

4.3 Stylesheet

Where used, the XSLT stylesheet listed in annex B shall be used to convert an AES31-4-XML document to an AES31-3-compliant ADL.

This stylesheet is also available at

<http://www.aes.org/standards/schemas/aes31-4-151112-stylesheet.xsd>

5 ADL elements mappings

5.1 General

The element definitions shall be as defined in AES31-3.

The following tables provide the mapping from EDML section elements and keywords to their XML counterparts in the included schema.

5.2 ADL root

Table 1 - adl (root)

EDML Section	XML
<ADL>	adl (root element)
	adl/@id

5.3 Header sections

Table 2 - Header Sections

EDML Section	XML
<VERSION>	adl/version
<PROJECT>	adl/project
<SYSTEM>	adl/system
<SEQUENCE>	adl/sequence
<TRACKLIST>	adl/trackList
<SOURCEINDEX>	adl/sourceIndex
<EVENT_LIST>	adl/eventList
<FADER_LIST>	adl/faderList
<PAN_LIST>	adl/panList
<MUTE_LIST>	adl/muteList
<MARKER_LIST>	adl/markerList
<REF_LIST>	adl/refList

Table 3 - ADL version header

EDML Keywords	XML
(ADL_ID)	adl/version/adlId
(ADL_UID)	adl/version/adlUid
(VER_ADL_VERSION)	adl/version/verAdlVersion
(VER_CREATOR)	adl/version/verCreator
(VER_CRTR)	adl/version/verCrtr
	adl/version/@id

Table 4 - Project header

EDML Keywords	XML
(PROJ_TITLE)	adl/project/projTitle
(PROJ_ORIGINATOR)	adl/project/projOriginator
(PROJ_CREATE_DATE)	adl/project/projCreateDate
(PROJ_NOTES)	adl/project/projNotes
(PROJ_CLIENT_DATA)	adl/project/projClientData
	adl/project/@id

Table 5 - System header

EDML Keywords	XML
(SYS_SRC_OFFSET)	adl/system/sysSrcOffset
(SYS_BIT_DEPTH)	adl/system/sysBitDepth
(SYS_AUD_CODEC)	adl/system/sysAudCodec
(SYS_XFADE_LEN)	adl/system/sysXfadeLen
(SYS_GAIN)	adl/system/sysGain
	adl/system/@id

Table 6 - Sequence header

EDML Keywords	XML
(SEQ_TITLE)	adl/sequence/seqTitle
(SEQ_DESCRIPTOR)	adl/sequence/seqDescriptor
(SEQ_SAMPLE_RATE)	adl/sequence/seqSampleRate
(SEQ_SAMPLE_RATE_FACTOR)	adl/sequence/seqSampleRateFactor <i>VALUE=1 2 4</i>
(SEQ_FRAME_RATE)	adl/sequence/seqFrameRate
(SEQ_ADL_LEVEL)	adl/sequence/seqAdlLevel
(SEQ_CLEAN)	adl/sequence/seqClean
(SEQ_SORT)	adl/sequence/seqSort
(SEQ_MULTICHAN)	adl/sequence/seqMultichan
(SEQ_DEST_START)	adl/sequence/seqDestStart
	adl/sequence/@id

Table 7 - Track-list header

EDML Keywords	XML
(Track)	adl/trackList/track
Track#	adl/trackList/track/@trackNumber
TrackName	adl/trackList/track/@trackName
	adl/trackList/track/@id
	adl/trackList/@id

5.4 Source index

Table 8 - Source index header

EDML Keywords	XML
(Index)	adl/sourceIndex/index
	adl/sourceIndex/@id

Table 9 - Source index structure - Index

EDML Keywords	XML
Index Number	adl/sourceIndex/index/@indexNumber
(F)	adl/sourceIndex/index/filePathSource
(U)	adl/sourceIndex/index/uriSource
(T)	adl/sourceIndex/indextapeSource
	adl/sourceIndex/index/@id
	adl/sourceIndex/@id

Table 10 - Source index structure - Index - File path source

EDML Keywords	XML
(F)	adl/sourceIndex/index/filePathSource/@keyLetter
<i>File_Path</i>	adl/sourceIndex/index/filePathSource/filePath
UID	adl/sourceIndex/index/filePathSource/uid
File_In	adl/sourceIndex/index/filePathSource/fileIn
File_Len	adl/sourceIndex/index/filePathSource/fileLen
Descr	adl/sourceIndex/index/filePathSource/descr
Code	adl/sourceIndex/index/filePathSource/code <i>VALUE=N X A</i>
	adl/sourceIndex/index/filePathSource/@id

Table 11 - Source index structure - index - URI source

EDML Keywords	XML
(U)	adl/sourceIndex/index/uriSource/@keyLetter
<i>URI</i>	adl/sourceIndex/index/uriSource/fileLocator
UID	adl/sourceIndex/index/uriSource/uid
File_In	adl/sourceIndex/index/uriSource/fileIn
File_Len	adl/sourceIndex/index/uriSource/fileLen
Descr	adl/sourceIndex/index/uriSource/descr
Code	adl/sourceIndex/index/uriSource/code <i>VALUE=N X A</i>
	adl/sourceIndex/index/uriSource/@id

Table 12 - Source index structure - index - Tape source

EDML Keywords	XML
(T)	adl/sourceIndex/index/tapeSource/@keyLetter
Tapename	adl/sourceIndex/index/tapeSource/tapeName
Tape_Orig	adl/sourceIndex/index/tapeSource/tapeOrig
File_Orig	adl/sourceIndex/index/tapeSource/fileOrig
T_Chan	adl/sourceIndex/index/tapeSource/tChan
F_Chan	adl/sourceIndex/index/tapeSource/fChan
	adl/sourceIndex/index/tapeSource/@id

5.5 ADL events

Table 13 - ADL event section - Edit event - Cut (default)

EDML Keywords	XML
EventType (Cut)	adl/eventList/eventEntry/cut
EntryNo	adl/eventList/eventEntry/@entryNumber
SrcType	adl/eventList/eventEntry/cut/srcType
SrcIndx	adl/eventList/eventEntry/cut/srcIndex
SrcChan	adl/eventList/eventEntry/cut/srcChannel
DestChan	a adl/eventList/eventEntry/cut/destChannel
SrcIn	adl/eventList/eventEntry/cut/srcIn
DestIn	adl/eventList/eventEntry/cut/destIn
DestOut	adl/eventList/eventEntry/cut/destOut
Status	adl/eventList/eventEntry/cut/statusCode <i>VALUE=R E D X M</i>
	adl/eventList/eventEntry/cut/@id
	adl/eventList/eventEntry/@id
	adl/eventList/@id

Table 14 - ADL event section - Edit events, other types - Silence

EDML Keywords	XML
EventType (Silence)	adl/eventList/eventEntry/silence
EntryNo	adl/eventList/eventEntry/@entryNumber
DestChan	adl/eventList/eventEntry/silence/destChannel
DestIn	adl/eventList/eventEntry/silence/destIn
DestOut	adl/eventList/eventEntry/silence/destOut
Status	adl/eventList/eventEntry/silence/statusCode <i>VALUE=R E D X M</i>
	adl/eventList/eventEntry/silence/@id
	adl/eventList/eventEntry/@id

Table 15 - ADL event section - Edit events, other types - Auxiliary

EDML Keywords	XML
EventType (Aux)	adl/eventList/eventEntry/aux
EntryNo	adl/eventList/eventEntry/@entryNumber
DestChan	adl/eventList/eventEntry/aux/destChannel
DestIn	adl/eventList/eventEntry/aux/destIn
DestOut	adl/eventList/eventEntry/aux/destOut
Status	adl/eventList/eventEntry/aux/statusCode <i>VALUE=R E D X M</i>
	adl/eventList/eventEntry/aux/@id
	adl/eventList/eventEntry/@id

Table 16 - ADL event section, other types - Video

EDML Keywords	XML
EventType (Vid)	adl/eventList/eventEntry/vid
EntryNo	adl/eventList/eventEntry/@entryNumber
SrcType	adl/eventList/eventEntry/vid/srcType
SrcIndx	adl/eventList/eventEntry/vid/srcIndex
SrcIn	adl/eventList/eventEntry/vid/srcIn
DestIn	adl/eventList/eventEntry/vid/destIn
DestOut	adl/eventList/eventEntry/vid/destOut
Status	adl/eventList/eventEntry/vid/statusCode <i>VALUE=R E D X M</i>
	adl/eventList/eventEntry/vid/@id
	adl/eventList/eventEntry/@id

