

INTERNATIONAL STANDARD

NORME INTERNATIONALE

**Common control interface for networked digital audio and video products –
Part 2: Audio**

**Interface de commande commune pour produits audio et vidéo numériques
connectés en réseau –
Partie 2: Audio**

IECNORM.COM : Click to view the full PDF of IEC 62379-2:2008



THIS PUBLICATION IS COPYRIGHT PROTECTED

Copyright © 2008 IEC, Geneva, Switzerland

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from either IEC or IEC's member National Committee in the country of the requester. If you have any questions about IEC copyright or have an enquiry about obtaining additional rights to this publication, please contact the address below or your local IEC member National Committee for further information.

Droits de reproduction réservés. Sauf indication contraire, aucune partie de cette publication ne peut être reproduite ni utilisée sous quelque forme que ce soit et par aucun procédé, électronique ou mécanique, y compris la photocopie et les microfilms, sans l'accord écrit de l'IEC ou du Comité national de l'IEC du pays du demandeur. Si vous avez des questions sur le copyright de l'IEC ou si vous désirez obtenir des droits supplémentaires sur cette publication, utilisez les coordonnées ci-après ou contactez le Comité national de l'IEC de votre pays de résidence.

IEC Central Office
3, rue de Varembe
CH-1211 Geneva 20
Switzerland

Tel.: +41 22 919 02 11
info@iec.ch
www.iec.ch

About the IEC

The International Electrotechnical Commission (IEC) is the leading global organization that prepares and publishes International Standards for all electrical, electronic and related technologies.

About IEC publications

The technical content of IEC publications is kept under constant review by the IEC. Please make sure that you have the latest edition, a corrigendum or an amendment might have been published.

IEC publications search - webstore.iec.ch/advsearchform

The advanced search enables to find IEC publications by a variety of criteria (reference number, text, technical committee, ...). It also gives information on projects, replaced and withdrawn publications.

IEC Just Published - webstore.iec.ch/justpublished

Stay up to date on all new IEC publications. Just Published details all new publications released. Available online and once a month by email.

IEC Customer Service Centre - webstore.iec.ch/csc

If you wish to give us your feedback on this publication or need further assistance, please contact the Customer Service Centre: sales@iec.ch.

IEC online collection - oc.iec.ch

Discover our powerful search engine and read freely all the publications previews. With a subscription you will always have access to up to date content tailored to your needs.

Electropedia - www.electropedia.org

The world's leading online dictionary on electrotechnology, containing more than 22 000 terminological entries in English and French, with equivalent terms in 18 additional languages. Also known as the International Electrotechnical Vocabulary (IEV) online.

A propos de l'IEC

La Commission Electrotechnique Internationale (IEC) est la première organisation mondiale qui élabore et publie des Normes internationales pour tout ce qui a trait à l'électricité, à l'électronique et aux technologies apparentées.

A propos des publications IEC

Le contenu technique des publications IEC est constamment revu. Veuillez vous assurer que vous possédez l'édition la plus récente, un corrigendum ou amendement peut avoir été publié.

Recherche de publications IEC -

webstore.iec.ch/advsearchform

La recherche avancée permet de trouver des publications IEC en utilisant différents critères (numéro de référence, texte, comité d'études, ...). Elle donne aussi des informations sur les projets et les publications remplacées ou retirées.

IEC Just Published - webstore.iec.ch/justpublished

Restez informé sur les nouvelles publications IEC. Just Published détaille les nouvelles publications parues. Disponible en ligne et une fois par mois par email.

Service Clients - webstore.iec.ch/csc

Si vous désirez nous donner des commentaires sur cette publication ou si vous avez des questions contactez-nous: sales@iec.ch.

IEC online collection - oc.iec.ch

Découvrez notre puissant moteur de recherche et consultez gratuitement tous les aperçus des publications. Avec un abonnement, vous aurez toujours accès à un contenu à jour adapté à vos besoins.

Electropedia - www.electropedia.org

Le premier dictionnaire d'électrotechnologie en ligne au monde, avec plus de 22 000 articles terminologiques en anglais et en français, ainsi que les termes équivalents dans 16 langues additionnelles. Egalement appelé Vocabulaire Electrotechnique International (IEV) en ligne.

INTERNATIONAL STANDARD

NORME INTERNATIONALE

**Common control interface for networked digital audio and video products –
Part 2: Audio**

**Interface de commande commune pour produits audio et vidéo numériques
connectés en réseau –
Partie 2: Audio**

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

COMMISSION
ELECTROTECHNIQUE
INTERNATIONALE

ICS 33.160.01

ISBN 978-2-8322-9445-1

**Warning! Make sure that you obtained this publication from an authorized distributor.
Attention! Veuillez vous assurer que vous avez obtenu cette publication via un distributeur agréé.**

CONTENTS

FOREWORD.....	4
INTRODUCTION.....	6
1 Scope.....	7
2 Normative references	7
3 Terms, definitions and abbreviations	7
3.1 Abbreviations	7
4 Audio format definitions.....	8
4.1 Audio signal format definitions.....	8
4.1.1 Audio parameters	8
4.1.2 Audio signal formats	9
4.2 Audio transport format definitions	12
4.3 Audio metadata format definitions	12
5 MIB definitions for audio blocks	13
5.1 General.....	13
5.2 Type definitions.....	13
5.2.1 Textual conventions.....	13
5.2.2 Sequences	15
5.3 Audio port and associated managed object type definitions	16
5.3.1 Generic port functionality.....	16
5.3.2 AES3 ancillary data	18
5.3.3 Phantom power	19
5.3.4 Audio locked to reference.....	19
5.4 Other audio block and associated managed object type definitions.....	20
5.4.1 Audio mixer blocks	20
5.4.2 Audio crosspoint blocks	23
5.4.3 Audio clip player blocks	26
5.4.4 Audio limiter blocks	29
5.4.5 Audio converter blocks	30
5.4.6 Audio level alarm blocks	32
6 Status broadcasts.....	34
6.1 General.....	34
6.2 Type definitions.....	34
6.2.1 Textual conventions.....	34
6.2.2 Sequences	34
6.3 Audio formats mapping.....	34
6.3.1 audioFormatsMapTable	35
6.3.2 audioFormatsMapEntry.....	35
6.3.3 afmNumber.....	35
6.3.4 afmFormat.....	35
6.4 Page formats.....	35
6.4.1 Audio port page	35
6.4.2 AES3 ancillary data page	35
6.4.3 Audio mixer page.....	36
6.4.4 Audio crosspoint page	36
6.4.5 Audio clip player page	37
6.4.6 Audio limiter page.....	37

6.4.7	Audio converter page	38
6.4.8	Audio level alarm page	38
6.5	Page groups	39
6.5.1	audioPorts	39
6.5.2	standardAudioBlocks	39
6.5.3	audioAlarms	40
Annex A (informative)	Machine-readable audio format definitions	41
Annex B (informative)	Machine-readable audio block definitions	56
Annex C (informative)	Machine-readable status page group definitions	74
Annex D (informative)	Machine-readable status page MIB definitions	75
Annex E (informative)	Worked examples	77
Annex F (informative)	Tree of example audio formats	86
Figure 1	– Audio port blocks	17
Figure 2	– Audio mixer block	21
Figure 3	– Audio crosspoint block	23
Figure 4	– Audio clip player block	26
Figure 5	– Audio limiter block	29
Figure 6	– Audio converter block	31
Figure 7	– Audio level alarm block	32
Table 1	– Managed objects for audio ports	17
Table 2	– Managed objects for AES3 ancillary data	18
Table 3	– Managed objects for phantom power	19
Table 4	– Managed objects for audio locked	20
Table 5	– Managed objects for audio mixer blocks	21
Table 6	– Managed objects for audio crosspoint blocks	23
Table 7	– Managed objects for audio clip player blocks	27
Table 8	– Managed objects for audio limiter blocks	30
Table 9	– Managed objects for audio converter blocks	31
Table 10	– Managed objects for audio level alarm blocks	33
Table 11	– Managed objects for audio format mappings	35
Table 12	– Status entries for audio port page	35
Table 13	– Status entries for AES3 ancillary data page	36
Table 14	– Status entries for audio mixer page	36
Table 15	– Status entries for audio crosspoint page	37
Table 16	– Status entries for audio clip player page	37
Table 17	– Status entries for audio limiter page	38
Table 18	– Status entries for audio converter page	38
Table 19	– Status entries for audio level alarm page	39

INTERNATIONAL ELECTROTECHNICAL COMMISSION

COMMON CONTROL INTERFACE FOR NETWORKED DIGITAL AUDIO AND VIDEO PRODUCTS –**Part 2: Audio****FOREWORD**

- 1) The International Electrotechnical Commission (IEC) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, IEC publishes International Standards, Technical Specifications, Technical Reports, Publicly Available Specifications (PAS) and Guides (hereafter referred to as "IEC Publication(s)"). Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation. IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.
- 2) The formal decisions or agreements of IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC National Committees.
- 3) IEC Publications have the form of recommendations for international use and are accepted by IEC National Committees in that sense. While all reasonable efforts are made to ensure that the technical content of IEC Publications is accurate, IEC cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.
- 4) In order to promote international uniformity, IEC National Committees undertake to apply IEC Publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any IEC Publication and the corresponding national or regional publication shall be clearly indicated in the latter.
- 5) IEC provides no marking procedure to indicate its approval and cannot be rendered responsible for any equipment declared to be in conformity with an IEC Publication.
- 6) All users should ensure that they have the latest edition of this publication.
- 7) No liability shall attach to IEC or its directors, employees, servants or agents including individual experts and members of its technical committees and IEC National Committees for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication, use of, or reliance upon, this IEC Publication or any other IEC Publications.
- 8) Attention is drawn to the Normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.
- 9) Attention is drawn to the possibility that some of the elements of this IEC Publication may be the subject of patent rights. IEC shall not be held responsible for identifying any or all such patent rights.

International Standard IEC 62379-2 has been prepared technical area 4: Digital system interfaces and protocols, of IEC technical committee 100: Audio, video and multimedia systems and equipment.

The text of this standard is based on the following documents:

FDIS	Report on voting
100/1405/FDIS	100/1445/RVD

Full information on the voting for the approval of this standard can be found in the report on voting indicated in the above table.

This publication has been drafted in accordance with the ISO/IEC Directives, Part 2.

The committee has decided that the contents of this publication will remain unchanged until the maintenance result date indicated on the IEC web site under "<http://webstore.iec.ch>" in the data related to the specific publication. At this date, the publication will be

- reconfirmed,
- withdrawn,
- replaced by a revised edition, or
- amended.

IECNORM.COM : Click to view the full PDF of IEC 62379-2:2008

INTRODUCTION

IEC 62379 specifies the common control interface, a protocol for managing equipment which conveys audio and/or video across digital networks.

This part of IEC 62379 specifies those aspects that are specific to audio equipment.

An introduction to the common control interface is given in IEC 62739-1.

IECNORM.COM : Click to view the full PDF of IEC 62379-2:2008

COMMON CONTROL INTERFACE FOR NETWORKED DIGITAL AUDIO AND VIDEO PRODUCTS –

Part 2: Audio

1 Scope

This part of IEC 62379 specifies aspects of the common control interface of IEC 62379-1 that are specific to audio.

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.

AES3-2003, *AES standard for digital audio – Digital input-output interfacing – Serial transmission format for two-channel linearly represented digital audio data*

AES10-2003, *AES recommended practice for digital audio engineering – Serial multichannel audio digital interface (MADI)*

AES50-2005, *AES standard for digital audio engineering – High-resolution multi-channel audio interconnection (HRMAI)*

IEC 62379-1:2007, *Common control interface for networked audio and video products – Part 1: General*

ITU-T Recommendation G.711, *Pulse code modulation (PCM) of voice frequencies*

ITU-T Recommendation G.722, *7kHz audio-coding within 64 kbit/s*

ITU-T Recommendation J.41, *Characteristics of equipment for the coding of analogue high quality sound programme signals for transmission on 384 kbit/s channels*

ITU-T Recommendation J.57, *Transmission of digital studio quality sound signals over H1 channels*

3 Terms, definitions and abbreviations

For the purposes of this document, the terms and definitions given in IEC 62379-1 apply along with the following abbreviations.

3.1 Abbreviations

3.1.1

pulse code modulation
PCM

3.1.2

motion pictures expert group
MPEG

3.1.3 advanced audio coding AAC

4 Audio format definitions

4.1 Audio signal format definitions

At any point in the audio signal chain, the audio data will be in a particular format. For management purposes, the format shall be identified by an object identifier, either a “common control interface standard” object identifier defined in this standard or an object identifier defined elsewhere.

NOTE Permitting audio format identifiers to be defined outside this standard allows use of proprietary formats within the standard protocol and also allows industry standard formats to emerge that may eventually be incorporated into future revisions of this standard.

4.1.1 Audio parameters

4.1.1.1 General

The definitions in 4.1.2 make reference to parameters which provide additional information about the format. These parameters shall be mapped to sub-identifier values as specified in the other subclauses of 4.1.1. Any parameter may be unspecified.

The sub-identifier values shall be appended to the object identifiers as additional arcs, in the order in which the parameters are listed in the relevant subclause of 4.1.2; except that if a parameter is unspecified and either it is the last parameter or all subsequent parameters are also unspecified, then it shall be omitted.

NOTE For all parameters, “unspecified” is coded as zero, so this rule ensures that the OID does not end with a zero arc.

EXAMPLE: If the last two parameters are bit depth and sampling frequency, then 16-bit 48kHz would be coded as 16.48000, 16-bit (with sampling frequency unspecified) as .16, and 48kHz (with bit depth unspecified) as .0.48000.

4.1.1.2 Channel arrangement

The sub-identifier for channel arrangement shall be a value of the following type:

```
ChannelArrangement ::= INTEGER {
    unspecified          (0),
    discreteMono         (1),
    stereo               (2),
    jointStereo          (3),
    surround             (4),
    surroundWithDownmix (5)
} (unspecified.. surroundWithDownmix)
```

`discreteMono` shall indicate that each channel (if there is more than one) is a separate audio signal.

`stereo` should only be used with an even number of channels, and shall indicate that each pair of channels (if there is more than one pair) is a separate stereo audio signal, with the first channel of the pair being the left channel.