Table 17 - ADL event section - Event modifiers - Alternate source modifier

EDML Keywords	XML
EventType (Alt)	adl/eventList/eventEntry/altSource
EntryNo	adl/eventList/eventEntry/@entryNumber
SrcType	adl/eventList/eventEntry/altSource/srcType
SrcIndx	adl/eventList/eventEntry/altSource/srcIndex
SrcChan	adl/eventList/eventEntry/altSource/srcChannel
SrcIn	adl/eventList/eventEntry/altSource/srcIn
	adl/eventList/eventEntry/altSource/@id
	adl/eventList/eventEntry/@id

Table 18 - ADL event section - Event modifiers - Fade in

EDML Keywords	XML
(InFade)	adl/eventList/eventEntry/fadeIn
Duration	adl/eventList/eventEntry/fadeIn/duration (default=0)
Shape	adl/eventList/eventEntry/fadeIn/shape <i>VALUE=LIN CURVE</i>
Curve_A	adl/eventList/eventEntry/fadeIn/curveA
Curve_B	adl/eventList/eventEntry/fadeIn/curveB
Curve_C	adl/eventList/eventEntry/fadeIn/curveC
SrcType	adl/eventList/eventEntry/fadeIn/srcType
SrcIndx	adl/eventList/eventEntry/fadeIn/srcIndex
SrcIn	adl/eventList/eventEntry/fadeIn/srcIn
	adl/eventList/eventEntry/fadeIn/@id
	adl/eventList/eventEntry/@id

Table 19 - ADL event section - Event modifiers - Fade out

EDML Keywords	XML
(OutFade)	adl/eventList/eventEntry/fadeOut
Duration	adl/eventList/eventEntry/fadeOut/duration (default=0)
Shape	adl/eventList/eventEntry/fadeOut/shape <i>VALUE=LIN CURVE</i>
Curve_A	adl/eventList/eventEntry/fadeOut/curveA
Curve_B	adl/eventList/eventEntry/fadeOut/curveB
Curve_C	adl/eventList/eventEntry/fadeOut/curveC
SrcType	adl/eventList/eventEntry/fadeOut/srcType
SrcIndx	adl/eventList/eventEntry/fadeOut/srcIndex
SrcIn	adl/eventList/eventEntry/fadeOut/srcIn
	adl/eventList/eventEntry/fadeOut/@id
	adl/eventList/eventEntry/@id

Table 20 - ADL event section - Event modifiers - Cross fade

EDML Keywords	XML
(Xfade)	adl/eventList/eventEntry/crossFade
PreviousClip	adl/eventList/eventEntry/crossFade/previousClip
DestIn	adl/eventList/eventEntry/crossFade/destIn
DestOut	adl/eventList/eventEntry/crossFade/destOut
SrcType	adl/eventList/eventEntry/crossFade/srcType

SrcIndx	adl/eventList/eventEntry/crossFade/srcIndex
SrcIn	adl/eventList/eventEntry/crossFade/srcIn
	adl/eventList/eventEntry/crossFade/@id
	adl/eventList/eventEntry/@id

Table 21 - ADL event section - Event modifiers - Signal modifiers - Gain Modifier

EDML Keywords	XML
(Gain)	adl/eventList/eventEntry/gain
Channel	adl/eventList/eventEntry/gain/channel
Gain	adl/eventList/eventEntry/gain/gainValue
	adl/eventList/eventEntry/gain/@id
	adl/eventList/eventEntry/@id

Table 22 - ADL event section - Remarks

EDML Keywords	XML
(Rem)	adl/eventList/eventEntry/remark
RemType	adl/eventList/eventEntry/remark/remType <i>VALUE=NAME SOURCE DESC USER</i>
Remark	adl/eventList/eventEntry/remark/remValue
	adl/eventList/eventEntry/remark/@id
	adl/eventList/eventEntry/@id

5.6 Edit automation

Table 23 - ADL - Fader section - Fader-point entries

EDML Keywords	XML
(FP)	adl/faderList/faderPointEntry
Destination Track	adl/faderList/faderPointEntry/destinationTrack
DestIN	adl/faderList/faderPointEntry/destIn
Fader Value	adl/faderList/faderPointEntry/faderValue
	adl/faderList/faderPointEntry/@id
	adl/faderList/@id

Table 24 - ADL Pan section - Pan-point entries

EDML Keywords	XML
(PP)	adl/panList/panPointEntry
DestinationTrack	adl/panList/panPointEntry/destinationTrack
DestIN	adl/panList/panPointEntry/destIn
LeftRightPosition	adl/panList/panPointEntry/leftRightPosition
FrontRearPosition	adl/panList/panPointEntry/frontRearPosition
	adl/panList/panPointEntry/@id
	adl/panList/@id

Table 25 - ADL - Mute section - Mute-point entries

EDML Keywords	XML
(MP)	adl/muteList/mutePointEntry
DestinationTrack	adl/muteList/mutePointEntry/destinationTrack
DestIN	adl/muteList/mutePointEntry/destIn
Mute	adl/muteList/mutePointEntry/muteValue <i>VALUE=M U (audio muted audio unmuted)</i>
	adl/muteList/mutePointEntry/@id
	adl/muteList/@id

Table 26 - ADL - Marker section - Marker-point entries

EDML Keywords	XML
(MK)	adl/markerList/markerPointEntry/markerKeyword <i>VALUE=MK</i>
(MK-PQ-START)	adl/markerList/markerPointEntry/markerKeyword <i>VALUE=MK-PQ-START</i>
(MK-PQ-END)	adl/markerList/markerPointEntry/ markerKeyword <i>VALUE=MK-PQ-END</i>
(MK-PQ-INDEX)	adl/markerList/markerPointEntry/ markerKeyword <i>VALUE=MK-PQ-INDEX</i>
DestinationTrack	adl/markerList/markerPointEntry/destinationTrack
DestIN	adl/markerList/markerPointEntry/destIn
DestOUT	adl/markerList/markerPointEntry/destOut
MarkerName	adl/markerList/markerPointEntry/markerName
	adl/markerList/markerPointEntry/@id
	adl/markerList/@id

5.7 Reference lists

Table 27 - ADL - Reference lists

EDML Section	XML
<PROJECT>	adl/refList/ref/project
<SYSTEM>	adl/refList/ref/system
<SEQUENCE>	adl/refList/ref/sequence
<EVENT_LIST>	adl/refList/ref/eventList
	adl/refList/ref/@id
	adl/refList/@id

Annex A (Normative): Schema

```

<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema targetNamespace="http://www.aes.org/aes31"
  xmlns="http://www.aes.org/aes31"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:tcf="http://www.aes.org/tcf"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified">
  <xsd:import namespace="http://www.aes.org/tcf" schemaLocation="tcf.xsd" />

  <xsd:element name="adl" type="adlType" />

  <xsd:complexType name="adlType">
  <xsd:sequence>
    <xsd:element name="version" type="versionType" minOccurs="1" maxOccurs="1" />
    <xsd:element name="project" type="projectType" minOccurs="1" maxOccurs="1" />
    <xsd:element name="system" type="systemType" minOccurs="0" maxOccurs="1" />
    <xsd:element name="sequence" type="sequenceType" minOccurs="1" maxOccurs="1" />
    <xsd:element name="trackList" type="trackListType" minOccurs="0" maxOccurs="1" />
    <xsd:element name="sourceIndex" type="sourceIndexType" minOccurs="1" maxOccurs="1" />
    <xsd:element name="eventList" type="eventListType" minOccurs="1" maxOccurs="1" />
    <xsd:element name="faderList" type="faderListType" minOccurs="0" maxOccurs="1" />
    <xsd:element name="panList" type="panListType" minOccurs="0" maxOccurs="1" />
    <xsd:element name="muteList" type="muteListType" minOccurs="0" maxOccurs="1" />
    <xsd:element name="markerList" type="markerListType" minOccurs="0" maxOccurs="1" />
    <xsd:element name="refList" type="refListType" minOccurs="0" maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:attribute name="id" type="xsd:ID" use="required" />
</xsd:complexType>

  <xsd:complexType name="versionType">
  <xsd:sequence>
    <xsd:element name="adlId" type="xsd:string"
      default="urn:smp:ul:060e2b34.0401010c.0d0a0101.00000000" maxOccurs="1" />
    <xsd:element name="adlUid" type="uuidType" minOccurs="1" maxOccurs="1" />
    <xsd:element name="verAdlVersion" type="edmlVersionType" minOccurs="1" maxOccurs="1" />
    <xsd:element name="verCreator" type="xsd:string" minOccurs="1" maxOccurs="1" />
    <xsd:element name="verCrtr" type="edmlVersionType" minOccurs="1" maxOccurs="1" />
  </xsd:sequence>
  <xsd:attribute name="id" type="xsd:ID" use="required" />
</xsd:complexType>

  <xsd:complexType name="projectType">
  <xsd:sequence>
    <xsd:element name="projTitle" type="xsd:string" minOccurs="1" maxOccurs="1" />
    <xsd:element name="projOriginator" type="xsd:string" minOccurs="1" maxOccurs="1" />
    <xsd:element name="projCreateDate" type="xsd:dateTime" minOccurs="1" maxOccurs="1" />
    <xsd:element name="projNotes" type="xsd:string" minOccurs="1" maxOccurs="1" />
    <xsd:element name="projClientData" type="xsd:string" minOccurs="1" maxOccurs="1" />
  </xsd:sequence>
  <xsd:attribute name="id" type="xsd:ID" use="required" />
</xsd:complexType>

  <xsd:complexType name="systemType">
  <xsd:sequence>
    <xsd:element name="sysSrcOffset" type="xsd:string" minOccurs="0" maxOccurs="1" />
    <xsd:element name="sysBitDepth" type="xsd:integer" default="16" maxOccurs="1" />
  </xsd:sequence>
  </xsd:complexType>

```

```

<xsd:element name="sysAudCodec" type="xsd:string" default="BWF" maxOccurs="1" />
<xsd:element name="sysXfadeLen" type="xsd:integer" minOccurs="0" maxOccurs="1" />
<xsd:element name="sysGain" type="xsd:decimal" minOccurs="0" maxOccurs="1" />
</xsd:sequence>
<xsd:attribute name="id" type="xsd:ID" use="required" />
</xsd:complexType>

<xsd:complexType name="sequenceType">
<xsd:sequence>
<xsd:element name="seqTitle" type="xsd:string" minOccurs="0" maxOccurs="1" />
<xsd:element name="seqDescript" type="xsd:string" minOccurs="0" maxOccurs="1" />
<xsd:element name="seqSampleRate" type="xsd:string" minOccurs="1" maxOccurs="1" />
<xsd:element name="seqSampleRateFactor" type="seqSampleRateFactorType" minOccurs="0"
maxOccurs="1" />
<xsd:element name="seqFrameRate" type="tcf.frameCountType" minOccurs="1" maxOccurs="1" />
<xsd:element name="seqAdlLevel" type="xsd:integer" default="1" maxOccurs="1" />
<xsd:element name="seqClean" type="xsd:boolean" default="false" maxOccurs="1" />
<xsd:element name="seqSort" type="xsd:integer" default="0" maxOccurs="1" />
<xsd:element name="seqMultichan" type="xsd:boolean" default="false" maxOccurs="1" />
<xsd:element name="seqDestStart" type="xsd:string" minOccurs="1" maxOccurs="1" />
</xsd:sequence>
<xsd:attribute name="id" type="xsd:ID" use="required" />
</xsd:complexType>

<xsd:complexType name="trackListType">
<xsd:sequence>
<xsd:element name="track" type="trackType" minOccurs="1" maxOccurs="unbounded" />
</xsd:sequence>
<xsd:attribute name="id" type="xsd:ID" use="required" />
</xsd:complexType>

<xsd:complexType name="sourceIndexType">
<xsd:sequence>
<xsd:element name="index" type="indexType" minOccurs="1" maxOccurs="unbounded" />
</xsd:sequence>
<xsd:attribute name="id" type="xsd:ID" use="required" />
</xsd:complexType>

<xsd:complexType name="eventListType">
<xsd:sequence>
<xsd:element name="eventEntry" type="eventEntryType" minOccurs="1" maxOccurs="unbounded" />
</xsd:sequence>
<xsd:attribute name="id" type="xsd:ID" use="required" />
</xsd:complexType>

<xsd:complexType name="faderListType">
<xsd:sequence>
<xsd:element name="faderPointEntry" type="faderPointEntryType" minOccurs="1" maxOccurs="unbounded"
/>
</xsd:sequence>
<xsd:attribute name="id" type="xsd:ID" use="required" />
</xsd:complexType>

<xsd:complexType name="panListType">
<xsd:sequence>
<xsd:element name="panPointEntry" type="panPointEntryType" minOccurs="1" maxOccurs="unbounded" />
</xsd:sequence>
<xsd:attribute name="id" type="xsd:ID" use="required" />
</xsd:complexType>

```

```

<xsd:complexType name="muteListType">
  <xsd:sequence>
    <xsd:element name="mutePointEntry" type="mutePointEntryType" minOccurs="1" maxOccurs="unbounded"
  />
  </xsd:sequence>
  <xsd:attribute name="id" type="xsd:ID" use="required" />
</xsd:complexType>

<xsd:complexType name="markerListType">
  <xsd:sequence>
    <xsd:element name="markPointEntry" type="markPointEntryType" minOccurs="1" maxOccurs="unbounded"
  />
  </xsd:sequence>
  <xsd:attribute name="id" type="xsd:ID" use="required" />
</xsd:complexType>

<xsd:complexType name="refListType">
  <xsd:sequence>
    <xsd:element name="ref" type="refType" minOccurs="1" maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:attribute name="id" type="xsd:ID" use="required" />
</xsd:complexType>

<xsd:simpleType name='uuidType'>
  <xsd:restriction base='xsd:string'>
    <xsd:pattern value="[0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{12}"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:complexType name="edmlVersionType">
  <xsd:attribute name="majorVersionNumber" type="xsd:nonNegativeInteger" use="required" />
  <xsd:attribute name="minorReleaseNumber" type="xsd:nonNegativeInteger" use="required" />
  <xsd:attribute name="bugRevisionNumber" type="xsd:nonNegativeInteger" use="optional" />
  <xsd:attribute name="stageNumber" type="xsd:nonNegativeInteger" use="optional" />
  <xsd:attribute name="revisionNumber" type="xsd:nonNegativeInteger" use="optional" />
</xsd:complexType>

<xsd:simpleType name="seqSampleRateFactorType">
  <xsd:restriction base="xsd:integer">
    <xsd:enumeration value="1"/>
    <xsd:enumeration value="2"/>
    <xsd:enumeration value="4"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:complexType name="trackType">
  <xsd:attribute name="trackNumber" type="xsd:positiveInteger" use="required" />
  <xsd:attribute name="trackName" type="xsd:string" use="required" />
  <xsd:attribute name="id" type="xsd:ID" use="required" />
</xsd:complexType>

<xsd:complexType name="indexType">
  <xsd:sequence>
    <xsd:element name="filePathSource" type="filePathSourceType" minOccurs="0" maxOccurs="1" />
    <xsd:element name="uriSource" type="uriSourceType" minOccurs="0" maxOccurs="1" />
    <xsd:element name="tapeSource" type="tapeSourceType" minOccurs="0" maxOccurs="1" />
  </xsd:sequence>
  <xsd:attribute name="indexNumber" type="xsd:nonNegativeInteger" use="required" />
  <xsd:attribute name="id" type="xsd:ID" use="required" />
</xsd:complexType>

```

```

<xsd:complexType name="filePathSourceType">
<xsd:sequence>
  <xsd:element name="filePath" type="xsd:string" minOccurs="1" maxOccurs="1" />
  <xsd:element name="uid" type="xsd:string" minOccurs="1" maxOccurs="1" />
  <xsd:element name="fileIn" type="xsd:string" minOccurs="0" maxOccurs="1" />
  <xsd:element name="fileLen" type="xsd:string" minOccurs="0" maxOccurs="1" />
  <xsd:element name="descr" type="xsd:string" minOccurs="0" maxOccurs="1" />
  <xsd:element name="code" type="usageCodeType" minOccurs="0" maxOccurs="1" />
</xsd:sequence>
<xsd:attribute name="keyLetter" type="xsd:string" use="required" fixed="F" />
<xsd:attribute name="id" type="xsd:ID" use="required" />
</xsd:complexType>

<xsd:simpleType name="usageCodeType">
<xsd:restriction base="xsd:string">
  <xsd:enumeration value="N"/>
  <xsd:enumeration value="X"/>
  <xsd:enumeration value="A"/>
</xsd:restriction>
</xsd:simpleType>

<xsd:complexType name="uriSourceType">
<xsd:sequence>
  <xsd:element name="fileLocator" type="xsd:anyURI" minOccurs="1" maxOccurs="1" />
  <xsd:element name="uid" type="xsd:string" minOccurs="1" maxOccurs="1" />
  <xsd:element name="fileIn" type="xsd:string" minOccurs="0" maxOccurs="1" />
  <xsd:element name="fileLen" type="xsd:string" minOccurs="0" maxOccurs="1" />
  <xsd:element name="descr" type="xsd:string" minOccurs="0" maxOccurs="1" />
  <xsd:element name="code" type="usageCodeType" minOccurs="0" maxOccurs="1" />
</xsd:sequence>
<xsd:attribute name="keyLetter" type="xsd:string" use="required" fixed="U" />
<xsd:attribute name="id" type="xsd:ID" use="required" />
</xsd:complexType>

<xsd:complexType name="tapeSourceType">
<xsd:sequence>
  <xsd:element name="tapeName" type="xsd:string" minOccurs="1" maxOccurs="1" />
  <xsd:element name="tapeOrig" type="xsd:string" minOccurs="1" maxOccurs="1" />
  <xsd:element name="fileOrig" type="xsd:string" minOccurs="1" maxOccurs="1" />
  <xsd:element name="tChan" type="integerRangeType" minOccurs="1" maxOccurs="1" />
  <xsd:element name="fChan" type="integerRangeType" minOccurs="1" maxOccurs="1" />
</xsd:sequence>
<xsd:attribute name="keyLetter" type="xsd:string" use="required" fixed="T" />
<xsd:attribute name="id" type="xsd:ID" use="required" />
</xsd:complexType>

<xsd:complexType name="integerRangeType">
<xsd:sequence />
<xsd:attribute name="start" type="xsd:integer" use="required" />
<xsd:attribute name="end" type="xsd:integer" use="optional" />
</xsd:complexType>

<xsd:complexType name="eventEntryType">
<xsd:sequence>
<xsd:choice>
  <xsd:element name="cut" type="cutType" minOccurs="1" maxOccurs="1" />
  <xsd:element name="silence" type="silenceType" minOccurs="1" maxOccurs="1" />
  <xsd:element name="aux" type="auxType" minOccurs="1" maxOccurs="1" />
  <xsd:element name="vid" type="vidType" minOccurs="1" maxOccurs="1" />