`jointStereo` should only be used with an even number of channels, and shall indicate that each pair of channels (if there is more than one pair) is a separate stereo audio signal, with the first channel of the pair being the M channel and the second the S channel.

`surround` and `surroundWithDownmix` should only be used with 3 to 8 channels (inclusive). Each shall indicate that the first six channels contain left, right, centre, low frequency effects, left surround, and right surround respectively. `surround` shall indicate that the next two channels contain left and right channels (respectively) of a stereo signal suitable for matrix decoding. `surroundWithDownmix` shall indicate that the next two channels contain left and right channels (respectively) of a stereo downmix.

Where more than one arrangement can be used to describe a format, the smallest applicable value should be used.

EXAMPLE 1 A single stereo pair (2 channels) could be described by values `stereo` (2), `surround` (4), or `surroundWithDownmix` (5). The smallest of these values, i.e. 2, should be used.

EXAMPLE 2 Surround sound with no accompanying stereo signal (6 channels) could be described by values `surround` (4), or `surroundWithDownmix` (5). The smaller of these values, i.e. 4, should be used.

4.1.1.3 Number of channels

The sub-identifier for number of channels shall be a value of the following type:

```
NumberChannels ::= INTEGER
-- An integer representing the number of audio channels
-- A value of zero shall indicate unspecified.
```

4.1.1.4 Bit depth

The sub-identifier for bit depth shall be a value of the following type:

```
BitDepth ::= INTEGER
-- An integer representing the audio bit depth in bits per sample.
-- A value of zero shall indicate unspecified.
```

4.1.1.5 Sampling frequency

The sub-identifier for sampling frequency shall be a value of the following type:

```
SamplingFrequency ::= INTEGER
-- An integer representing the audio sampling frequency in Hz
-- A value of zero shall indicate unspecified.
```

4.1.1.6 Bit rate

The sub-identifier for bit rate shall be a value of the following type:

```
BitRate ::= INTEGER
-- An integer representing the bit rate of the encoded signal in bits per
-- second.
-- A value of zero shall indicate unspecified.
```

4.1.2 Audio signal formats

Audio signal formats shall be rooted at the following location in the MIB tree:

```
iec62379          OBJECT IDENTIFIER ::= { iso(1) standard(0) 62379 }
audioFormat       OBJECT IDENTIFIER ::= { iec62379 audio(2) format(2) }
audioSignalFormat OBJECT IDENTIFIER ::= { audioFormat signal(1) }
```

The following definitions shall be used to identify the specified formats.

NOTE Annex A contains an exemplar set of formats defined by this standard.

4.1.2.1 Unspecified audio

```
unspecifiedAudio      OBJECT IDENTIFIER ::=
                        { audioSignalFormat unspecified(0) }
-- wildcard - any supported format allowed
```

4.1.2.2 No audio

```
noAudio               OBJECT IDENTIFIER ::= { audioSignalFormat none(1) }
-- indicates the output is muted
```

4.1.2.3 Analogue audio

```
analogueAudio        OBJECT IDENTIFIER ::= { audioSignalFormat analogue(2)
}
-- analogue audio
```

The analogue audio identifier shall have two parameters. The first shall be the channel arrangement and the second shall be the number of channels.

4.1.2.4 PCM audio

```
pcmAudio              OBJECT IDENTIFIER ::= { audioSignalFormat pcm(3) }
-- linear PCM audio
```

The linear PCM format identifier shall have four parameters. The first shall be the channel arrangement, the second shall be the number of channels, the third shall be the bit depth and the fourth shall be the sampling frequency.

4.1.2.5 MPEG-1 layer 2 audio

```
mp2Audio              OBJECT IDENTIFIER ::= { audioSignalFormat mp2(4) }
-- MPEG-1 layer 2 audio
```

The MP2 format identifier shall have four parameters. The first shall be the channel arrangement, the second shall be the number of channels, the third shall be the sampling frequency, and the fourth shall be the bit rate.

4.1.2.6 MPEG-1 layer 3 audio

```
mp3Audio              OBJECT IDENTIFIER ::= { audioSignalFormat mp3(5) }
-- MPEG-1 layer 3 audio
```

The MP3 format identifier shall have four parameters. The first shall be the channel arrangement, the second shall be the number of channels, the third shall be the sampling frequency and the fourth shall be the bit rate.

4.1.2.7 AAC audio

```
aacAudio              OBJECT IDENTIFIER ::= { audioSignalFormat aac(6) }
-- AAC audio
```

```
aacLC                 OBJECT IDENTIFIER ::= { aacAudio aacLC(1) }
-- AAC audio with the low complexity profile
```

```
aacMain               OBJECT IDENTIFIER ::= { aacAudio aacMain(2) }
-- AAC audio with the main profile
```

```
aacSRS                OBJECT IDENTIFIER ::= { aacAudio aacSRS(3) }
-- AAC audio with the sample-rate-scalable profile
```

```
aacLTP          OBJECT IDENTIFIER ::= { aacAudio aacLTP(4) }
-- AAC audio with the long term prediction profile
```

```
aacLD           OBJECT IDENTIFIER ::= { aacAudio aacLD(5) }
-- AAC audio with the low delay profile
```

The AAC format identifier shall have four parameters. The first shall be the channel arrangement, the second shall be the number of channels, the third shall be the sampling frequency and the fourth shall be the bit rate.

4.1.2.8 Audio conforming to ITU-T Recommendation G.711

```
g711Audio       OBJECT IDENTIFIER ::= { audioSignalFormat g711(7) }
-- G711 audio
```

```
g711ALaw        OBJECT IDENTIFIER ::= { g711 aLaw(1) }
-- G711 audio, A-Law encoded
```

```
g711MuLaw       OBJECT IDENTIFIER ::= { g711 muLaw(2) }
-- G711 audio, mu-Law encoded
```

4.1.2.9 Audio conforming to ITU-T Recommendation G.722

```
g722Audio       OBJECT IDENTIFIER ::= { audioSignalFormat g722(8) }
-- G722 audio
```

The G722 identifier shall have one parameter. This shall be the bit rate.

4.1.2.10 APT-X audio

```
aptXAudio       OBJECT IDENTIFIER ::= { audioSignalFormat aptX(9) }
-- apt-X audio
```

The APT-X format identifier shall have five parameters. The first shall be the channel arrangement, the second shall be the number of channels, the third shall be the bit depth, the fourth shall be the sampling frequency, and the fifth shall be the bit rate.

4.1.2.11 Enhanced APT-X audio

```
enhancedAptXAudio OBJECT IDENTIFIER ::= { audioSignalFormat enhAptX(10) }
-- enhanced apt-X audio
```

The enhanced APT-X format identifier shall have five parameters. The first shall be the channel arrangement, the second shall be the number of channels, the third shall be the bit depth, the fourth shall be the sampling frequency and the fifth shall be the bit rate.

4.1.2.12 Audio conforming to ITU-T Recommendation J.41

```
j41Audio        OBJECT IDENTIFIER ::= { audioSignalFormat j41(11) }
-- J41 audio
```

```
j41ALawA        OBJECT IDENTIFIER ::= { j41 aLawA(1) }
-- J41 audio using A-law companding, variant A
```

```
j41ALawB        OBJECT IDENTIFIER ::= { j41 aLawB(2) }
-- J41 audio using A-law companding, variant B
```

```
j41Nic          OBJECT IDENTIFIER ::= { j41 nic(3) }
-- J41 audio using near instantaneous companding
```

4.1.2.13 Audio conforming to ITU-T Recommendation J.57

```
j57Audio          OBJECT IDENTIFIER ::= { audioSignalFormat j57(12) }
-- J57 audio

j57H11            OBJECT IDENTIFIER ::= { j57 h11(1) }
-- J57 audio using an H11 channel

j57H12            OBJECT IDENTIFIER ::= { j57 h12(2) }
-- J57 audio using an H12 channel
```

4.1.2.14 Invalid audio

```
invalidAudio      OBJECT IDENTIFIER ::= { audioSignalFormat invalid(13)
}
-- indicates an error, such as inability to decode a signal earlier in
-- the chain
```

4.2 Audio transport format definitions

Audio transport formats shall be rooted at the following location in the MIB tree:

```
audioTransportFormat OBJECT IDENTIFIER ::= { audioFormat transport(2) }
```

The following definitions shall be used to identify the specified transport formats.

```
unspecifiedTransport OBJECT IDENTIFIER ::=
{ audioTransportFormat unspecified(0) }

analogue      OBJECT IDENTIFIER ::= { audioTransportFormat analogue(1) }
AES3          OBJECT IDENTIFIER ::= { audioTransportFormat aes3(2) }
AES10         OBJECT IDENTIFIER ::= { audioTransportFormat aes10(3) }
AES50         OBJECT IDENTIFIER ::= { audioTransportFormat aes50(4) }
```

4.3 Audio metadata format definitions

Audio metadata formats shall be rooted at the following location in the MIB tree:

```
audioMetadataFormat OBJECT IDENTIFIER ::= { audioFormat metadata(3) }
```

The following definitions shall be used to identify the specified metadata formats.

```
unspecifiedMetadata OBJECT IDENTIFIER ::=
{ audioMetadataFormat unspecified(0) }
```

5 MIB definitions for audio blocks

5.1 General

This clause defines a set of managed object types for representing control functions in network controlled audio equipment. The format of the definitions is as specified in IEC 62379-1.

For management purposes, a piece of audio equipment shall be modelled as a number of discrete audio blocks and audio connectors, as specified in IEC 62379-1. Each audio block may have zero or more inputs and zero or more outputs, and each input or output may carry one or more channels. Each audio connector shall connect one audio block output to one audio block input with a one-to-one mapping of channels between the blocks.

NOTE 1 A piece of equipment may be fixed-function, in which case the number of audio blocks present and the connections between them will be immutable, or it may be programmable, in which case the number of audio blocks present and/or the connections between them may be changed by the user.

Each audio block shall be modelled either by one of the standard audio block types defined in this standard or by an audio block type defined elsewhere. Associated with each defined block type shall be a (possibly empty) group of managed object types that represent the control functions for that block. A block type shall be identified by the node in the object identifier tree that is the root node for the group of managed object types associated with that block type.

NOTE 2 Permitting audio block types to be defined outside this standard allows control of proprietary functions using the standard protocol and also allows industry standard block types to emerge that may eventually be incorporated into future revisions of this standard.

NOTE 3 An empty group of managed object types is permitted to allow for blocks that have no associated control functions.

NOTE 4 Annex E contains worked examples of the block structure.

5.2 Type definitions

In addition to the types defined in IEC 62379-1, the following types are used to specify the syntax of the abstract data structures representing managed object values.

5.2.1 Textual conventions

```
AudioTransportType ::= OBJECT IDENTIFIER
-- A reference to the transport used for an audio connection.
-- The value may be defined in 4.2, or in a sub-part of IEC 62379-5, or
-- elsewhere.
```

```
AudioLevel ::= INTEGER {
    mInfinity (-20000),
    fullScale (0),
    pInfinity (20000)
} (mInfinity..pInfinity)
-- An absolute or relative audio level in units of 0.01dB.
```

```
AudioPhase ::= INTEGER (-18000..18000)
-- An absolute or relative phase value in units of 0.01 degree.
```

```
AudioQuality ::= INTEGER {
    worst (1),
    low (32),
    high (96),
    best (127)
} (worst..best)
-- An enumeration identifying an audio signal processing quality level.
```

-- Semantics are equipment specific. Equipment that supports selectable
-- quality levels shall as a minimum support the values low and high.

AudioChannel ::= INTEGER {
 left (1),
 right (2)
} (1..240)
-- An enumeration identifying an audio channel. Stereo equipment shall
-- accept the values left and right. Multi-channel equipment should
-- accept a contiguous range of values starting from 1 and may disregard
-- the association of the values 1 and 2 with left and right.

Aes3ChannelData ::= OCTET STRING (SIZE(24))
-- The channel status data associated with an AES3 channel.

Aes3UserData ::= OCTET STRING (SIZE(24))
-- User data associated with an AES3 channel.

AudioClipSelector ::= INTEGER (1..2147483647)
-- An enumeration identifying an audio clip. Semantics are equipment
-- specific.

AudioClipActivation ::= INTEGER {
 play (1),
 stop (2),
 stopAtEnd (3)
} (play..stopAtEnd)
-- An enumeration for audio clip playback control.

AudioClipTrackControl ::= INTEGER {
 previous (1),
 next (2)
} (previous..next)
-- An enumeration for audio clip track control.

AudioClipRepeat ::= INTEGER {
 none (1),
 track (2),
 all (3)
} (none..all)
-- An enumeration for audio clip repeat control.

AudioRecoveryMode ::= INTEGER {
 auto (1),
 slow (2),
 fast (3)
} (auto..fast)
-- An enumeration identifying an audio limiter recovery mode. Semantics
-- are equipment specific.

AudioAlarmStatus ::= INTEGER {
 ok (1),
 warning (2),
 failure (3)
} (ok..failure)
-- An enumeration representing the status of an audio alarm

AudioLevelAlarmType ::= INTEGER {
 lower (1),
 higher (2)
} (lower..higher)
-- An enumeration representing the trigger condition for an audio level

```
-- alarm. If lower, the alarm is triggered by the audio level being less
-- than the threshold; if higher, the alarm is triggered by the audio
-- level being greater than the threshold.
```

5.2.2 Sequences

```
APortEntry ::= SEQUENCE {
    aPortBlockId      BlockId,
    aPortDirection    PortDirection,
    aPortFormat        MediaFormat,
    aPortTransport     AudioTransportType,
    aPortName          Utf8String
}

Aes3DataEntry ::= SEQUENCE {
    aes3BlockId        BlockId,
    aes3ChannelId       AudioChannel,
    aes3ChannelData     Aes3ChannelData,
    aes3UserData        Aes3UserData,
    aes3ValidityError   TruthValue
}

APhantomEntry ::= SEQUENCE {
    aPhantomBlockId    BlockId,
    aPhantomEnabled     TruthValue,
    aPhantomLevel       CardinalNumber
}

ALockedEntry ::= SEQUENCE {
    aLockedBlockId      BlockId,
    aLockedTime          CardinalNumber,
    aLockedSamplesInserted CardinalNumber,
    aLockedSamplesDropped CardinalNumber
}

AMixerBlockEntry ::= SEQUENCE {
    aMixerBlockId        BlockId,
    aMixerFadeDuration    CardinalNumber,
    aMixerFadeNow         TruthValue
}

AMixerInputEntry ::= SEQUENCE {
    aMixerInputBlockId    BlockId,
    aMixerInputNumber      IndexNumber,
    aMixerInputLevel       AudioLevel,
    aMixerInputFadeToLevel AudioLevel,
    aMixerInputDelay        CardinalNumber
}

ACrosspointBlockEntry ::= SEQUENCE {
    aCrosspointBlockId    BlockId,
    aCrosspointConfigure   TruthValue,
    aCrosspointCopy        BlockId
}

ACrosspointPathEntry ::= SEQUENCE {
    aCrosspointPathBlockId BlockId,
    aCrosspointPathSrc       AudioChannel,
    aCrosspointPathDst       AudioChannel,
    aCrosspointPathGain      AudioLevel,
    aCrosspointPathNewGain   AudioLevel,
    aCrosspointPathPhase     AudioPhase,
    aCrosspointPathNewPhase  AudioPhase
}
```

```

}

AClipPlayerBlockEntry ::= SEQUENCE {
    acpBlockId          BlockId,
    acpSelection         AudioClipSelector,
    acpMainFaderLevelNow AudioLevel,
    acpMainFaderLevelStart AudioLevel,
    acpMainFaderLevelStop AudioLevel,
    acpMainFaderDuration CardinalNumber,
    acpClipFaderLevelNow AudioLevel,
    acpClipFaderLevelStart AudioLevel,
    acpClipFaderLevelStop AudioLevel,
    acpClipFaderDuration CardinalNumber,
    acpStartDelay        IntegerNumber,
    acpActivation         AudioClipActivation,
    acpTrackControl       AudioClipTrackControl,
    acpTrackNumber        CardinalNumber,
    acpRepeat             AudioClipRepeat
}

AClipSequenceEntry ::= SEQUENCE {
    aClipSequenceBlockId BlockId,
    aClipSequenceId       AudioClipSelector,
    aClipSequenceName     Utf8String
}

ALimiterBlockEntry ::= SEQUENCE {
    alimiterBlockId      BlockId,
    alimiterThreshold     AudioLevel,
    alimiterAttackTime    CardinalNumber,
    alimiterGainMakeup    AudioLevel,
    alimiterRecoveryTime  CardinalNumber,
    alimiterRecoveryMode  AudioRecoveryMode
}

AConverterBlockEntry ::= SEQUENCE {
    aConverterBlockId     BlockId,
    aConverterQuality      AudioQuality,
    aConverterEnabled      TruthValue,
    aConverterDithering    TruthValue,
    aConverterOutputFormat MediaFormat,
    aConverterError        TruthValue
}

ALevelAlarmBlockEntry ::= SEQUENCE {
    alaBlockId           BlockId,
    alaType              AudioLevelAlarmType,
    alaThreshold          AudioLevel,
    alaWarningTime        CardinalNumber,
    alaFailureTime        CardinalNumber,
    alaCounter            CardinalNumber,
    alaEnabled            TruthValue,
    alaStatus             AudioAlarmStatus
}

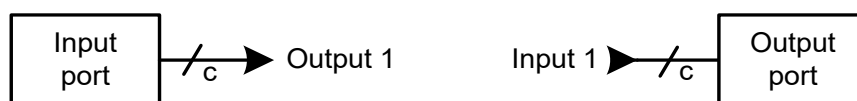
```

5.3 Audio port and associated managed object type definitions

5.3.1 Generic port functionality

All audio inputs to and outputs from the unit shall be represented using an audio port block. A base table of managed objects provides control common to all ports; extension tables provide functionality specific to certain types of port.

An audio port block shall have the following structure, where c is the number of channels on the input or output:



IEC 1492/08

Figure 1 – Audio port blocks

The group of objects in Table 1 shall be implemented by all compliant audio equipment that contains one or more audio ports. The root node for these objects shall be

{ iso(1) standard(0) iec62379 audio(2) audioMIB(1) audioPort(1) }

This node shall be used as the audio block type identifier for audio port blocks

Table 1 – Managed objects for audio ports

Identifier	Syntax	Index	Readable	Writable	Volatile	Status
aPortTable(1)	SEQUENCE OF APortEntry		none	none	no	m
lAPortEntry(1)	APortEntry		none	none	no	m
lAPortBlockId(1)	BlockId	yes	none	none	no	m
lAPortDirection(2)	PortDirection		listener	none	no	m
lAPortFormat(3)	MediaFormat		listener	none	yes	m
lAPortTransport(4)	AudioTransportType		listener	none	no	o
lAPortName(5)	Utf8String		listener	supervisor	no	o

5.3.1.1 aPortTable

A table of audio port descriptors for this unit. Each audio port on the unit has a corresponding entry in this table.

5.3.1.2 aPortEntry

An entry in the audio port table.

5.3.1.3 aPortBlockId

The block identifier for this port. Used as an index when accessing the audio port table.

5.3.1.4 aPortDirection

The direction (input or output) of this port.

5.3.1.5 aPortFormat

The format of the audio data currently being received or transmitted via this port. If the port is not active, the value `noAudio` shall be returned.

5.3.1.6 aPortTransport

The type of transport used by the port.

5.3.1.7 aPortName

The name assigned to this port. This is an arbitrary text string assigned by the system manager.

5.3.2 AES3 ancillary data

The group of objects in Table 2 shall be implemented by all compliant audio equipment that allows access to AES3 ancillary data. The root node for these objects shall be

{ iso(1) standard(0) iec62379 audio(2) audioMIB(1) audioPort(1) }

Table 2 – Managed objects for AES3 ancillary data

Identifier	Syntax	Index	Readable	Writable	Volatile	Status
aes3DataTable(2)	SEQUENCE OF Aes3DataEntry		none	none	no	m
└aes3DataEntry(1)	Aes3DataEntry		none	none	no	m
└└aes3BlockId(1)	BlockId	yes	none	none	no	m
└└aes3ChannelId(2)	AudioChannel	yes	none	none	no	m
└└└aes3ChannelData(3)	Aes3ChannelData		listener	none	yes	o
└└└aes3UserData(4)	Aes3UserData		listener	none	yes	o
└└└└aes3ValidityError(5)	TruthValue		listener	supervisor	yes	o

5.3.2.1 aes3DataTable

A table of AES3 ancillary data descriptors for this unit. Each channel of each AES3 port on the unit has a corresponding entry in this table.

5.3.2.2 aes3DataEntry

An entry in the AES3 ancillary data table.

5.3.2.3 aes3BlockId

The block identifier of the associated audio port. Used as an index when accessing the AES3 ancillary data table.

5.3.2.4 aes3ChannelId

The channel number. Used as an index when accessing the AES3 ancillary data table.

5.3.2.5 aes3ChannelData

The current channel status data (24 bytes) for this channel.

5.3.2.6 aes3UserData

The current user data for this channel.

5.3.2.7 aes3ValidityError

An indication that shows whether this channel's validity bit has been set in any subframe. It shall be automatically set to `true` when a subframe with V=1 is detected. It shall only be set to `false` as a result of a `SET` operation on this object.

NOTE V=1 indicates an error in the case of linear PCM; for other formats V=1 should be set in every subframe.

5.3.3 Phantom power

The group of objects in Table 3 shall be implemented by all compliant audio equipment that provides phantom power on an audio port. The root node for these objects shall be

```
{ iso(1) standard(0) iec62379 audio(2) audioMIB(1) audioPort(1) }
```

Table 3 – Managed objects for phantom power

Identifier	Syntax	Index	Readable	Writable	Volatile	Status
aPhantomTable(3)	SEQUENCE OF APhantomEntry		none	none	no	m
lAPhantomEntry(1)	APhantomEntry		none	none	no	m
lAPhantomBlockId(1)	BlockId	yes	none	none	no	m
lAPhantomEnabled(2)	TruthValue		listener	operator	no	m
lAPhantomLevel(3)	CardinalNumber		listener	supervisor	no	o

5.3.3.1 aPhantomTable

A table of audio port phantom power descriptors for this unit. Each audio port on the unit that supports phantom power has a corresponding entry in this table.

5.3.3.2 aPhantomEntry

An entry in the phantom power table.

5.3.3.3 aPhantomBlockId

The block identifier of the associated audio port. Used as an index when accessing the phantom power table.

5.3.3.4 aPhantomEnabled

When set to `true`, phantom power is provided by this port. When set to `false`, phantom power is not provided by this port.

5.3.3.5 aPhantomLevel

The level of the phantom power provided in millivolts.

5.3.4 Audio locked to reference

The group of objects in Table 4 shall be implemented by all compliant audio equipment that provides statistics for audio signals being locked to a reference signal. The root node for these objects shall be

```
{ iso(1) standard(0) iec62379 audio(2) audioMIB(1) audioPort(1) }
```

Table 4 – Managed objects for audio locked

Identifier	Syntax	Index	Readable	Writable	Volatile	Status
aLockedTable(3)	SEQUENCE OF ALockedEntry		none	none	no	m
LaLockedEntry(1)	ALockedEntry		none	none	no	m
LaLockedBlockId(1)	BlockId	yes	none	none	no	m
LaLockedTime(2)	CardinalNumber		listener	none	yes	m
LaLockedSamplesInserted(3)	CardinalNumber		listener	supervisor	yes	m
LaLockedSamplesDropped(4)	CardinalNumber		listener	supervisor	yes	m

5.3.4.1 aLockedTable

A table of audio port locked descriptors for this unit. Each audio port on the unit that supports measurement of lock against the reference has a corresponding entry in this table.

5.3.4.2 aLockedEntry

An entry in the locked table.

5.3.4.3 aLockedBlockId

The block identifier of the associated audio port. Used as an index when accessing the locked table.

5.3.4.4 aLockedTime

The time (in seconds) that the audio signal on this port has been locked to the unit's reference clock.

5.3.4.5 aLockedSamplesInserted

The number of audio samples that have been inserted due to buffer underrun or overrun on this port. The value shall be updated every 10 s by adding the number of samples inserted in the last 10 s to half the previous value.

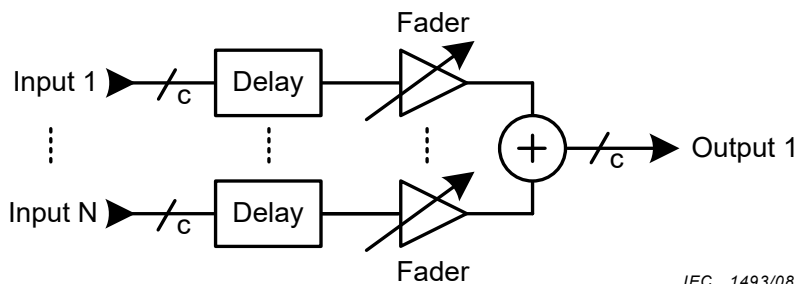
5.3.4.6 aLockedSamplesDropped

The number of audio samples that have been dropped due to buffer underrun or overrun on this port. The value shall be updated every 10 s by adding the number of samples inserted in the last 10 s to half the previous value.

5.4 Other audio block and associated managed object type definitions

5.4.1 Audio mixer blocks

An audio mixer block shall have the following structure:



IEC 1493/08

Figure 2 – Audio mixer block

where c is the number of channels on a connection.

NOTE 1 An audio mixer block may be used to represent a simple switched selector or combiner, by limiting the permitted values for the fader level controls to `mInfinity` or `fullScale`.

NOTE 2 The delay function permits audio streams that have passed through various processing or transport paths to be brought back into time alignment, either with other audio streams or with associated video streams. Equipment that does not support this functionality is represented as having a fixed zero delay.

The group of objects in Table 5 shall be implemented by all compliant audio equipment that has a management model that incorporates one or more audio mixer blocks. The root node for these objects shall be

{ iso(1) standard(0) iec62379 audio(2) audioMIB(1) audioMixer(2) }

This node shall be used as the block type identifier for audio mixer blocks.

Table 5 – Managed objects for audio mixer blocks

Identifier	Syntax	Index	Readable	Writable	Volatile	Status
<code>aMixerBlockTable(1)</code>	SEQUENCE OF <code>AMixerBlockEntry</code>		none	none	no	m
<code>laMixerBlockEntry(1)</code>	<code>AMixerBlockEntry</code>		none	none	no	m
<code>laMixerBlockId(1)</code>	<code>BlockId</code>	yes	none	none	no	m
<code>laMixerFadeDuration(2)</code>	<code>CardinalNumber</code>		listener	operator	no	o
<code>laMixerFadeNow(3)</code>	<code>TruthValue</code>		listener	operator	yes	o
<code>aMixerInputTable(2)</code>	SEQUENCE OF <code>AMixerInputEntry</code>		none	none	no	m
<code>laMixerInputEntry(1)</code>	<code>AMixerInputEntry</code>		none	none	no	m
<code>laMixerInputBlockId(1)</code>	<code>BlockId</code>	yes	none	none	no	m
<code>laMixerInputNumber(2)</code>	<code>IndexNumber</code>	yes	none	none	no	m
<code>laMixerInputLevel(3)</code>	<code>AudioLevel</code>		listener	operator	no	m
<code>laMixerInputFadeToLevel(4)</code>	<code>AudioLevel</code>		listener	operator	no	o
<code>laMixerInputDelay(5)</code>	<code>CardinalNumber</code>		listener	operator	no	o

5.4.1.1 aMixerBlockTable

A table of audio mixer block descriptors for this unit. Each audio mixer block in the unit has a corresponding entry in this table.

5.4.1.2 aMixerBlockEntry

An entry in the audio mixer block table.

5.4.1.3 **aMixerBlockId**

The block identifier for this block. Used as an index when accessing the audio mixer block table.

5.4.1.4 **aMixerFadeDuration**

The transition time (in milliseconds) for a smooth fade or cross-fade operation performed by this block. A value of zero indicates a hard switchover.

5.4.1.5 **aMixerFadeNow**

When set to `true`, causes the block to perform a smooth fade or cross-fade operation. Using values taken from this block's entries in the mixer input table, the unit shall simultaneously ramp the fader level for each input in `aMixerInputEntry` from its current value to the value specified by `aMixerInputFadeToLevel`. Automatically reset to `false` when the operation is complete.

5.4.1.6 **aMixerInputTable**

A table of audio mixer input descriptors for this unit. Each input of each audio mixer block in the unit has a corresponding entry in this table.