```

```

</xsd:choice>
<xsd:element name="altSource" type="altSourceType" minOccurs="0" maxOccurs="unbounded" />
<xsd:element name="fadeIn" type="fadeInType" minOccurs="0" maxOccurs="1" />
<xsd:element name="fadeOut" type="fadeOutType" minOccurs="0" maxOccurs="1" />
<xsd:element name="crossfade" type="crossfadeType" minOccurs="0" maxOccurs="1" />
<xsd:element name="gain" type="gainType" minOccurs="0" maxOccurs="unbounded" />
<xsd:element name="remark" type="remarkType" minOccurs="0" maxOccurs="unbounded" />
</xsd:sequence>
<xsd:attribute name="entryNumber" type="xsd:positiveInteger" use="required" />
<xsd:attribute name="id" type="xsd:ID" use="required" />
</xsd:complexType>

<xsd:complexType name="cutType">
<xsd:sequence>
<xsd:element name="srcType" type="xsd:string" minOccurs="1" maxOccurs="1" default="I" />
<xsd:element name="srcIndex" type="xsd:integer" minOccurs="1" maxOccurs="1" />
<xsd:element name="srcChannel" type="integerRangeType" minOccurs="1" maxOccurs="1" />
<xsd:element name="destChannel" type="integerRangeType" minOccurs="1" maxOccurs="1" />
<xsd:element name="srcIn" type="xsd:string" minOccurs="1" maxOccurs="1" />
<xsd:element name="destIn" type="xsd:string" minOccurs="1" maxOccurs="1" />
<xsd:element name="destOut" type="xsd:string" minOccurs="1" maxOccurs="1" />
<xsd:element name="statusCode" type="statusCodeType" minOccurs="0" maxOccurs="1" />
</xsd:sequence>
<xsd:attribute name="id" type="xsd:ID" use="required" />
</xsd:complexType>

<xsd:complexType name="silenceType">
<xsd:sequence>
<xsd:element name="destChannel" type="integerRangeType" minOccurs="1" maxOccurs="1" />
<xsd:element name="destIn" type="xsd:string" minOccurs="1" maxOccurs="1" />
<xsd:element name="destOut" type="xsd:string" minOccurs="1" maxOccurs="1" />
<xsd:element name="statusCode" type="statusCodeType" minOccurs="0" maxOccurs="1" />
</xsd:sequence>
<xsd:attribute name="id" type="xsd:ID" use="required" />
</xsd:complexType>

<xsd:complexType name="auxType">
<xsd:sequence>
<xsd:element name="destChannel" type="integerRangeType" minOccurs="1" maxOccurs="1" />
<xsd:element name="destIn" type="xsd:string" minOccurs="1" maxOccurs="1" />
<xsd:element name="destOut" type="xsd:string" minOccurs="1" maxOccurs="1" />
<xsd:element name="statusCode" type="statusCodeType" minOccurs="0" maxOccurs="1" />
</xsd:sequence>
<xsd:attribute name="id" type="xsd:ID" use="required" />
</xsd:complexType>

<xsd:complexType name="vidType">
<xsd:sequence>
<xsd:element name="srcType" type="xsd:string" minOccurs="1" maxOccurs="1" default="I" />
<xsd:element name="srcIndex" type="xsd:integer" minOccurs="1" maxOccurs="1" />
<xsd:element name="srcIn" type="xsd:string" minOccurs="1" maxOccurs="1" />
<xsd:element name="destIn" type="xsd:string" minOccurs="1" maxOccurs="1" />
<xsd:element name="destOut" type="xsd:string" minOccurs="1" maxOccurs="1" />
<xsd:element name="statusCode" type="statusCodeType" minOccurs="0" maxOccurs="1" />
</xsd:sequence>
<xsd:attribute name="id" type="xsd:ID" use="required" />
</xsd:complexType>

<xsd:simpleType name="statusCodeType">
<xsd:restriction base="xsd:string">

```

```

<xsd:enumeration value="R"/>
<xsd:enumeration value="E"/>
<xsd:enumeration value="D"/>
<xsd:enumeration value="X"/>
<xsd:enumeration value="M"/>
</xsd:restriction>
</xsd:simpleType>

<xsd:complexType name="altSourceType">
<xsd:sequence>
  <xsd:element name="srcType" type="xsd:string" minOccurs="1" maxOccurs="1" default="I" />
  <xsd:element name="srcIndex" type="xsd:integer" minOccurs="1" maxOccurs="1" />
  <xsd:element name="srcChannel" type="integerRangeType" minOccurs="1" maxOccurs="1" />
  <xsd:element name="srcIn" type="xsd:string" minOccurs="1" maxOccurs="1" />
</xsd:sequence>
<xsd:attribute name="id" type="xsd:ID" use="required" />
</xsd:complexType>

<xsd:complexType name="fadeInType">
<xsd:sequence>
  <xsd:element name="duration" type="xsd:string" minOccurs="1" maxOccurs="1" />
  <xsd:element name="shape" type="shapeType" minOccurs="1" maxOccurs="1" default="LIN" />
  <xsd:element name="curveA" type="xsd:decimal" minOccurs="0" maxOccurs="1" />
  <xsd:element name="curveB" type="xsd:decimal" minOccurs="0" maxOccurs="1" />
  <xsd:element name="curveC" type="xsd:decimal" minOccurs="0" maxOccurs="1" />
  <xsd:element name="srcType" type="xsd:string" minOccurs="0" maxOccurs="1" default="I" />
  <xsd:element name="srcIndex" type="xsd:integer" minOccurs="0" maxOccurs="1" />
  <xsd:element name="srcIn" type="xsd:string" minOccurs="0" maxOccurs="1" />
</xsd:sequence>
<xsd:attribute name="id" type="xsd:ID" use="required" />
</xsd:complexType>

<xsd:complexType name="fadeOutType">
<xsd:sequence>
  <xsd:element name="duration" type="xsd:string" minOccurs="1" maxOccurs="1" />
  <xsd:element name="shape" type="shapeType" minOccurs="1" maxOccurs="1" default="LIN" />
  <xsd:element name="curveA" type="xsd:decimal" minOccurs="0" maxOccurs="1" />
  <xsd:element name="curveB" type="xsd:decimal" minOccurs="0" maxOccurs="1" />
  <xsd:element name="curveC" type="xsd:decimal" minOccurs="0" maxOccurs="1" />
  <xsd:element name="srcType" type="xsd:string" minOccurs="0" maxOccurs="1" default="I" />
  <xsd:element name="srcIndex" type="xsd:integer" minOccurs="0" maxOccurs="1" />
  <xsd:element name="srcIn" type="xsd:string" minOccurs="0" maxOccurs="1" />
</xsd:sequence>
<xsd:attribute name="id" type="xsd:ID" use="required" />
</xsd:complexType>

<xsd:complexType name="crossfadeType">
<xsd:sequence>
  <xsd:element name="previousClip" type="xsd:integer" minOccurs="1" maxOccurs="1" />
  <xsd:element name="destIn" type="xsd:string" minOccurs="1" maxOccurs="1" />
  <xsd:element name="destOut" type="xsd:string" minOccurs="1" maxOccurs="1" />
  <xsd:element name="srcType" type="xsd:string" minOccurs="1" maxOccurs="1" default="I" />
  <xsd:element name="srcIndex" type="xsd:integer" minOccurs="1" maxOccurs="1" />
  <xsd:element name="srcIn" type="xsd:string" minOccurs="1" maxOccurs="1" />
</xsd:sequence>
<xsd:attribute name="id" type="xsd:ID" use="required" />
</xsd:complexType>

<xsd:complexType name="gainType">
<xsd:sequence>