5.4.1.7 **aMixerInputEntry**

An entry in the audio mixer input table.

5.4.1.8 **aMixerInputBlockId**

The block identifier of the associated block. Used as an index when accessing the audio mixer input table.

5.4.1.9 **aMixerInputNumber**

The block input number for this input. Used as an index when accessing the audio mixer input table.

5.4.1.10 **aMixerInputLevel**

The fader level for this input. When a level is set, the fader immediately changes to the level specified, over the duration in `aMixerFadeDuration`. For blocks that only support switching between inputs, the only permitted values are `mInfinity` and `fullScale`. Blocks that automatically switch between inputs may reject `SET` operations on this object.

5.4.1.11 **aMixerInputFadeToLevel**

The fader level for this input that will be applied when `aMixerFadeNow` is set to `true`. For blocks that only support switching between inputs, the only permitted values are `mInfinity` and `fullScale`. Blocks that automatically switch between inputs may reject `SET` operations on this object.

5.4.1.12 **aMixerInputDelay**

The delay (in microseconds) applied to samples arriving at this input.

5.4.2 Audio crosspoint blocks

An audio crosspoint block shall have the following structure:

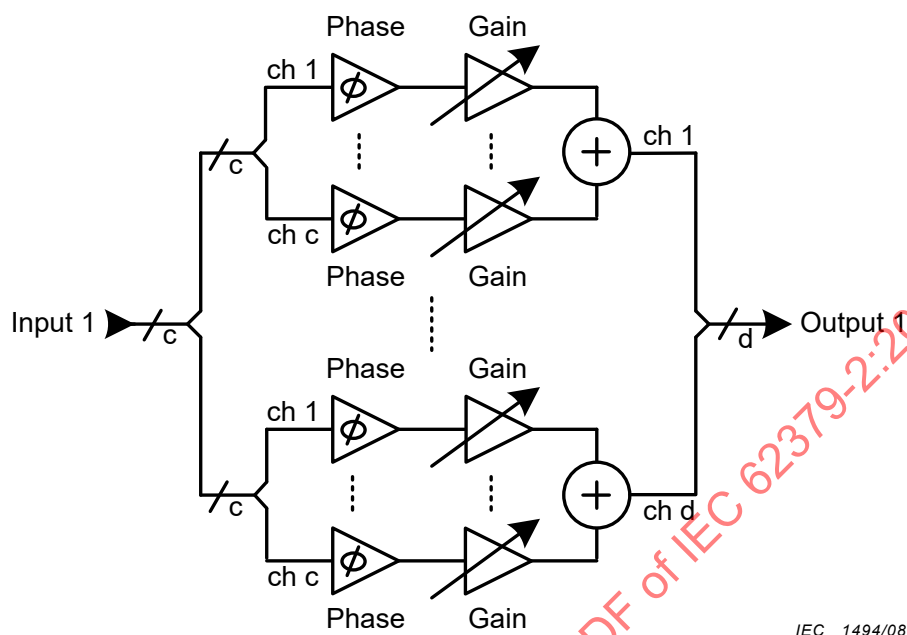


Figure 3 – Audio crosspoint block

where c is the number of input channels and d is the number of output channels.

The group of objects in Table 6 shall be implemented by all compliant audio equipment that has a management model that incorporates one or more audio crosspoint blocks. The root node for these objects shall be

{ iso(1) standard(0) iec62379 audio(2) audioMIB(1) audioCrosspoint(3) }

This node shall be used as the block type identifier for audio crosspoint blocks.

Table 6 – Managed objects for audio crosspoint blocks

Identifier	Syntax	Index	Readable	Writable	Volatile	Status
aCrosspointBlockTable(1)	SEQUENCE OF ACrosspointBlockEntry		none	none	no	m
lACrosspointBlockEntry(1)	ACrosspointBlockEntry		none	none	no	m
lACrosspointBlockId(1)	BlockId	yes	none	none	no	m
lACrosspointConfigure(2)	TruthValue		listener	operator	yes	m
lACrosspointCopy(3)	BlockId		none	operator	yes	o
aCrosspointPathTable(2)	SEQUENCE OF ACrosspointPathEntry		none	none	no	m
lACrosspointPathEntry(1)	ACrosspointPathEntry		none	none	no	m
lACrosspointPathBlockId(1)	BlockId	yes	none	none	no	m
lACrosspointPathSrc(2)	AudioChannel	yes	none	none	no	m
lACrosspointPathDst(3)	AudioChannel	yes	none	none	no	m
lACrosspointPathGain(4)	AudioLevel		listener	operator	no	m
lACrosspointPathNewGain(5)	AudioLevel		listener	operator	no	o
lACrosspointPathPhase(6)	AudioPhase		listener	operator	no	m
lACrosspointPathNewPhase(7)	AudioPhase		listener	operator	no	o

5.4.2.1 aCrosspointBlockTable

A table of audio crosspoint block descriptors for this unit. Each audio crosspoint block in the unit has a corresponding entry in this table.

5.4.2.2 aCrosspointBlockEntry

An entry in the audio crosspoint block table.

5.4.2.3 aCrosspointBlockId

The block identifier for this block. Used as an index when accessing the audio crosspoint block table.

5.4.2.4 aCrosspointConfigure

When set to `true`, causes this block to be configured as specified by the corresponding `aCrosspointPathNewGain` and `aCrosspointPathNewPhase` entries in the crosspoint path table. Automatically reset to `false` if the block supports delayed configuration and any of the corresponding entries in the crosspoint path table are modified. Permanently `true` if the block does not support delayed configuration.

5.4.2.5 aCrosspointCopy

When set to a block identifier that identifies another audio crosspoint block in the unit with an identical structure to this block, copies the values of `aCrosspointPathGain` and `aCrosspointPathPhase` for each path in the crosspoint from the identified block to this block. If set to a block identifier that does not identify an audio crosspoint block in the unit with an identical structure to this block, the SET operation shall be rejected.

NOTE A possible application is the ability to have some common configurations available as presets by creating 'dummy' crosspoint blocks with the required settings which are referenced in the block table but which are not actually part of the audio path.

5.4.2.6 aCrosspointPathTable

A table of audio crosspoint path descriptors for this unit. Each path in each audio crosspoint block in the unit has a corresponding entry in this table.

5.4.2.7 aCrosspointPathEntry

An entry in the audio crosspoint path table.

5.4.2.8 aCrosspointPathBlockId

The block identifier of the associated block. Used as an index when accessing the audio crosspoint path table.

5.4.2.9 aCrosspointPathSrc

The source audio channel associated with this path. Used as an index when accessing the audio crosspoint path table.

5.4.2.10 aCrosspointPathDst

The destination audio channel associated with this path. Used as an index when accessing the audio crosspoint path table.

5.4.2.11 aCrosspointPathGain

The signal gain from source to destination of this path. If this value is set, the change occurs immediately.

5.4.2.12 aCrosspointPathNewGain

The signal gain from source to destination of this path that will be applied when `aCrosspointConfigure` is set to `true`.

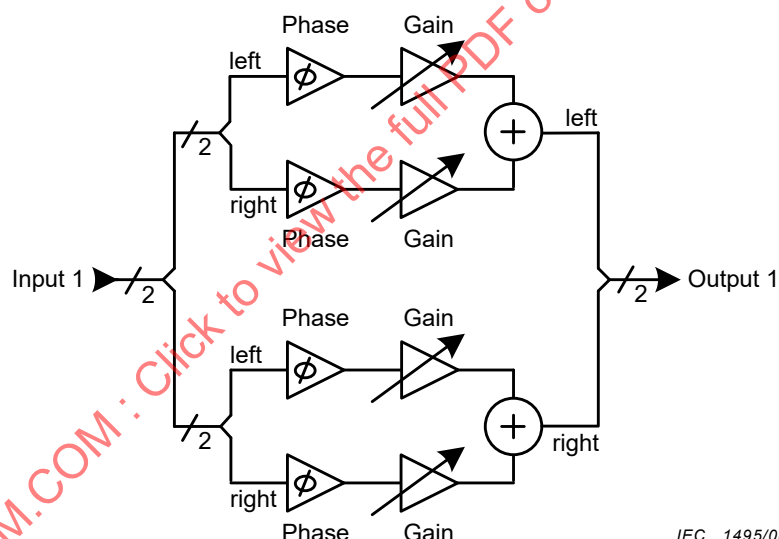
5.4.2.13 aCrosspointPathPhase

The phase shift from source to destination of this path. If this value is set, the change occurs immediately.

5.4.2.14 aCrosspointPathNewPhase

The phase shift from source to destination of this path that will be applied when `aCrosspointConfigure` is set to `true`.

EXAMPLE: The crosspoint block permits a variety of audio functions to be modelled. As an example, consider the following stereo crosspoint:



IEC 1495/08

There are four routes through the crosspoint representing what effect an input channel (the source, represented by `aCrosspointPathSrc`) has on an output channel (the destination, represented by `aCrosspointPathDst`). The source / destination pairs represent:

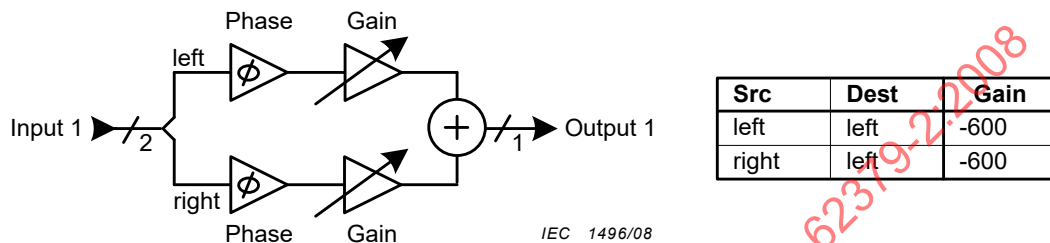
- the effect the left channel of the input has on the left channel of the output;
- the effect the right channel of the input has on the left channel of the output,
- the effect the left channel of the input has on the right channel of the output;
- the effect the right channel of the input has on the right channel of the output.

By setting different values for phase and gain for each source/destination pair the audio may be manipulated. For example, different gain values may be used to mono or leg-swap the incoming audio.

The following shows the fader settings (via `aCrosspointPathGain`) for different effects; *stereo* passes the audio unaltered, *swap* swaps the incoming left and right channels and *mono* places (M6) mono audio on both channels of the output.

Src	Dst	Gain		
		stereo	swap	mono
left	left	fullScale	mInfinity	-600
right	left	mInfinity	fullScale	-600
left	right	mInfinity	fullScale	-600
right	right	fullScale	mInfinity	-600

The number of input and output channels does not have to be the same, so the following models a block that takes a two channel stereo input to give a single channel mono output:



5.4.3 Audio clip player blocks

An audio clip player block shall have the following structure:

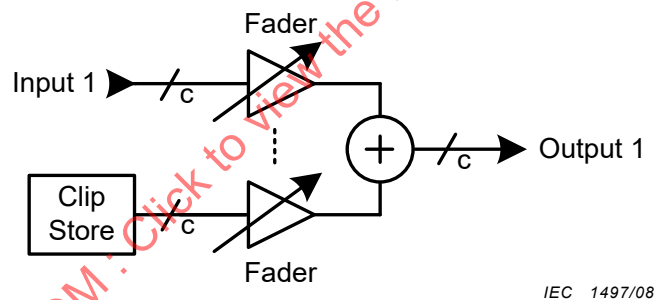


Figure 4 – Audio clip player block

where c is the number of channels on a connection. This represents a block that can either switch or mix a selected pre-stored audio sequence into an audio stream.

The group of objects in Table 7 shall be implemented by all compliant audio equipment that has a management model that incorporates one or more audio clip player blocks. The root node for these objects shall be

```
{ iso(1) standard(0) iec62379 audio(2) audioMIB(1) audioClipPlayer(4) }
```

This node shall be used as the block type identifier for audio clip player blocks.

Table 7 – Managed objects for audio clip player blocks

Identifier	Syntax	Index	Readable	Writable	Volatile	Status
aClipPlayerBlockTable(1)	SEQUENCE OF AClipPlayerBlockEntry		none	none	no	m
lAClipPlayerBlockEntry(1)	AClipPlayerBlockEntry		none	none	no	m
lACpBlockId(1)	BlockId	yes	none	none	no	m
lACpSelection(2)	AudioClipSelector		listener	operator	no	o
lACpMainFaderLevelNow(3)	AudioLevel		listener	operator	no	o
lACpMainFaderLevelStart(4)	AudioLevel		listener	operator	no	o
lACpMainFaderLevelStop(5)	AudioLevel		listener	operator	no	o
lACpMainFaderDuration(6)	CardinalNumber		listener	operator	no	o
lACpClipFaderLevelNow(7)	AudioLevel		listener	operator	no	o
lACpClipFaderLevelStart(8)	AudioLevel		listener	operator	no	o
lACpClipFaderLevelStop(9)	AudioLevel		listener	operator	no	o
lACpClipFaderDuration(10)	CardinalNumber		listener	operator	no	o
lACpStartDelay(11)	IntegerNumber		listener	operator	no	o
lACpActivation(12)	AudioClipActivation		listener	operator	yes	m
lACpTrackControl(13)	AudioClipTrackControl		none	operator	yes	o
lACpTrackNumber(14)	CardinalNumber		listener	operator	yes	o
lACpRepeat(15)	AudioClipRepeat		listener	operator	yes	o
aClipSequenceTable(2)	SEQUENCE OF AClipSequenceEntry		none	none	no	o
lAClipSequenceEntry(1)	AClipSequenceEntry		none	none	no	m
lAClipSequenceBlockId(1)	BlockId	yes	none	none	no	m
lAClipSequenceId(2)	AudioClipSelector	yes	none	none	no	m
lAClipSequenceName(3)	Utf8String		listener	operator	no	m

5.4.3.1 aClipPlayerBlockTable

A table of audio clip player block descriptors for this unit. Each audio clip player block in the unit has a corresponding entry in this table.

5.4.3.2 aClipPlayerBlockEntry

An entry in the audio clip player block table.

5.4.3.3 acpBlockId

The block identifier for this block. Used as an index when accessing the audio clip player block table.

5.4.3.4 acpSelection

The pre-stored sequence to be played by this block.

5.4.3.5 acpMainFaderLevelNow

The fader level for the main signal path. When a level is set, the fader immediately changes to the level specified, over the duration `acpMainFaderDuration`.