```



```

<xsd:element name="channel" type="integerRangeType" minOccurs="0" maxOccurs="1" />
<xsd:element name="gainValue" type="xsd:decimal" minOccurs="1" maxOccurs="1" default="0.0" />
</xsd:sequence>
<xsd:attribute name="id" type="xsd:ID" use="required" />
</xsd:complexType>

<xsd:complexType name="remarkType">
<xsd:sequence>
<xsd:element name="remType" type="remTypeType" minOccurs="1" maxOccurs="1" />
<xsd:element name="remValue" type="xsd:string" minOccurs="1" maxOccurs="1" />
</xsd:sequence>
<xsd:attribute name="id" type="xsd:ID" use="required" />
</xsd:complexType>

<xsd:simpleType name="shapeType">
<xsd:restriction base="xsd:string">
<xsd:enumeration value="LIN"/>
<xsd:enumeration value="CURVE"/>
</xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="remTypeType">
<xsd:restriction base="xsd:string">
<xsd:enumeration value="NAME"/>
<xsd:enumeration value="SOURCE"/>
<xsd:enumeration value="DESC"/>
<xsd:enumeration value="USER"/>
</xsd:restriction>
</xsd:simpleType>

<xsd:complexType name="faderPointEntryType">
<xsd:sequence>
<xsd:element name="destinationTrack" type="xsd:positiveInteger" minOccurs="1" maxOccurs="1" />
<xsd:element name="destIn" type="xsd:string" minOccurs="1" maxOccurs="1" />
<xsd:element name="faderValue" type="xsd:decimal" minOccurs="1" maxOccurs="1" />
</xsd:sequence>
<xsd:attribute name="id" type="xsd:ID" use="required" />
</xsd:complexType>

<xsd:complexType name="panPointEntryType">
<xsd:sequence>
<xsd:element name="destinationTrack" type="xsd:positiveInteger" minOccurs="1" maxOccurs="1" />
<xsd:element name="destIn" type="xsd:string" minOccurs="1" maxOccurs="1" />
<xsd:element name="leftRightPosition" type="xsd:decimal" minOccurs="1" maxOccurs="1" />
<xsd:element name="frontRearPosition" type="xsd:decimal" minOccurs="1" maxOccurs="1" />
</xsd:sequence>
<xsd:attribute name="id" type="xsd:ID" use="required" />
</xsd:complexType>

<xsd:complexType name="mutePointEntryType">
<xsd:sequence>
<xsd:element name="destinationTrack" type="xsd:positiveInteger" minOccurs="1" maxOccurs="1" />
<xsd:element name="destIn" type="xsd:string" minOccurs="1" maxOccurs="1" />
<xsd:element name="muteValue" type="muteValueType" minOccurs="1" maxOccurs="1" />
</xsd:sequence>
<xsd:attribute name="id" type="xsd:ID" use="required" />
</xsd:complexType>

<xsd:simpleType name="muteValueType">
<xsd:restriction base="xsd:string">

```

```
<xsd:enumeration value="M"/>
<xsd:enumeration value="U"/>
</xsd:restriction>
</xsd:simpleType>

<xsd:complexType name="markPointEntryType">
<xsd:sequence>
<xsd:element name="markerKeyword" type="markerKeywordType" minOccurs="1" maxOccurs="1" />
<xsd:element name="destinationTrack" type="integerRangeType" minOccurs="1" maxOccurs="1" />
<xsd:element name="destIn" type="xsd:string" minOccurs="1" maxOccurs="1" />
<xsd:element name="destOut" type="xsd:string" minOccurs="0" maxOccurs="1" />
<xsd:element name="markerName" type="xsd:string" minOccurs="0" maxOccurs="1" />
</xsd:sequence>
<xsd:attribute name="id" type="xsd:ID" use="required" />
</xsd:complexType>

<xsd:simpleType name="markerKeywordType">
<xsd:restriction base="xsd:string">
<xsd:enumeration value="MK"/>
<xsd:enumeration value="MK-PQ-START"/>
<xsd:enumeration value="MK-PQ-END"/>
<xsd:enumeration value="MK-PQ-INDEX"/>
</xsd:restriction>
</xsd:simpleType>

<xsd:complexType name="refType">
<xsd:sequence>
<xsd:element name="project" type="projectType" minOccurs="1" maxOccurs="1" />
<xsd:element name="system" type="systemType" minOccurs="0" maxOccurs="1" />
<xsd:element name="sequence" type="sequenceType" minOccurs="0" maxOccurs="1" />
<xsd:element name="eventList" type="eventListType" minOccurs="1" maxOccurs="1" />
</xsd:sequence>
<xsd:attribute name="id" type="xsd:ID" use="required" />
</xsd:complexType>
</xsd:schema>
```


Annex B (normative): XSLT Stylesheet

```

<?xml version="1.0" encoding="UTF-8" ?>
<!--
  aes31-3
  XSLT Style sheet for experimental AES31 XML (SIMPLE) -> Edit Decision Markup Language
-->

<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

  <xsl:output encoding="US-ASCII" omit-xml-declaration="yes" method="text"/>

  <xsl:variable name="margin">35</xsl:variable>

  <xsl:template match="/">
    <pre>
      <xsl:text>&lt;ADL&gt;</xsl:text>
      <xsl:apply-templates select="adl/version" />
      <xsl:apply-templates select="adl/project" />
      <xsl:apply-templates select="adl/system" />
      <xsl:apply-templates select="adl/sequence" />
      <xsl:apply-templates select="adl/trackList" />
      <xsl:apply-templates select="adl/sourceIndex" />
      <xsl:apply-templates select="adl/eventList" />
      <xsl:apply-templates select="adl/faderList" />
      <xsl:apply-templates select="adl/panList" />
      <xsl:apply-templates select="adl/muteList" />
      <xsl:apply-templates select="adl/markList" />
      <xsl:apply-templates select="adl/refList" />
      <xsl:text>&#xA;&lt;/ADL&gt;</xsl:text>
    </pre>
  </xsl:template>

  <xsl:template match="version">
    <xsl:text>&#xA;&#x9;&lt;VERSION&gt;</xsl:text>
    <xsl:text>&#xA;&#x9;&#x9;(ADL_ID)</xsl:text>
    <xsl:call-template name="whitespace">
      <xsl:with-param name="count" select="$margin - 8" />
    </xsl:call-template>
    <xsl:text>&quot;</xsl:text>
    <xsl:value-of select="adlId" />
    <xsl:text>&quot;&#xA;&#x9;&#x9;(ADL_UID)</xsl:text>
    <xsl:call-template name="whitespace">
      <xsl:with-param name="count" select="$margin - 9" />
    </xsl:call-template>
    <xsl:value-of select="adlUid" />
    <xsl:text>&#xA;&#x9;&#x9;(VER_ADL_VERSION)</xsl:text>
    <xsl:call-template name="whitespace">
      <xsl:with-param name="count" select="$margin - 17" />
    </xsl:call-template>
    <xsl:for-each select="verAdlVersion">
      <xsl:call-template name="edmlVersionType" />
    </xsl:for-each>
    <xsl:text>&#xA;&#x9;&#x9;(VER_CREATOR)</xsl:text>
    <xsl:call-template name="whitespace">
      <xsl:with-param name="count" select="$margin - 13" />
    </xsl:call-template>
  </xsl:template>