5.4.3.6 acpMainFaderLevelStart

The fader level for the main signal path when the clip player starts. The fader adopts this value automatically over the duration `acpMainFaderDuration`.

5.4.3.7 acpMainFaderLevelStop

The fader level for the main signal path when the clip player stops. The fader adopts this value automatically over the duration `acpMainFaderDuration`.

5.4.3.8 acpMainFaderDuration

The transition time (in milliseconds) for a smooth fade operation performed by the main fader. A value of zero indicates a hard switchover.

5.4.3.9 acpClipFaderLevelNow

The fader level for the pre-stored sequence. When a level is set, the fader immediately changes to the level specified, over the duration `acpClipFaderDuration`.

5.4.3.10 acpClipFaderLevelStart

The fader level for the pre-stored sequence when the clip player starts. The fader adopts this value automatically over the duration `acpClipFaderDuration`.

5.4.3.11 acpClipFaderLevelStop

The fader level for the pre-stored sequence when the clip player stops. The fader adopts this value automatically over the duration `acpClipFaderDuration`.

5.4.3.12 acpClipFaderDuration

The transition time (in milliseconds) for a smooth fade operation performed by the clip fader. A value of zero indicates a hard switchover.

5.4.3.13 acpStartDelay

The time offset (in milliseconds) between the clip player being activated and the pre-stored sequence being played. Equipment that supports scheduled operations may permit negative values.

5.4.3.14 acpActivation

When set to `play`, causes the clip player to start playing the selected pre-recorded sequence. Automatically `reset` to `stop` when playback is complete. Equipment may permit manual setting to `stop` to abort playback immediately, or to `stopAtEnd` to abort playback when the current track finishes. If the current state is `stopAtEnd`, setting to `stop` should cancel the stop at end of track request and stops playback immediately.

5.4.3.15 acpTrackControl

When set to `next`, causes the clip player to jump to the start of the next track in the selected sequence. If the clip player is playing the last track in the selected sequence, it may either jump to the first track in the sequence or reject the `SET` operation.

When set to `previous`, causes the clip player to jump to the start of the previous track in the selected sequence. If the clip player is playing the first track in the selected sequence, it may either jump to the last track in the sequence or reject the `SET` operation.

5.4.3.16 acpTrackNumber

The number of the track currently being played. If the selected clip is not divided into tracks, the value 1 shall be returned. If the clip player is inactive, the value 0 shall be returned.

When set to a valid track number for the selected sequence, causes the clip player to jump to the start of that track. If set to an invalid track number, the `SET` operation shall be rejected.

5.4.3.17 `acpTrackRepeat`

When set to `none`, the clip player shall automatically stop playback when the end of the selected sequence is reached. When set to `all`, the clip player shall automatically jump back to the start of the sequence whenever the end of the selected sequence is reached. When set to `track`, the clip player shall automatically jump back to the start of the track whenever the end of a track in the selected sequence is reached.

5.4.3.18 `aClipSequenceTable`

A table of audio clip sequence descriptors for this unit. Each audio clip sequence available for each audio clip player block in the unit has a corresponding entry in this table.

5.4.3.19 `aClipSequenceEntry`

An entry in the audio clip sequence table.

5.4.3.20 `aClipSequenceBlockId`

The block identifier for the associated audio clip player block. Used as an index when accessing the audio clip sequence table.

5.4.3.21 `aClipSequenceId`

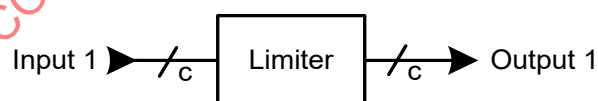
The clip sequence identifier. Used as an index when accessing the audio clip sequence table.

5.4.3.22 `aClipSequenceName`

The name assigned to this clip sequence. This is an arbitrary text string.

5.4.4 Audio limiter blocks

An audio limiter block shall have the following structure:



IEC 1498/08

Figure 5 – Audio limiter block

where `c` is the number of channels on a connection. This represents a block that can limit the incoming audio signal to a preset maximum level.

The group of objects in Table 8 shall be implemented by all compliant audio equipment that has a management model that incorporates one or more audio limiter blocks. The root node for these objects shall be

```
{ iso(1) standard(0) iec62379 audio(2) audioMIB(1) audioLimiter(5) }
```

This node shall be used as the block type identifier for audio limiter blocks.

Table 8 – Managed objects for audio limiter blocks

Identifier	Syntax	Index	Readable	Writable	Volatile	Status
aLimiterBlockTable(1)	SEQUENCE OF ALimiterBlockEntry		none	none	no	m
lALimiterBlockEntry(1)	ALimiterBlockEntry		none	none	no	m
lALimiterBlockId(1)	BlockId	yes	none	none	no	m
lALimiterThreshold(2)	AudioLevel		listener	supervisor	no	m
lALimiterAttackTime(3)	CardinalNumber		listener	supervisor	no	o
lALimiterGainMakeup(4)	AudioLevel		listener	supervisor	no	o
lALimiterRecoveryTime(5)	CardinalNumber		listener	supervisor	no	o
lALimiterRecoveryMode(6)	AudioRecoveryMode		listener	supervisor	no	o

5.4.4.1 aLimiterBlockTable

A table of audio limiter block descriptors for this unit. Each audio limiter block in the unit has a corresponding entry in this table.

5.4.4.2 aLimiterBlockEntry

An entry in the audio limiter block table.

5.4.4.3 aLimiterBlockId

The block identifier for this block. Used as an index when accessing the audio limiter block table.

5.4.4.4 aLimiterThreshold

The limiter threshold for this block. This describes the level at which the limiter starts to attenuate the input signal.

5.4.4.5 aLimiterAttackTime

The attack time (in milliseconds) for this block. This describes how long it takes for the limiter to start attenuating when the input signal exceeds the limiter threshold.

5.4.4.6 aLimiterGainMakeup

The gain makeup for this block. This describes the amount of gain applied to the signal after limiting has been applied.

5.4.4.7 aLimiterRecoveryTime

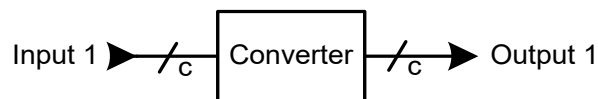
The recovery time (in milliseconds) for this block. This describes how long it takes for the limiter to stop attenuating when the input signal drops below the limiter threshold.

5.4.4.8 aLimiterRecoveryMode

The recovery mode for this block. This describes the way in which the limiter will recover after being activated. Three descriptions are currently used: *auto*, *slow* and *fast*, but how these are used depends upon the implementation within the equipment.

5.4.5 Audio converter blocks

An audio converter block shall have the following structure:



IEC 1499/08

Figure 6 – Audio converter block

where *c* is the number of channels on a connection. This represents a block that converts an incoming audio signal in one audio format to an outgoing audio signal in a different audio format.

NOTE This block may be used for any kind of conversion including the encoding and decoding of compressed formats.

The block's mode table shall be used to determine what format the converter should output; if only one mode is enabled then the converter block is forced to perform that conversion, if it is able. If more than one mode is enabled, the block should pick the output format according to its own implementation rules. If the block does not support any of the output formats that are enabled, it shall set `aConverterError` to `true`.

The group of objects in Table 9 shall be implemented by all compliant audio equipment that has a management model that incorporates one or more audio converter blocks. The root node for these objects shall be

```
{ iso(1) standard(0) iec62379 audio(2) audioMIB(1) audioConverter(6) }
```

This node shall be used as the block type identifier for audio converter blocks.

Table 9 – Managed objects for audio converter blocks

Identifier	Syntax	Index	Readable	Writable	Volatile	Status
<code>aConverterBlockTable</code> (1)	SEQUENCE OF <code>AConverterBlockEntry</code>		none	none	no	m
<code>LaConverterBlockEntry</code> (1)	<code>AConverterBlockEntry</code>		none	none	no	m
<code>LaConverterBlockId</code> (1)	<code>BlockId</code>	yes	none	none	no	m
<code>LaConverterQuality</code> (2)	<code>AudioQuality</code>		listener	supervisor	no	o
<code>LaConverterEnabled</code> (3)	<code>TruthValue</code>		listener	supervisor	no	o
<code>LaConverterDithering</code> (4)	<code>TruthValue</code>		listener	supervisor	no	o
<code>LaConverterOutputFormat</code> (5)	<code>MediaFormat</code>		listener	none	no	m
<code>LaConverterError</code> (6)	<code>TruthValue</code>		listener	none	no	o

5.4.5.1 `aConverterBlockTable`

A table of audio converter block descriptors for this unit. Each audio converter block in the unit has a corresponding entry in this table.

5.4.5.2 `aConverterBlockEntry`

An entry in the audio converter block table.

5.4.5.3 `aConverterBlockId`

The block identifier for this block. Used as an index when accessing the audio converter block table.

5.4.5.4 `aConverterQuality`

The quality of the conversion performed by this block.

5.4.5.5 aConverterEnabled

If `true`, indicates the incoming audio signal may be converted to one of the enabled audio formats for this block's output in the mode table. If `false`, indicates the incoming audio signal must be output in the same format as it arrives.

5.4.5.6 aConverterDithering

If `true`, indicates dithering is applied to the converted audio signal. If `false`, indicates no dithering is applied to the converted audio signal.

5.4.5.7 aConverterOutputFormat

A description of the current audio signal format being output by the converter block.

5.4.5.8 aConverterError

If `true`, indicates the audio cannot be converted to any of the output formats enabled in the mode table.

5.4.6 Audio level alarm blocks

An audio level alarm block shall have the following structure:

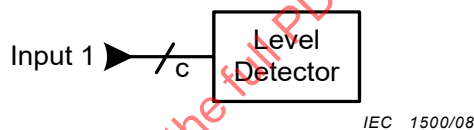


Figure 7 – Audio level alarm block

where `c` is the number of channels on a connection. This represents a block that detects audio level fault conditions in an audio stream.

NOTE An audio level alarm block may be used to represent, for example, an audio loss detector or an overload indicator.

The group of objects in Table 10 shall be implemented by all compliant audio equipment that has a management model that incorporates one or more audio level alarm blocks. The root node for these objects shall be

```
{ iso(1) standard(0) iec62379 audio(2) audioMIB(1) audioLevelAlarm(7) }
```

This node shall be used as the block type identifier for audio level alarm blocks.

Table 10 – Managed objects for audio level alarm blocks

Identifier	Syntax	Index	Readable	Writable	Volatile	Status
aLevelAlarmBlockTable(1)	SEQUENCE OF ALevelAlarmBlockEntry		none	none	no	m
lAlevelAlarmBlockEntry(1)	ALevelAlarmBlockEntry		none	none	no	m
lalaBlockId(1)	BlockId	yes	none	none	no	m
lalaType(2)	AudioLevelAlarmType		listener	supervisor	no	m
lalaThreshold(3)	AudioLevel		listener	supervisor	no	m
lalaWarningTime(4)	CardinalNumber		listener	supervisor	no	o
lalaFailureTime(5)	CardinalNumber		listener	supervisor	no	m
lalaCounter(6)	CardinalNumber		listener	supervisor	no	m
lalaEnabled(7)	TruthValue		listener	supervisor	no	m
lalaStatus(8)	AudioAlarmStatus		listener	none	yes	m

5.4.6.1 aLevelAlarmBlockTable

A table of audio level alarm block descriptors for this unit. Each audio level alarm block in the unit has a corresponding entry in this table.

5.4.6.2 aLevelAlarmBlockEntry

An entry in the audio level alarm block table.

5.4.6.3 alaBlockId

The block identifier for this block. Used as an index when accessing the audio level alarm block table.

5.4.6.4 alaType

The type of audio level alarm. If lower, the alarm is triggered by the audio level being less than the threshold; if higher, the alarm is triggered by the audio level being greater than the threshold.

5.4.6.5 alaThreshold

The detection threshold for an audio level alarm to be raised by this block.

5.4.6.6 alaWarningTime

The detection time (in seconds) for an audio level warning alarm to be raised by this block; see 5.4.6.10.

5.4.6.7 alaFailureTime

The detection time (in seconds) for an audio level failure alarm to be raised by this block; see 5.4.6.10.

5.4.6.8 alaCounter

The length of time (in seconds) for which the audio level has been in breach of (i.e. above or below, as determined by alaType) the detection threshold.

The counter shall be held at zero whenever the audio level is not in breach of the detection threshold.

The counter may be set by the management entity; if at the time of the `SET` request the audio is in breach of the detection threshold, the counter shall continue from the value set.

5.4.6.9 alaEnabled

Controls whether the alarm is primed for operation. If `true` warning and failure alarms will be raised when the audio level alarm counter breaches the warning and failure times respectively. If `false`, the alarm is disabled and no alarms will be raised.

5.4.6.10 alaStatus

Indicates the status of the alarm block. The value shall be `failure` if the audio level is in breach of the detection threshold and `alaCounter` is greater than or equal to `alaFailureTime`, otherwise `warning` if the audio level is in breach of the detection threshold and `alaCounter` is greater than or equal to `alaWarningTime`, otherwise `ok`.

6 Status broadcasts

6.1 General

NOTE Status broadcasts are specified in IEC 62379-1. This standard specifies audio-related status pages and groups.

6.2 Type definitions

In addition to the types defined in IEC 62379-1, the following types are used to specify the syntax of the abstract data structures used in status pages:

6.2.1 Textual conventions

```
AudioConverterStatus ::= OCTET STRING (SIZE(1))
-- a set of bits representing the status of an audio converter
-- bit 0 (lsb) = converter enabled (1=enabled, 0=not enabled)
-- bit 1       = dithering applied (1=dithering, 0=no dithering)
-- bit 2       = conversion ok (1=input converted, 0=converter error)
-- bit 3       = reserved
-- bit 4       = reserved
-- bit 5       = reserved
-- bit 6       = reserved
-- bit 7 (msb) = reserved
```

6.2.2 Sequences

```
AudioFormatsMapEntry ::= SEQUENCE {
    afmNumber  IndexNumber,
    afmFormat  MediaFormat
}
```

6.3 Audio formats mapping

NOTE 1 Due to the varying length of object identifiers, providing an indication of a particular audio format in a status page is problematic. Consequently, a table assigning an arbitrary id to an audio format is defined to allow a shorter, fixed length code to be associated with a format. The mapping chosen is equipment specific and management software should query the table with SNMP GET-NEXT requests on start-up to learn a device's map.

The group of objects in Table 11 shall be implemented by all compliant audio equipment that reports audio format information in status pages. The root node for these objects shall be

```
{ iso(1) standard(0) iec62379 audio(2) audioStatusMIB(4) }
```

Table 11 – Managed objects for audio format mappings

Identifier	Syntax	Index	Readable	Writable	Volatile	Status
audioFormatsMapTable(1)	SEQUENCE OF AudioFormatsMapEntry		none	none	no	m
↳audioFormatsMapEntry(1)	AudioFormatsMapEntry		none	none	no	m
↳afmNumber(1)	IndexNumber	yes	none	none	no	m
↳afmFormat(2)	MediaFormat		listener	supervisor	yes	m

6.3.1 audioFormatsMapTable

A table of audio format to index number mappings for this unit. Each mapping has a corresponding entry in this table.

6.3.2 audioFormatsMapEntry

An entry in audio formats map table.

6.3.3 afmNumber

The audio format number for this mapping.

6.3.4 afmFormat

The format of the audio data associated with this format number.

6.4 Page formats**6.4.1 Audio port page**

This page shall be produced by all audio ports and shall contain the entries in Table 12:

Table 12 – Status entries for audio port page

Octet(s)	Description	Value	Note(s)
1..2	Page number		
3..4	Block identifier	BlockId	1
5..8	Audio format	IndexNumber	2
9..10	Channel 1 peak level	AudioLevel	3
...	repeat previous entry for each remaining channel	...	4

NOTE 1 Coded with the block identifier for the block.

NOTE 2 Coded with the mapping from audioFormatsMapTable for the audio format of the port as given by aPortFormat.

NOTE 3 Coded with the audio peak level detected on the specified channel of the specified port since the last time this page was broadcast, in units of 0,01dB. A value of 0 represents full scale and values of 1 or greater indicate overload.

NOTE 4 The number of audio channels shall be inferred from the page length.

6.4.2 AES3 ancillary data page

This page shall be produced by all audio ports allowing access to AES3 ancillary data and shall contain the entries in Table 13:

Table 13 – Status entries for AES3 ancillary data page

Octet(s)	Description	Value	Note(s)
1..2	Page number		
3..4	Block identifier	BlockId	1
5..28	Channel 1 channel data	Aes3ChannelData	2
29..52	Channel 1 user data	Aes3UserData	3
53	Channel 1 validity error	TruthValue	4
...	repeat previous 3 entries for each remaining channel	...	5
NOTE 1 Coded with the block identifier for the block.			
NOTE 2 Coded with the current value of aes3ChannelData in the table entry associated with this channel.			
NOTE 3 Coded with the current value of aes3UserData in the table entry associated with this channel.			
NOTE 4 Coded with the current value of aes3ValidityError in the table entry associated with this channel.			
NOTE 5 The number of audio channels shall be inferred from the page length.			

6.4.3 Audio mixer page

This page shall be produced by all audio mixer blocks and shall contain the entries in Table 14:

Table 14 – Status entries for audio mixer page

Octet(s)	Description	Value	Note(s)
1..2	Page number		
3..4	Block identifier	BlockId	1
5..8	Input number	IndexNumber	2
9..12	Delay	CardinalNumber	3
13..14	Level	AudioLevel	4
...	repeat previous three entries for each input	...	5
NOTE 1 Coded with the block identifier for the block.			
NOTE 2 Coded with the number of the current input as given by aMixerInputNumber in the table entry associated with this mixer input.			
NOTE 3 Coded with the current value of aMixerInputDelay in the table entry associated with this mixer input.			
NOTE 4 Coded with the current value of aMixerInputLevel in the table entry associated with this mixer input.			
NOTE 5 The number of inputs to the mixer shall be inferred from the page length.			

6.4.4 Audio crosspoint page

This page shall be produced by all audio crosspoint blocks and shall contain the entries in Table 15:

Table 15 – Status entries for audio crosspoint page

Octet(s)	Description	Value	Note(s)
1..2	Page number		
3..4	Block identifier	BlockId	1
5..6	Crosspoint path source	AudioChannel	2
7..8	Crosspoint path destination	AudioChannel	3
9..10	Crosspoint path phase	AudioPhase	4
11..12	Crosspoint path gain	AudioLevel	5
...	repeat previous four entries for each path	...	6
NOTE 1 Coded with the block identifier for the block.			
NOTE 2 Coded with the source of the current path as given by aCrosspointPathSrc in the table entry associated with this crosspoint path.			
NOTE 3 Coded with the destination of the current path as given by aCrosspointPathDst in the table entry associated with this crosspoint path.			
NOTE 4 Coded with the current value of aCrosspointPathPhase in the table entry associated with this crosspoint path.			
NOTE 5 Coded with the current value of aCrosspointPathGain in the table entry associated with this crosspoint path.			
NOTE 6 The number of paths through the crosspoint shall be inferred from the page length.			

6.4.5 Audio clip player page

This page shall be produced by all audio clip player blocks and shall contain the entries in Table 16:

Table 16 – Status entries for audio clip player page

Octet(s)	Description	Value	Note(s)
1..2	Page number		
3..4	Block identifier	BlockId	1
5	Clip player activation	AudioClipActivation	2
6..9	Clip player selection	AudioClipSelector	3
10..13	Clip player track number	CardinalNumber	4
14..15	Main fader level	AudioLevel	5
16..17	Clip player fader level	AudioLevel	6
NOTE 1 Coded with the block identifier for the block.			
NOTE 2 Coded with the current value of acpActivation in the table entry associated with this clip player.			
NOTE 3 Coded with the current value of acpSelection in the table entry associated with this clip player.			
NOTE 4 Coded with the current value of acpTrackNumber in the table entry associated with this clip player.			
NOTE 5 Coded with the current value of acpMainFaderLevelNow in the table entry associated with this clip player.			
NOTE 6 Coded with the current value of acpClipFaderLevelNow in the table entry associated with this clip player.			

6.4.6 Audio limiter page

This page shall be produced by all audio limiter blocks and shall contain the entries in Table 17:

Table 17 – Status entries for audio limiter page

Octet(s)	Description	Value	Note(s)
1..2	Page number		
3..4	Block identifier	BlockId	1
5..6	Limiter threshold	AudioLevel	2
7..10	Limiter attack time	CardinalNumber	3
11..12	Limiter gain makeup	AudioLevel	4
13..16	Limiter recovery time	CardinalNumber	5
17	Limiter recovery mode	AudioRecoveryMode	6
NOTE 1 Coded with the block identifier for the block.			
NOTE 2 Coded with the current value of aLimiterThreshold in the table entry associated with this limiter.			
NOTE 3 Coded with the current value of aLimiterAttackTime in the table entry associated with this limiter.			
NOTE 4 Coded with the current value of aLimiterGainMakeup in the table entry associated with this limiter.			
NOTE 5 Coded with the current value of aLimiterRecoveryTime in the table entry associated with this limiter.			
NOTE 6 Coded with the current value of aLimiterRecoveryMode in the table entry associated with this limiter.			

6.4.7 Audio converter page

This page shall be produced by all audio converter blocks and shall contain the entries in Table 18:

Table 18 – Status entries for audio converter page

Octet(s)	Description	Value	Note(s)
1..2	Page number		
3..4	Block identifier	BlockId	1
5	Converter status	AudioConverterStatus	2
6..9	Output format	IndexNumber	3
NOTE 1 Coded with the block identifier for the block.			
NOTE 2 Coded with the current value of the converter status.			
NOTE 3 Coded with the mapping from audioFormatsMapTable for the output audio format of the converter as given by aConverterOutputFormat.			

6.4.8 Audio level alarm page

This page shall be produced by all audio level alarm blocks and shall contain the entries in Table 19:

Table 19 – Status entries for audio level alarm page

Octet(s)	Description	Value	Note(s)
1..2	Page number		
3..4	Block identifier	BlockId	1
5	Alarm enabled	TruthValue	2
6	Alarm status	AudioAlarmStatus	3
7..10	Counter	CardinalNumber	4
11..12	Audio threshold	AudioLevel	5
13..16	Warning time	CardinalNumber	6
17..20	Failure time	CardinalNumber	7
NOTE 1 Coded with the block identifier for the block.			
NOTE 2 Coded with the current value of <code>alaEnabled</code> in the table entry associated with the audio level alarm.			
NOTE 3 Coded with the current value of <code>alaStatus</code> in the table entry associated with the audio level alarm.			
NOTE 4 Coded with the current value of <code>alaCounter</code> in the table entry associated with the audio level alarm.			
NOTE 5 Coded with the current value of <code>alaThreshold</code> in the table entry associated with the audio level alarm.			
NOTE 6 Coded with the current value of <code>alaWarningTime</code> in the table entry associated with the audio level alarm.			
NOTE 7 Coded with the current value of <code>alaFailureTime</code> in the table entry associated with the audio level alarm.			

6.5 Page groups

6.5.1 `audioPorts`

This group shall be supported by all audio units with ports. The root node for this group shall be

```
{ iso(1) standard(0) iec62379 audio(2) audioStatusGroup(3) audioPorts(1) }
```

A broadcast of this group shall consist of the following:

- page 1: Audio port, broadcast at the base page rate;
- page 2: AES3 ancillary data, broadcast at half the base page rate.

The default base page rate for this group shall be 60 pages per minute (1 page every second).

6.5.2 `standardAudioBlocks`

This group shall be supported by all audio units with some blocks defined in this standard. The root node for this group shall be

```
{ iso(1) standard(0) iec62379 audio(2) audioStatusGroup(3) standardAudioBlocks(2) }
```

A broadcast of this group shall consist of the following:

- page 1: Audio mixer, broadcast at the base page rate;
- page 2: Audio crosspoint, broadcast at the base page rate;
- page 3: Audio clip player, broadcast at the base page rate;
- page 4: Audio limiter, broadcast at the base page rate;
- page 5: Audio converter, broadcast at the base page rate.

The default base page rate for this group shall be 60 pages per minute (1 page every second).

6.5.3 audioAlarms

This group shall be supported by all audio units with alarms. The root node for this group shall be

{ iso(1) standard(0) iec62379 audio(2) audioStatusGroup(3) audioAlarms (3) }

A broadcast of this group shall consist of the following:

- page 1: Audio Level Alarm, broadcast at the base page rate.

The default base page rate for this group shall be 60 pages per minute (1 page every second).

IECNORM.COM : Click to view the full PDF of IEC 62379-2:2008

Annex A (informative)

Machine-readable audio format definitions

This annex provides a machine-readable version of the audio data format definitions which is intended to be interpretable by standard MIB browsing software tools. If there is any inconsistency between this annex and Clause 4, Clause 4 takes precedence.

The format used to describe the data format identifiers conforms to IETF STD 58 (SMIv2).

NOTE 1 This annex is not intended to cover every format permitted by the definitions in Clause 4.

```
IEC62379-2-FORMATS DEFINITIONS ::= BEGIN

IMPORTS

    MODULE-IDENTITY, OBJECT-IDENTITY
        FROM SNMPv2-SMI

    iec62379
        FROM IEC62379-1-MIB;

audioFormat MODULE-IDENTITY
    LAST-UPDATED "200804301500Z"
    ORGANIZATION "IEC PT62379"
    CONTACT-INFO "Not specified"
    DESCRIPTION "The audio format identifiers defined in clause 4 of
        IEC 62379-2."
    REVISION "200804301500Z"
    DESCRIPTION "Edition 1.0 final draft."
    ::= { audio 2 }

audio OBJECT IDENTIFIER ::= { iec62379 2 }

-- 4.1 Audio signal format definitions
-- 4.1.1 Audio parameters

ChannelArrangement ::= TEXTUAL-CONVENTION
    STATUS current
    DESCRIPTION "The audio channel arrangement in the audio format."
    SYNTAX INTEGER (0..5)
    -- {
    --     unspecified          (0),
    --     discreteMono         (1),
    --     stereo                (2),
    --     jointStereo           (3),
    --     surround              (4),
    --     surroundWithDownmix   (5)
    -- } (unspecified..surroundWithDownmix)

NumberChannels ::= TEXTUAL-CONVENTION
    STATUS current
    DESCRIPTION "An integer representing the number of audio channels."
    SYNTAX INTEGER (0..2147483647)

BitDepth ::= TEXTUAL-CONVENTION
    STATUS current
    DESCRIPTION "An integer representing the audio bit depth in bits per
        sample."
    SYNTAX INTEGER (0..2147483647)
```

```

SamplingFrequency ::= TEXTUAL-CONVENTION
    STATUS          current
    DESCRIPTION     "An integer representing the audio sample frequency in Hz."
    SYNTAX          INTEGER (0..2147483647)

BitRate ::= TEXTUAL-CONVENTION
    STATUS          current
    DESCRIPTION     "An integer representing the audio bit rate in bits per second."
    SYNTAX          INTEGER (0..2147483647)

-- 4.1.2 Audio signal formats

audioSignalFormat OBJECT IDENTIFIER ::= { audioFormat 1 }

-- 4.1.2.1 Unspecified audio

unspecifiedAudio OBJECT-IDENTITY
    STATUS          current
    DESCRIPTION     "AudioFormat wildcard value - any supported format allowed."
    ::= { audioSignalFormat 0 }

-- 4.1.2.2 No audio

noAudio OBJECT-IDENTITY
    STATUS          current
    DESCRIPTION     "Indicates the output is muted."
    ::= { audioSignalFormat 1 }

-- 4.1.2.3 Analogue audio

analogueAudio OBJECT-IDENTITY
    STATUS          current
    DESCRIPTION     "Analogue audio."
    ::= { audioSignalFormat 2 }

-- analogue discrete mono

analogueMono OBJECT-IDENTITY
    STATUS          current
    DESCRIPTION     "Discrete mono analogue audio."
    ::= { analogueAudio 1 }

analogueMono1Chan OBJECT-IDENTITY
    STATUS          current
    DESCRIPTION     "Discrete mono analogue audio with 1 channel."
    ::= { analogueMono 1 }

analogueMono2Chan OBJECT-IDENTITY
    STATUS          current
    DESCRIPTION     "Discrete mono analogue audio with 2 channels."
    ::= { analogueMono 2 }

analogueMono3Chan OBJECT-IDENTITY
    STATUS          current
    DESCRIPTION     "Discrete mono analogue audio with 3 channels."
    ::= { analogueMono 3 }

analogueMono4Chan OBJECT-IDENTITY
    STATUS          current
    DESCRIPTION     "Discrete mono analogue audio with 4 channels."
    ::= { analogueMono 4 }

-- analogue stereo