```

```

<xsl:text>&quot;;</xsl:text>
<xsl:value-of select="verCreator" />
<xsl:text>&quot;;&xA;&x9;&x9;(VER_CRTR)</xsl:text>
<xsl:call-template name="whitespace">
  <xsl:with-param name="count" select="$margin - 10" />
</xsl:call-template>
<xsl:for-each select="verCrtr">
  <xsl:call-template name="edmlVersionType" />
</xsl:for-each>
<xsl:text>&xA;&x9;&lt;/VERSION&gt;</xsl:text>
</xsl:template>

<xsl:template match="project">
  <xsl:text>&xA;&x9;&x9;&lt;PROJECT&gt;</xsl:text>
  <xsl:text>&x9;&x9;&x9;(PROJ_TITLE)</xsl:text>
  <xsl:call-template name="whitespace">
    <xsl:with-param name="count" select="$margin - 12" />
  </xsl:call-template>
  <xsl:text>&quot;;</xsl:text>
  <xsl:value-of select="projTitle" />
  <xsl:text>&quot;;&x9;&x9;&x9;(PROJ_ORIGINATOR)</xsl:text>
  <xsl:call-template name="whitespace">
    <xsl:with-param name="count" select="$margin - 17" />
  </xsl:call-template>
  <xsl:text>&quot;;</xsl:text>
  <xsl:value-of select="projOriginator" />
  <xsl:text>&quot;;&x9;&x9;&x9;(PROJ_CREATE_DATE)</xsl:text>
  <xsl:call-template name="whitespace">
    <xsl:with-param name="count" select="$margin - 18" />
  </xsl:call-template>
  <xsl:value-of select="projCreateDate" />
  <xsl:text>&x9;&x9;&x9;(PROJ_NOTES)</xsl:text>
  <xsl:call-template name="whitespace">
    <xsl:with-param name="count" select="$margin - 12" />
  </xsl:call-template>
  <xsl:text>&quot;;</xsl:text>
  <xsl:value-of select="projNotes" />
  <xsl:text>&quot;;&x9;&x9;&x9;(PROJ_CLIENT_DATA)</xsl:text>
  <xsl:call-template name="whitespace">
    <xsl:with-param name="count" select="$margin - 18" />
  </xsl:call-template>
  <xsl:text>&quot;;</xsl:text>
  <xsl:value-of select="projClientData" />
<xsl:text>&quot;;&x9;&x9;&lt;/PROJECT&gt;</xsl:text>
</xsl:template>

<xsl:template match="system">
<xsl:text>&x9;&x9;&x9;&lt;SYSTEM&gt;</xsl:text>
<xsl:if test="count(sysSrcOffset) &gt; 0">
  <xsl:text>&x9;&x9;&x9;(SYS_SRC_OFFSET)</xsl:text>
  <xsl:call-template name="whitespace">
    <xsl:with-param name="count" select="$margin - 16" />
  </xsl:call-template>
  <xsl:value-of select="sysSrcOffset" />
</xsl:if>
<xsl:text>&x9;&x9;&x9;(SYS_BIT_DEPTH)</xsl:text>
<xsl:call-template name="whitespace">
  <xsl:with-param name="count" select="$margin - 15" />
</xsl:call-template>
<xsl:value-of select="sysBitDepth" />

```

```

<xsl:if test="count(sysAudCodec) > 0">
  <xsl:text>&#xA;&#x9;&#x9;(SYS_AUD_CODEC)</xsl:text>
  <xsl:call-template name="whitespace">
    <xsl:with-param name="count" select="$margin - 15" />
  </xsl:call-template>
  <xsl:text>&quot;</xsl:text>
  <xsl:value-of select="sysAudCodec" />
  <xsl:text>&quot;</xsl:text>
</xsl:if>
<xsl:if test="count(sysXfadeLen) > 0">
  <xsl:text>&#xA;&#x9;&#x9;(SYS_XFADE_LEN)</xsl:text>
  <xsl:call-template name="whitespace">
    <xsl:with-param name="count" select="$margin - 15" />
  </xsl:call-template>
  <xsl:value-of select="sysXfadeLen" />
</xsl:if>
<xsl:if test="count(sysGain) > 0">
  <xsl:text>&#xA;&#x9;&#x9;(SYS_GAIN)</xsl:text>
  <xsl:call-template name="whitespace">
    <xsl:with-param name="count" select="$margin - 10" />
  </xsl:call-template>
  <xsl:value-of select="sysGain" />
</xsl:if>
<xsl:text>&#xA;&#x9;&#x9;&lt;/SYSTEM&gt;</xsl:text>
</xsl:template>

<xsl:template match="sequence">
<xsl:text>&#xA;&#xA;&#x9;&lt;SEQUENCE&gt;</xsl:text>
  <xsl:if test="count(seqTitle) > 0">
    <xsl:text>&#xA;&#x9;&#x9;(SEQ_TITLE)</xsl:text>
    <xsl:call-template name="whitespace">
      <xsl:with-param name="count" select="$margin - 11" />
    </xsl:call-template>
    <xsl:text>&quot;</xsl:text>
    <xsl:value-of select="seqTitle" />
    <xsl:text>&quot;</xsl:text>
  </xsl:if>
  <xsl:if test="count(seqDescript) > 0">
    <xsl:text>&#xA;&#x9;&#x9;(SEQ_DESCRIPT)</xsl:text>
    <xsl:call-template name="whitespace">
      <xsl:with-param name="count" select="$margin - 14" />
    </xsl:call-template>
    <xsl:text>&quot;</xsl:text>
    <xsl:value-of select="seqDescript" />
    <xsl:text>&quot;</xsl:text>
  </xsl:if>
  <xsl:text>&#xA;&#x9;&#x9;(SEQ_SAMPLE_RATE)</xsl:text>
  <xsl:call-template name="whitespace">
    <xsl:with-param name="count" select="$margin - 17" />
  </xsl:call-template>
  <xsl:value-of select="seqSampleRate" />
  <xsl:if test="count(seqSampleRateFactor) > 0">
    <xsl:text>&#xA;&#x9;&#x9;(SEQ_SAMPLE_RATE_FACTOR)</xsl:text>
    <xsl:call-template name="whitespace">
      <xsl:with-param name="count" select="$margin - 26" />
    </xsl:call-template>
    <xsl:value-of select="seqSampleRateFactor" />
  </xsl:if>
  <xsl:text>&#xA;&#x9;&#x9;(SEQ_FRAME_RATE)</xsl:text>
  <xsl:call-template name="whitespace">

```



```

<xsl:text>#x20;</xsl:text>
<xsl:choose>
  <xsl:when test="count(fileIn) > 0">
    <xsl:value-of select="fileIn" />
  </xsl:when>
  <xsl:otherwise>
    <xsl:text>_</xsl:text>
  </xsl:otherwise>
</xsl:choose>
<xsl:text>#x20;</xsl:text>
<xsl:choose>
  <xsl:when test="count(fileLen) > 0">
    <xsl:value-of select="fileLen" />
  </xsl:when>
  <xsl:otherwise>
    <xsl:text>_</xsl:text>
  </xsl:otherwise>
</xsl:choose>
<xsl:text>#x20;</xsl:text>
<xsl:choose>
  <xsl:when test="count(descr) > 0">
    <xsl:text>#x20;</xsl:text>
    <xsl:value-of select="descr" />
    <xsl:text>#x20;</xsl:text>
  </xsl:when>
  <xsl:otherwise>
    <xsl:text>_</xsl:text>
  </xsl:otherwise>
</xsl:choose>
<xsl:text>#x20;</xsl:text>
<xsl:choose>
  <xsl:when test="count(code) > 0">
    <xsl:value-of select="code" />
  </xsl:when>
  <xsl:otherwise>
    <xsl:text>_</xsl:text>
  </xsl:otherwise>
</xsl:choose>
<xsl:text>#x20;</xsl:text>
</xsl:for-each>
<xsl:for-each select="uriSource">
<xsl:text>#xA;#x9;#x9;</xsl:text>
  <xsl:call-template name="whitespace">
    <xsl:with-param name="count" select="string-length($indexLength) + 9" />
  </xsl:call-template>
  <xsl:text>(</xsl:text>
  <xsl:value-of select="@keyLetter" />
  <xsl:text>)<x20;&quot;</xsl:text>
  <xsl:value-of select="fileLocator" />
  <xsl:text>#x20;</xsl:text>
  <xsl:value-of select="uid" />
  <xsl:text>#x20;</xsl:text>
  <xsl:choose>
    <xsl:when test="count(fileIn) > 0">
      <xsl:value-of select="fileIn" />
    </xsl:when>
    <xsl:otherwise>
      <xsl:text>_</xsl:text>
    </xsl:otherwise>
  </xsl:choose>
</xsl:for-each>

```



```

<xsl:text>#xA;#x9;#x9;(Entry)#x9;</xsl:text>
<xsl:value-of select="@entryNumber" />
<xsl:text>#x20;</xsl:text>
<xsl:choose>
  <xsl:when test="cut">
    <xsl:text>#x20;(CUT)#x20;</xsl:text>
    <xsl:value-of select="cut/srcType" />
    <xsl:text>#x20;</xsl:text>
    <xsl:value-of select="cut/srcIndex" />
    <xsl:text>#x20;</xsl:text>
    <xsl:for-each select="cut/srcChannel">
      <xsl:call-template name="rangeType" />
    </xsl:for-each>
    <xsl:text>#x20;</xsl:text>

    <xsl:for-each select="cut/destChannel">
      <xsl:call-template name="rangeType" />
    </xsl:for-each>
    <xsl:text>#x20;</xsl:text>
    <xsl:value-of select="cut/srcIn" />
    <xsl:text>#x20;</xsl:text>
    <xsl:value-of select="cut/destIn" />
    <xsl:text>#x20;</xsl:text>
    <xsl:value-of select="cut/destOut" />
    <xsl:text>#x20;</xsl:text>
    <xsl:choose>
      <xsl:when test="count(cut/statusCode) > 0">
        <xsl:value-of select="cut/statusCode" />
      </xsl:when>
      <xsl:otherwise>
        <xsl:text>_</xsl:text>
      </xsl:otherwise>
    </xsl:choose>
  </xsl:when>
  <xsl:when test="silence">
    <xsl:text>#x20;(Silence)#x20;</xsl:text>
    <xsl:for-each select="silence/destChannel">
      <xsl:call-template name="rangeType" />
    </xsl:for-each>
    <xsl:text>#x20;</xsl:text>
    <xsl:value-of select="silence/destIn" />
    <xsl:text>#x20;</xsl:text>
    <xsl:value-of select="silence/destOut" />
    <xsl:text>#x20;</xsl:text>
    <xsl:choose>
      <xsl:when test="count(silence/statusCode) > 0">
        <xsl:value-of select="silence/statusCode" />
      </xsl:when>
      <xsl:otherwise>
        <xsl:text>_</xsl:text>
      </xsl:otherwise>
    </xsl:choose>
  </xsl:when>
  <xsl:when test="aux">
    <xsl:text>#x20;(Aux)#x20;</xsl:text>
    <xsl:for-each select="aux/destChannel">
      <xsl:call-template name="rangeType" />
    </xsl:for-each>
    <xsl:text>#x20;</xsl:text>
    <xsl:value-of select="aux/destIn" />

```



```

<xsl:text>#x20;</xsl:text>
<xsl:value-of select="aux/destOut" />
<xsl:text>#x20;</xsl:text>
<xsl:choose>
  <xsl:when test="count(aux/statusCode) > 0">
    <xsl:value-of select="aux/statusCode" />
  </xsl:when>
  <xsl:otherwise>
    <xsl:text>_</xsl:text>
  </xsl:otherwise>
</xsl:choose>
</xsl:when>
<xsl:when test="vid">
  <xsl:text>#x20;(Vid)#x20;</xsl:text>
  <xsl:value-of select="vid/srcType" />
  <xsl:text>#x20;</xsl:text>
  <xsl:value-of select="vid/srcIndex" />
  <xsl:text>#x20;</xsl:text>
  <xsl:value-of select="vid/srcIn" />
  <xsl:text>#x20;</xsl:text>
  <xsl:value-of select="vid/destIn" />
  <xsl:text>#x20;</xsl:text>
  <xsl:value-of select="vid/destOut" />
  <xsl:text>#x20;</xsl:text>
  <xsl:choose>
    <xsl:when test="count(vid/statusCode) > 0">
      <xsl:value-of select="vid/statusCode" />
    </xsl:when>
    <xsl:otherwise>
      <xsl:text>_</xsl:text>
    </xsl:otherwise>
  </xsl:choose>
</xsl:when>
</xsl:choose>

<xsl:for-each select="altSource">
  <xsl:call-template name="whitespace">
    <xsl:with-param name="count" select="8" />
  </xsl:call-template>
  <xsl:text>#xA;#x9;#x9;#x9;(Alt)#x20;</xsl:text>
  <xsl:value-of select="altSource/srcType" />
  <xsl:text>#x20;</xsl:text>
  <xsl:value-of select="altSource/srcIndex" />
  <xsl:text>#x20;</xsl:text>
  <xsl:for-each select="altSource/srcChannel">
    <xsl:call-template name="rangeType" />
  </xsl:for-each>
  <xsl:text>#x20;</xsl:text>
  <xsl:value-of select="altSource/srcIn" />
  <xsl:text>#x20;</xsl:text>
</xsl:for-each>
<xsl:for-each select="fadeIn">
  <xsl:call-template name="whitespace">
    <xsl:with-param name="count" select="8" />
  </xsl:call-template>
  <xsl:text>#xA;#x9;#x9;#x9;(Infade)#x20;</xsl:text>
  <xsl:value-of select="duration" />
  <xsl:text>#x20;</xsl:text>
  <xsl:value-of select="shape" />
  <xsl:text>#x20;</xsl:text>

```



```

<xsl:choose>
  <xsl:when test="count(curveA) &gt; 0">
    <xsl:value-of select="curveA" />
  </xsl:when>
  <xsl:otherwise>
    <xsl:text>_</xsl:text>
  </xsl:otherwise>
</xsl:choose>
<xsl:text>#x20;</xsl:text>
<xsl:choose>
  <xsl:when test="count(curveB) &gt; 0">
    <xsl:value-of select="curveB" />
  </xsl:when>
  <xsl:otherwise>
    <xsl:text>_</xsl:text>
  </xsl:otherwise>
</xsl:choose>
<xsl:text>#x20;</xsl:text>
<xsl:choose>
  <xsl:when test="count(curveC) &gt; 0">
    <xsl:value-of select="curveC" />
  </xsl:when>
  <xsl:otherwise>
    <xsl:text>_</xsl:text>
  </xsl:otherwise>
</xsl:choose>
<xsl:text>#x20;</xsl:text>
<xsl:value-of select="srcType" />
<xsl:text>#x20;</xsl:text>
<xsl:value-of select="srcIndex" />
<xsl:text>#x20;</xsl:text>
<xsl:value-of select="srcIn" />
<xsl:text>#x20;</xsl:text>
</xsl:for-each>
<xsl:for-each select="fadeOut">
  <xsl:call-template name="whitespace">
    <xsl:with-param name="count" select="8" />
  </xsl:call-template>
  <xsl:text>#xA;#x9;#x9;(Outfade)#x20;</xsl:text>
  <xsl:value-of select="duration" />
  <xsl:text>#x20;</xsl:text>
  <xsl:value-of select="shape" />
  <xsl:text>#x20;</xsl:text>
  <xsl:choose>
    <xsl:when test="count(curveA) &gt; 0">
      <xsl:value-of select="curveA" />
    </xsl:when>
    <xsl:otherwise>
      <xsl:text>_</xsl:text>
    </xsl:otherwise>
  </xsl:choose>
  <xsl:text>#x20;</xsl:text>
  <xsl:choose>
    <xsl:when test="count(curveB) &gt; 0">
      <xsl:value-of select="curveB" />
    </xsl:when>
    <xsl:otherwise>
      <xsl:text>_</xsl:text>
    </xsl:otherwise>
  </xsl:choose>

```

```

<xsl:text>#x20;</xsl:text>
<xsl:choose>
  <xsl:when test="count(curveC) > 0">
    <xsl:value-of select="curveC" />
  </xsl:when>
  <xsl:otherwise>
    <xsl:text>_</xsl:text>
  </xsl:otherwise>
</xsl:choose>
<xsl:text>#x20;</xsl:text>
<xsl:value-of select="srcType" />
<xsl:text>#x20;</xsl:text>
<xsl:value-of select="srcIndex" />
<xsl:text>#x20;</xsl:text>
<xsl:value-of select="srcln" />
<xsl:text>#x20;</xsl:text>
</xsl:for-each>
<xsl:for-each select="crossfade">
  <xsl:call-template name="whitespace">
    <xsl:with-param name="count" select="8" />
  </xsl:call-template>
  <xsl:text>#xA;#x9;#x9;(Xfade)#x20;</xsl:text>
  <xsl:value-of select="previousClip" />
  <xsl:text>#x20;</xsl:text>
  <xsl:value-of select="destIn" />
  <xsl:text>#x20;</xsl:text>
  <xsl:value-of select="destOut" />
  <xsl:text>#x20;</xsl:text>
  <xsl:value-of select="srcType" />
  <xsl:text>#x20;</xsl:text>
  <xsl:value-of select="srcIndex" />
  <xsl:text>#x20;</xsl:text>
  <xsl:value-of select="srcln" />
  <xsl:text>#x20;</xsl:text>
</xsl:for-each>
<xsl:for-each select="gain">
  <xsl:call-template name="whitespace">
    <xsl:with-param name="count" select="8" />
  </xsl:call-template>
  <xsl:text>#xA;#x9;#x9;(Gain)#x20;</xsl:text>
  <xsl:choose>
    <xsl:when test="count(channel) > 0">
      <xsl:for-each select="channel">
        <xsl:call-template name="rangeType" />
      </xsl:for-each>
    </xsl:when>
    <xsl:otherwise>
      <xsl:text>_</xsl:text>
    </xsl:otherwise>
  </xsl:choose>
  <xsl:text>#x20;</xsl:text>
  <xsl:value-of select="gainValue" />
  <xsl:text>#x20;</xsl:text>
</xsl:for-each>
<xsl:for-each select="remark">
  <xsl:call-template name="whitespace">
    <xsl:with-param name="count" select="8" />
  </xsl:call-template>
  <xsl:text>#xA;#x9;#x9;(Rem)#x20;</xsl:text>
  <xsl:value-of select="remType" />