```

```
analogueStereo OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "Stereo analogue audio."
  ::= { analogueAudio 2 }

analogueStereo2Chan OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "Stereo analogue audio with 2 channels (1 L-R pair).\"
  ::= { analogueStereo 2 }

analogueStereo4Chan OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "Stereo analogue audio with 4 channels (2 L-R pairs).\"
  ::= { analogueStereo 4}

-- analogue jointstereo

analogueJointStereo OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "Joint stereo analogue audio.\"
  ::= { analogueAudio 3 }

analogueJointStereo2Chan OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "Joint stereo analogue audio with 2 channels (1 M-S pair).\"
  ::= { analogueJointStereo 2 }

analogueJointStereo4Chan OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "Joint stereo analogue audio with 4 channels (2 M-S pairs).\"
  ::= { analogueJointStereo 4}

-- 4.1.2.4 PCM audio

pcmAudio OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "Linear PCM audio.\"
  ::= { audioSignalFormat 3 }

-- PCM discrete mono

pcmMono OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "Discrete mono linear PCM audio.\"
  ::= { pcmAudio 1 }

pcmMono1Chan OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "Discrete mono linear PCM audio with 1 channel.\"
  ::= { pcmMono 1 }

pcmMono1Chan16 OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "16-bit discrete mono linear PCM audio with 1 channel.\"
  ::= { pcmMono1Chan 16 }

pcmMono1Chan16at44100 OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "44.1k 16-bit discrete mono linear PCM audio with 1 channel.\"
  ::= { pcmMono1Chan16 44100 }

pcmMono1Chan16at48000 OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "48k 16-bit discrete mono linear PCM audio with 1 channel.\"
  ::= { pcmMono1Chan16 48000 }

pcmMono1Chan16at96000 OBJECT-IDENTITY
```

```

STATUS          current
DESCRIPTION      "96k 16-bit discrete mono linear PCM audio with 1 channel."
::= { pcmMono1Chan16 96000 }

pcmMono1Chan20 OBJECT-IDENTITY
STATUS          current
DESCRIPTION      "20-bit discrete mono linear PCM audio with 1 channel."
::= { pcmMono1Chan 20 }

pcmMono1Chan20at44100 OBJECT-IDENTITY
STATUS          current
DESCRIPTION      "44.1k 20-bit discrete mono linear PCM audio with 1 channel."
::= { pcmMono1Chan20 44100 }

pcmMono1Chan20at48000 OBJECT-IDENTITY
STATUS          current
DESCRIPTION      "48k 20-bit discrete mono linear PCM audio with 1 channel."
::= { pcmMono1Chan20 48000 }

pcmMono1Chan20at96000 OBJECT-IDENTITY
STATUS          current
DESCRIPTION      "96k 20-bit discrete mono linear PCM audio with 1 channel."
::= { pcmMono1Chan20 96000 }

pcmMono1Chan24 OBJECT-IDENTITY
STATUS          current
DESCRIPTION      "24-bit discrete mono linear PCM audio with 1 channel."
::= { pcmMono1Chan 24 }

pcmMono1Chan24at44100 OBJECT-IDENTITY
STATUS          current
DESCRIPTION      "44.1k 24-bit discrete mono linear PCM audio with 1 channel."
::= { pcmMono1Chan24 44100 }

pcmMono1Chan24at48000 OBJECT-IDENTITY
STATUS          current
DESCRIPTION      "48k 24-bit discrete mono linear PCM audio with 1 channel."
::= { pcmMono1Chan24 48000 }

pcmMono1Chan24at96000 OBJECT-IDENTITY
STATUS          current
DESCRIPTION      "96k 24-bit discrete mono linear PCM audio with 1 channel."
::= { pcmMono1Chan24 96000 }

pcmMono2Chan OBJECT-IDENTITY
STATUS          current
DESCRIPTION      "Discrete mono linear PCM audio with 2 channels."
::= { pcmMono 2 }

pcmMono2Chan16 OBJECT-IDENTITY
STATUS          current
DESCRIPTION      "16-bit discrete mono linear PCM audio with 2 channels."
::= { pcmMono2Chan 16 }

pcmMono2Chan16at44100 OBJECT-IDENTITY
STATUS          current
DESCRIPTION      "44.1k 16-bit discrete mono linear PCM audio with 2 channels."
::= { pcmMono2Chan16 44100 }

pcmMono2Chan16at48000 OBJECT-IDENTITY
STATUS          current
DESCRIPTION      "48k 16-bit discrete mono linear PCM audio with 2 channels."
::= { pcmMono2Chan16 48000 }

pcmMono2Chan16at96000 OBJECT-IDENTITY
STATUS          current
DESCRIPTION      "96k 16-bit discrete mono linear PCM audio with 2 channels."
::= { pcmMono2Chan16 96000 }

pcmMono2Chan20 OBJECT-IDENTITY

```

```
STATUS          current
DESCRIPTION     "20-bit discrete mono linear PCM audio with 2 channels."
::= { pcmMono2Chan 20 }

pcmMono2Chan20at44100 OBJECT-IDENTITY
STATUS          current
DESCRIPTION     "44.1k 20-bit discrete mono linear PCM audio with 2 channels."
::= { pcmMono2Chan20 44100 }

pcmMono2Chan20at48000 OBJECT-IDENTITY
STATUS          current
DESCRIPTION     "48k 20-bit discrete mono linear PCM audio with 2 channels."
::= { pcmMono2Chan20 48000 }

pcmMono2Chan20at96000 OBJECT-IDENTITY
STATUS          current
DESCRIPTION     "96k 20-bit discrete mono linear PCM audio with 2 channels."
::= { pcmMono2Chan20 96000 }

pcmMono2Chan24 OBJECT-IDENTITY
STATUS          current
DESCRIPTION     "24-bit discrete mono linear PCM audio with 2 channels."
::= { pcmMono2Chan 24 }

pcmMono2Chan24at44100 OBJECT-IDENTITY
STATUS          current
DESCRIPTION     "44.1k 24-bit discrete mono linear PCM audio with 2 channels."
::= { pcmMono2Chan24 44100 }

pcmMono2Chan24at48000 OBJECT-IDENTITY
STATUS          current
DESCRIPTION     "48k 24-bit discrete mono linear PCM audio with 2 channels."
::= { pcmMono2Chan24 48000 }

pcmMono2Chan24at96000 OBJECT-IDENTITY
STATUS          current
DESCRIPTION     "96k 24-bit discrete mono linear PCM audio with 2 channels."
::= { pcmMono2Chan24 96000 }

-- PCM stereo

pcmStereo OBJECT-IDENTITY
STATUS          current
DESCRIPTION     "Stereo linear PCM audio."
::= { pcmAudio 2 }

pcmStereo2Chan OBJECT-IDENTITY
STATUS          current
DESCRIPTION     "Stereo linear PCM audio with 2 channels (1 L-R pair)."
::= { pcmStereo 2 }

pcmStereo2Chan16 OBJECT-IDENTITY
STATUS          current
DESCRIPTION     "16-bit stereo linear PCM audio with 2 channels."
::= { pcmStereo2Chan 16 }

pcmStereo2Chan16at44100 OBJECT-IDENTITY
STATUS          current
DESCRIPTION     "44.1k 16-bit stereo linear PCM audio with 2 channels."
::= { pcmStereo2Chan16 44100 }

pcmStereo2Chan16at48000 OBJECT-IDENTITY
STATUS          current
DESCRIPTION     "48k 16-bit stereo linear PCM audio with 2 channels."
::= { pcmStereo2Chan16 48000 }

pcmStereo2Chan16at96000 OBJECT-IDENTITY
STATUS          current
DESCRIPTION     "96k 16-bit stereo linear PCM audio with 2 channels."
```

```

::= { pcmStereo2Chan16 96000 }

pcmStereo2Chan20 OBJECT-IDENTITY
    STATUS          current
    DESCRIPTION      "20-bit stereo linear PCM audio with 2 channels."
    ::= { pcmStereo2Chan 20 }

pcmStereo2Chan20at44100 OBJECT-IDENTITY
    STATUS          current
    DESCRIPTION      "44.1k 20-bit stereo linear PCM audio with 2 channels."
    ::= { pcmStereo2Chan20 44100 }

pcmStereo2Chan20at48000 OBJECT-IDENTITY
    STATUS          current
    DESCRIPTION      "48k 20-bit stereo linear PCM audio with 2 channels."
    ::= { pcmStereo2Chan20 48000 }

pcmStereo2Chan20at96000 OBJECT-IDENTITY
    STATUS          current
    DESCRIPTION      "96k 20-bit stereo linear PCM audio with 2 channels."
    ::= { pcmStereo2Chan20 96000 }

pcmStereo2Chan24 OBJECT-IDENTITY
    STATUS          current
    DESCRIPTION      "24-bit stereo linear PCM audio with 2 channels."
    ::= { pcmStereo2Chan 24 }

pcmStereo2Chan24at44100 OBJECT-IDENTITY
    STATUS          current
    DESCRIPTION      "44.1k 24-bit stereo linear PCM audio with 2 channels."
    ::= { pcmStereo2Chan24 44100 }

pcmStereo2Chan24at48000 OBJECT-IDENTITY
    STATUS          current
    DESCRIPTION      "48k 24-bit stereo linear PCM audio with 2 channels."
    ::= { pcmStereo2Chan24 48000 }

pcmStereo2Chan24at96000 OBJECT-IDENTITY
    STATUS          current
    DESCRIPTION      "96k 24-bit stereo linear PCM audio with 2 channels."
    ::= { pcmStereo2Chan24 96000 }

-- PCM joint stereo

pcmJointStereo OBJECT-IDENTITY
    STATUS          current
    DESCRIPTION      "Joint stereo linear PCM audio."
    ::= { pcmAudio 3 }

pcmJointStereo2Chan OBJECT-IDENTITY
    STATUS          current
    DESCRIPTION      "Joint stereo linear PCM audio with 2 channels (1 M-S pair)."
    ::= { pcmJointStereo 2 }

pcmJointStereo2Chan16 OBJECT-IDENTITY
    STATUS          current
    DESCRIPTION      "16-bit joint stereo linear PCM audio with 2 channels."
    ::= { pcmJointStereo2Chan 16 }

pcmJointStereo2Chan20 OBJECT-IDENTITY
    STATUS          current
    DESCRIPTION      "20-bit joint stereo linear PCM audio with 2 channels."
    ::= { pcmJointStereo2Chan 20 }

pcmJointStereo2Chan24 OBJECT-IDENTITY
    STATUS          current
    DESCRIPTION      "24-bit joint stereo linear PCM audio with 2 channels."
    ::= { pcmJointStereo2Chan 24 }

```

```
-- 4.1.2.5 MPEG-1 layer 2 audio

mp2Audio OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "MPEG1 layer 2 audio."
  ::= { audioSignalFormat 4 }

-- MP2 discrete mono

mp2Mono OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "Discrete mono MP2 audio."
  ::= { mp2Audio 1 }

mp2Mono1Chan OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "Discrete mono MP2 audio with 1 channel."
  ::= { mp2Mono 1 }

-- MP2 discrete mono with 1 channel at 44100

mp2Mono1Chan44100 OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "44.1k discrete mono MP2 audio with 1 channel."
  ::= { mp2Mono1Chan 44100 }

mp2Mono1Chan44100at32 OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "44.1k discrete mono MP2 audio at 32kb/s with 1 channel."
  ::= { mp2Mono1Chan44100 32000 }

mp2Mono1Chan44100at40 OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "44.1k discrete mono MP2 audio at 40kb/s with 1 channel."
  ::= { mp2Mono1Chan44100 40000 }

mp2Mono1Chan44100at48 OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "44.1k discrete mono MP2 audio at 48kb/s with 1 channel."
  ::= { mp2Mono1Chan44100 48000 }

mp2Mono1Chan44100at56 OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "44.1k discrete mono MP2 audio at 56kb/s with 1 channel."
  ::= { mp2Mono1Chan44100 56000 }

mp2Mono1Chan44100at64 OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "44.1k discrete mono MP2 audio at 64kb/s with 1 channel."
  ::= { mp2Mono1Chan44100 64000 }

mp2Mono1Chan44100at80 OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "44.1k discrete mono MP2 audio at 80kb/s with 1 channel."
  ::= { mp2Mono1Chan44100 80000 }

mp2Mono1Chan44100at96 OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "44.1k discrete mono MP2 audio at 96kb/s with 1 channel."
  ::= { mp2Mono1Chan44100 96000 }

mp2Mono1Chan44100at112 OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "44.1k discrete mono MP2 audio at 112kb/s with 1 channel."
  ::= { mp2Mono1Chan44100 112000 }

mp2Mono1Chan44100at128 OBJECT-IDENTITY
```

```

STATUS          current
DESCRIPTION      "44.1k discrete mono MP2 audio at 128kb/s with 1 channel."
::= { mp2Mono1Chan44100 128000 }

mp2Mono1Chan44100at160 OBJECT-IDENTITY
STATUS          current
DESCRIPTION      "44.1k discrete mono MP2 audio at 160kb/s with 1 channel."
::= { mp2Mono1Chan44100 160000 }

mp2Mono1Chan44100at192 OBJECT-IDENTITY
STATUS          current
DESCRIPTION      "44.1k discrete mono MP2 audio at 192kb/s with 1 channel."
::= { mp2Mono1Chan44100 192000 }

mp2Mono1Chan44100at224 OBJECT-IDENTITY
STATUS          current
DESCRIPTION      "44.1k discrete mono MP2 audio at 224kb/s with 1 channel."
::= { mp2Mono1Chan44100 224000 }

mp2Mono1Chan44100at256 OBJECT-IDENTITY
STATUS          current
DESCRIPTION      "44.1k discrete mono MP2 audio at 256kb/s with 1 channel."
::= { mp2Mono1Chan44100 256000 }

mp2Mono1Chan44100at320 OBJECT-IDENTITY
STATUS          current
DESCRIPTION      "44.1k discrete mono MP2 audio at 320kb/s with 1 channel."
::= { mp2Mono1Chan44100 320000 }

-- MP2 discrete mono with 1 channel at 48000

mp2Mono1Chan48000 OBJECT-IDENTITY
STATUS          current
DESCRIPTION      "48k mono MP2 audio with 1 channel."
::= { mp2Mono1Chan 48000 }

mp2Mono1Chan48000at32 OBJECT-IDENTITY
STATUS          current
DESCRIPTION      "48k discrete mono MP2 audio at 32kb/s with 1 channel."
::= { mp2Mono1Chan48000 32000 }

mp2Mono1Chan48000at40 OBJECT-IDENTITY
STATUS          current
DESCRIPTION      "48k discrete mono MP2 audio at 40kb/s with 1 channel."
::= { mp2Mono1Chan48000 40000 }

mp2Mono1Chan48000at48 OBJECT-IDENTITY
STATUS          current
DESCRIPTION      "48k discrete mono MP2 audio at 48kb/s with 1 channel."
::= { mp2Mono1Chan48000 48000 }

mp2Mono1Chan48000at56 OBJECT-IDENTITY
STATUS          current
DESCRIPTION      "48k discrete mono MP2 audio at 56kb/s with 1 channel."
::= { mp2Mono1Chan48000 56000 }

mp2Mono1Chan48000at64 OBJECT-IDENTITY
STATUS          current
DESCRIPTION      "48k discrete mono MP2 audio at 64kb/s with 1 channel."
::= { mp2Mono1Chan48000 64000 }

mp2Mono1Chan48000at80 OBJECT-IDENTITY
STATUS          current
DESCRIPTION      "48k discrete mono MP2 audio at 80kb/s with 1 channel."
::= { mp2Mono1Chan48000 80000 }

mp2Mono1Chan48000at96 OBJECT-IDENTITY
STATUS          current
DESCRIPTION      "48k discrete mono MP2 audio at 96kb/s with 1 channel."