```

```

<xsl:text>#x20;&quot;</xsl:text>
<xsl:value-of select="remValue" />
<xsl:text>&quot;</xsl:text>
</xsl:for-each>
</xsl:for-each>
<xsl:text>#xA;#x9;&lt;/EVENT_LIST&gt;</xsl:text>
</xsl:template>

<xsl:template match="faderList">
  <xsl:text>#xA;#xA;#x9;&lt;/FADER_LIST&gt;</xsl:text>
  <xsl:for-each select="faderPointEntry">
    <xsl:text>#xA;#x9;#x9;(FP)#x9;</xsl:text>
    <xsl:value-of select="destinationTrack" />
    <xsl:text>#x9;</xsl:text>
    <xsl:value-of select="destIn" />
    <xsl:text>#x9;</xsl:text>
    <xsl:value-of select="faderValue" />
    <xsl:text>#x9;</xsl:text>
  </xsl:for-each>
  <xsl:text>#xA;#x9;&lt;/FADER_LIST&gt;</xsl:text>
</xsl:template>

<xsl:template match="panList">
  <xsl:text>#xA;#xA;#x9;&lt;/PAN_LIST&gt;</xsl:text>
  <xsl:for-each select="panPointEntry">
    <xsl:text>#xA;#x9;#x9;(PP)#x9;</xsl:text>
    <xsl:value-of select="destinationTrack" />
    <xsl:text>#x9;</xsl:text>
    <xsl:value-of select="destIn" />
    <xsl:text>#x9;</xsl:text>
    <xsl:value-of select="leftRightPosition" />
    <xsl:text>#x9;</xsl:text>
    <xsl:value-of select="frontRearPosition" />
  </xsl:for-each>
  <xsl:text>#xA;#x9;&lt;/PAN_LIST&gt;</xsl:text>
</xsl:template>

<xsl:template match="muteList">
  <xsl:text>#xA;#xA;#x9;&lt;/MUTE_LIST&gt;</xsl:text>
  <xsl:for-each select="mutePointEntry">
    <xsl:text>#xA;#x9;#x9;(MP)#x9;</xsl:text>
    <xsl:value-of select="destinationTrack" />
    <xsl:text>#x9;</xsl:text>
    <xsl:value-of select="destIn" />
    <xsl:text>#x9;</xsl:text>
    <xsl:value-of select="muteValue" />
  </xsl:for-each>
  <xsl:text>#xA;#x9;&lt;/MUTE_LIST&gt;</xsl:text>
</xsl:template>

<!--<xsl:template match="markList">
  <xsl:text>#xA;#xA;#x9;&lt;/MARK_LIST&gt;</xsl:text>
  <xsl:for-each select="markPointEntry">
    <xsl:text>#xA;#x9;#x9;(</xsl:text>
    <xsl:value-of select="markerKeyword" />
    <xsl:text>)#x9;</xsl:text>
    <xsl:for-each select="destinationTrack">
      <xsl:call-template name="rangeType" />
    </xsl:for-each>
    <xsl:text>#x9;</xsl:text>
  </xsl:for-each>
  <xsl:text>#xA;#x9;&lt;/MARK_LIST&gt;</xsl:text>
</xsl:template>

```

```

<xsl:value-of select="destIn" />
<xsl:text>#x9;</xsl:text>
<xsl:choose>
  <xsl:when test="count(destOut) > 0">
    <xsl:value-of select="destOut" />
  </xsl:when>
  <xsl:otherwise>
    <xsl:text>_</xsl:text>
    <xsl:call-template name="whitespace">
      <xsl:with-param name="count" select="15" />
    </xsl:call-template>
  </xsl:otherwise>
</xsl:choose>
<xsl:text>#x9;&quot;</xsl:text>
<xsl:value-of select="markerName" />
<xsl:text>&quot;</xsl:text>
</xsl:for-each>
<xsl:text>#xA;#x9;&lt;/MARK_LIST&gt;</xsl:text>
</xsl:template> -->
<xsl:template match="markList">
<xsl:text>#xA;#xA;#x9;&lt;/MARK_LIST&gt;</xsl:text>
  <xsl:for-each select="markPointEntry">
    <xsl:text>#xA;#x9;#x9;</xsl:text>
    <xsl:value-of select="markerKeyword" />
    <xsl:text>#x9;</xsl:text>
    <xsl:for-each select="destinationTrack">
      <xsl:call-template name="rangeType" />
    </xsl:for-each>
    <xsl:text>#x9;</xsl:text>
    <xsl:value-of select="destIn" />
    <xsl:text>#x9;</xsl:text>
    <xsl:choose>
      <xsl:when test="count(destOut) > 0">
        <xsl:value-of select="destOut" />
      </xsl:when>
      <xsl:otherwise>
        <xsl:text>_</xsl:text>
        <xsl:call-template name="whitespace">
          <xsl:with-param name="count" select="15" />
        </xsl:call-template>
      </xsl:otherwise>
    </xsl:choose>
    <xsl:text>#x9;&quot;</xsl:text>
    <xsl:value-of select="markerName" />
    <xsl:text>&quot;</xsl:text>
  </xsl:for-each>
<xsl:text>#xA;#x9;&lt;/MARK_LIST&gt;</xsl:text>
<xsl:text>#xA;#xA;#x9;&lt;/NUENDO_CUELIST&gt;</xsl:text>
  <xsl:for-each select="markPointEntry">
    <xsl:text>#xA;#x9;#x9;</xsl:text>
    <xsl:value-of select="destIn" />
    <xsl:text>#x9;</xsl:text>
    <xsl:text>#x9;&quot;</xsl:text>
    <xsl:value-of select="markerName" />
    <xsl:text>&quot;</xsl:text>
  </xsl:for-each>
<xsl:text>#xA;#x9;&lt;/NUENDO_CUELIST&gt;</xsl:text>
</xsl:template>

<xsl:template match="refList">

```

```
<xsl:text>&#xA;&#xA;&#x9;&lt;REF_LIST&gt;</xsl:text>
<xsl:apply-templates select="adl/project" />
<xsl:apply-templates select="adl/system" />
<xsl:apply-templates select="adl/sequence" />
<xsl:apply-templates select="adl/eventList" />
<xsl:text>&#xA;&#x9;&lt;/REF_LIST&gt;</xsl:text>
</xsl:template>

<xsl:template name="edmlVersionType">
  <xsl:number format="01" value="@majorVersionNumber" />
  <xsl:text>.</xsl:text>
  <xsl:number format="01" value="@minorReleaseNumber" />
  <xsl:text>.</xsl:text>
  <xsl:number format="01" value="@bugRevisionNumber" />
  <xsl:text>.</xsl:text>
  <xsl:number format="01" value="@stageNumber" />
  <xsl:text>.</xsl:text>
  <xsl:number format="01" value="@revisionNumber" />
</xsl:template>

<xsl:template name="rangeType">
  <xsl:value-of select="@start" />
  <xsl:if test="count(@end) &gt; 0">
    <xsl:text>~</xsl:text>
    <xsl:value-of select="@end" />
  </xsl:if>
</xsl:template>

<xsl:template name="whitespace">
  <xsl:param name="count" select="1"/>
  <xsl:if test="$count &gt; 0">
    <xsl:text>&#x20;</xsl:text>
    <xsl:call-template name="whitespace">
      <xsl:with-param name="count" select="$count - 1" />
    </xsl:call-template>
  </xsl:if>
</xsl:template>

</xsl:stylesheet>
```

Annex C (Informative) - Informative references

C.1 ADL version

The ADL version header requires a valid version number for the ADL structure (see 5.3).

Audio decision lists shall indicate their compliance with this 2022 revision of the standard by using the major and minor version numbers specified below:

01.02

C.2 ADL format identifier

The universal ID of the ADL format is registered as a Universal Label in the SMPTE Metadata Dictionary under the AES Class 13 registry. The ADL format specified in this standard is identified by:

urn:smppte:ul:060e2b34.0401010C.0d0a0101.00000000

Bibliography

AES31-1-2001 (r2011): *AES standard for network and file transfer of audio - Audio-file transfer and exchange - Part 1: Disk format.* Audio Engineering Society, NY, NY., US. (Now stabilized)

AES31-2-2012: *Audio-file transfer and exchange - Part 2: File Format for Transferring Digital Audio Data Between Systems of Different Type and Manufacture.* Audio Engineering Society, NY, NY., US.