```

```
::= { mp2Mono1Chan48000 96000 }

mp2Mono1Chan48000at112 OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "48k discrete mono MP2 audio at 112kb/s with 1 channel."
  ::= { mp2Mono1Chan48000 112000 }

mp2Mono1Chan48000at128 OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "48k discrete mono MP2 audio at 128kb/s with 1 channel."
  ::= { mp2Mono1Chan48000 128000 }

mp2Mono1Chan48000at160 OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "48k discrete mono MP2 audio at 160kb/s with 1 channel."
  ::= { mp2Mono1Chan48000 160000 }

mp2Mono1Chan48000at192 OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "48k discrete mono MP2 audio at 192kb/s with 1 channel."
  ::= { mp2Mono1Chan48000 192000 }

mp2Mono1Chan48000at224 OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "48k discrete mono MP2 audio at 224kb/s with 1 channel."
  ::= { mp2Mono1Chan48000 224000 }

mp2Mono1Chan48000at256 OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "48k discrete mono MP2 audio at 256kb/s with 1 channel."
  ::= { mp2Mono1Chan48000 256000 }

mp2Mono1Chan48000at320 OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "48k discrete mono MP2 audio at 320kb/s with 1 channel."
  ::= { mp2Mono1Chan48000 320000 }

-- MP2 stereo

mp2Stereo OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "Stereo MP2 audio."
  ::= { mp2Audio 2 }

mp2Stereo2Chan OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "Stereo MP2 audio with 2 channels (1 L-R pair)."
  ::= { mp2Stereo 2 }

-- MP2 stereo with 2 channels at 44100

mp2Stereo2Chan44100 OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "44.1k stereo MP2 audio."
  ::= { mp2Stereo2Chan 44100 }

mp2Stereo2Chan44100at32 OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "44.1k stereo MP2 audio at 32kb/s."
  ::= { mp2Stereo2Chan44100 32000 }

mp2Stereo2Chan44100at40 OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "44.1k stereo MP2 audio at 40kb/s."
  ::= { mp2Stereo2Chan44100 40000 }

mp2Stereo2Chan44100at48 OBJECT-IDENTITY
  STATUS      current
```

```

DESCRIPTION    "44.1k stereo MP2 audio at 48kb/s."
::= { mp2Stereo2Chan44100 48000 }

mp2Stereo2Chan44100at56 OBJECT-IDENTITY
STATUS          current
DESCRIPTION     "44.1k stereo MP2 audio at 56kb/s."
::= { mp2Stereo2Chan44100 56000 }

mp2Stereo2Chan44100at64 OBJECT-IDENTITY
STATUS          current
DESCRIPTION     "44.1k stereo MP2 audio at 64kb/s."
::= { mp2Stereo2Chan44100 64000 }

mp2Stereo2Chan44100at80 OBJECT-IDENTITY
STATUS          current
DESCRIPTION     "44.1k stereo MP2 audio at 80kb/s."
::= { mp2Stereo2Chan44100 80000 }

mp2Stereo2Chan44100at96 OBJECT-IDENTITY
STATUS          current
DESCRIPTION     "44.1k stereo MP2 audio at 96kb/s."
::= { mp2Stereo2Chan44100 96000 }

mp2Stereo2Chan44100at112 OBJECT-IDENTITY
STATUS          current
DESCRIPTION     "44.1k stereo MP2 audio at 112kb/s."
::= { mp2Stereo2Chan44100 112000 }

mp2Stereo2Chan44100at128 OBJECT-IDENTITY
STATUS          current
DESCRIPTION     "44.1k stereo MP2 audio at 128kb/s."
::= { mp2Stereo2Chan44100 128000 }

mp2Stereo2Chan44100at160 OBJECT-IDENTITY
STATUS          current
DESCRIPTION     "44.1k stereo MP2 audio at 160kb/s."
::= { mp2Stereo2Chan44100 160000 }

mp2Stereo2Chan44100at192 OBJECT-IDENTITY
STATUS          current
DESCRIPTION     "44.1k stereo MP2 audio at 192kb/s."
::= { mp2Stereo2Chan44100 192000 }

mp2Stereo2Chan44100at224 OBJECT-IDENTITY
STATUS          current
DESCRIPTION     "44.1k stereo MP2 audio at 224kb/s."
::= { mp2Stereo2Chan44100 224000 }

mp2Stereo2Chan44100at256 OBJECT-IDENTITY
STATUS          current
DESCRIPTION     "44.1k stereo MP2 audio at 256kb/s."
::= { mp2Stereo2Chan44100 256000 }

mp2Stereo2Chan44100at320 OBJECT-IDENTITY
STATUS          current
DESCRIPTION     "44.1k stereo MP2 audio at 320kb/s."
::= { mp2Stereo2Chan44100 320000 }

-- MP2 stereo with 2 channels at 48000

mp2Stereo2Chan48000 OBJECT-IDENTITY
STATUS          current
DESCRIPTION     "48k stereo MP2 audio."
::= { mp2Stereo2Chan 48000 }

mp2Stereo2Chan48000at32 OBJECT-IDENTITY
STATUS          current
DESCRIPTION     "48k stereo MP2 audio at 32kb/s."
::= { mp2Stereo2Chan48000 32000 }

```

```
mp2Stereo2Chan48000at40 OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "48k stereo MP2 audio at 40kb/s."
  ::= { mp2Stereo2Chan48000 40000 }

mp2Stereo2Chan48000at48 OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "48k stereo MP2 audio at 48kb/s."
  ::= { mp2Stereo2Chan48000 48000 }

mp2Stereo2Chan48000at56 OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "48k stereo MP2 audio at 56kb/s."
  ::= { mp2Stereo2Chan48000 56000 }

mp2Stereo2Chan48000at64 OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "48k stereo MP2 audio at 64kb/s."
  ::= { mp2Stereo2Chan48000 64000 }

mp2Stereo2Chan48000at80 OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "48k stereo MP2 audio at 80kb/s."
  ::= { mp2Stereo2Chan48000 80000 }

mp2Stereo2Chan48000at96 OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "48k stereo MP2 audio at 96kb/s."
  ::= { mp2Stereo2Chan48000 96000 }

mp2Stereo2Chan48000at112 OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "48k stereo MP2 audio at 112kb/s."
  ::= { mp2Stereo2Chan48000 112000 }

mp2Stereo2Chan48000at128 OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "48k stereo MP2 audio at 128kb/s."
  ::= { mp2Stereo2Chan48000 128000 }

mp2Stereo2Chan48000at160 OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "48k stereo MP2 audio at 160kb/s."
  ::= { mp2Stereo2Chan48000 160000 }

mp2Stereo2Chan48000at192 OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "48k stereo MP2 audio at 192kb/s."
  ::= { mp2Stereo2Chan48000 192000 }

mp2Stereo2Chan48000at224 OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "48k stereo MP2 audio at 224kb/s."
  ::= { mp2Stereo2Chan48000 224000 }

mp2Stereo2Chan48000at256 OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "48k stereo MP2 audio at 256kb/s."
  ::= { mp2Stereo2Chan48000 256000 }

mp2Stereo2Chan48000at320 OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "48k stereo MP2 audio at 320kb/s."
  ::= { mp2Stereo2Chan48000 320000 }
```

-- MP2 joint stereo

```
mp2JointStereo OBJECT-IDENTITY
```

```

STATUS          current
DESCRIPTION      "Joint stereo MP2 audio."
::= { mp2Audio 3 }

mp2JointStereo2Chan OBJECT-IDENTITY
STATUS          current
DESCRIPTION      "Joint stereo MP2 audio with 2 channels (1 M-S pair)."
```

```

::= { mp2JointStereo 2 }

mp2JointStereo2Chan44100 OBJECT-IDENTITY
STATUS          current
DESCRIPTION      "44.1k joint stereo MP2 audio."
::= { mp2JointStereo2Chan 44100 }

mp2JointStereo2Chan48000 OBJECT-IDENTITY
STATUS          current
DESCRIPTION      "48k joint stereo MP2 audio."
::= { mp2JointStereo2Chan 48000 }
```

-- 4.1.2.6 MPEG-1 layer 3 audio

```

mp3Audio OBJECT-IDENTITY
STATUS          current
DESCRIPTION      "MPEG1 layer 3 audio."
::= { audioSignalFormat 5 }

mp3Mono OBJECT-IDENTITY
STATUS          current
DESCRIPTION      "Discrete mono MP3 audio."
::= { mp3Audio 1 }

mp3Mono1Chan OBJECT-IDENTITY
STATUS          current
DESCRIPTION      "Discrete mono MP3 audio with 1 channel."
::= { mp3Mono 1 }

mp3Stereo OBJECT-IDENTITY
STATUS          current
DESCRIPTION      "Stereo MP3 audio."
::= { mp3Audio 2 }

mp3Stereo2Chan OBJECT-IDENTITY
STATUS          current
DESCRIPTION      "Stereo MP3 audio with 2 channels (1 L-R pair)."
```

```

::= { mp3Stereo 2 }

mp3JointStereo OBJECT-IDENTITY
STATUS          current
DESCRIPTION      "Joint stereo MP3 audio."
::= { mp3Audio 3 }

mp3JointStereo2Chan OBJECT-IDENTITY
STATUS          current
DESCRIPTION      "Joint stereo MP3 audio with 2 channels (1 M-S pair)."
```

```

::= { mp3JointStereo 2 }

-- 4.1.2.7 AAC audio

aacAudio OBJECT-IDENTITY
STATUS          current
DESCRIPTION      "aac audio."
::= { audioSignalFormat 6 }

aacLC OBJECT-IDENTITY
STATUS          current
DESCRIPTION      "aac audio with the low complexity profile."
::= { aacAudio 1 }
```

```
aacMain OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "aac audio with the main profile."
  ::= { aacAudio 2 }

aacSRS OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "aac audio with the sample-rate-scalable profile."
  ::= { aacAudio 3 }

aacLTP OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "aac audio with the long term prediction profile."
  ::= { aacAudio 4 }

aacLD OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "aac audio with the low delay profile."
  ::= { aacAudio 5 }

-- 4.1.2.8 G711 audio

g711Audio OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "G711 audio."
  ::= { audioSignalFormat 7 }

g711ALaw OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "G711 audio denoting A-Law encoding."
  ::= { g711Audio 1 }

g711MuLaw OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "G711 audio denoting mu-Law encoding."
  ::= { g711Audio 2 }

-- 4.1.2.9 G722 audio

g722Audio OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "G722 audio."
  ::= { audioSignalFormat 8 }

g722at48 OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "G722 audio at 48kbits/s."
  ::= { g722Audio 48 }

g722at56 OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "G722 audio at 56kbits/s."
  ::= { g722Audio 56 }

g722at64 OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "G722 audio at 64kbits/s."
  ::= { g722Audio 64 }

-- 4.1.2.10 apt-X audio

aptXAudio OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "APT-X audio."
  ::= { audioSignalFormat 9 }
```

```
-- 4.1.2.11 enhanced apt-X audio

enhAptXAudio OBJECT-IDENTITY
    STATUS          current
    DESCRIPTION      "Enhanced APT-X audio."
    ::= { audioSignalFormat 10 }

-- 4.1.2.12 J41 audio

j41Audio OBJECT-IDENTITY
    STATUS          current
    DESCRIPTION      "J41 audio."
    ::= { audioSignalFormat 11 }

j41ALawA OBJECT-IDENTITY
    STATUS          current
    DESCRIPTION      "J41 audio using A-law companding, variant A."
    ::= { j41Audio 1 }

j41ALawB OBJECT-IDENTITY
    STATUS          current
    DESCRIPTION      "J41 audio using A-law companding, variant B."
    ::= { j41Audio 2 }

j41Nic OBJECT-IDENTITY
    STATUS          current
    DESCRIPTION      "J41 audio using near instantaneous companding."
    ::= { j41Audio 3 }

-- 4.1.2.13 J57 audio

j57Audio OBJECT-IDENTITY
    STATUS          current
    DESCRIPTION      "J57 audio."
    ::= { audioSignalFormat 12 }

j57H11 OBJECT-IDENTITY
    STATUS          current
    DESCRIPTION      "J57 using an H11 channel."
    ::= { j57Audio 1 }

j57H12 OBJECT-IDENTITY
    STATUS          current
    DESCRIPTION      "J57 using an H12 channel."
    ::= { j57Audio 2 }

-- 4.1.2.14 Invalid audio

invalidAudio OBJECT-IDENTITY
    STATUS          current
    DESCRIPTION      "Indicates an error, such as an inability to decode a signal
                     earlier in the chain."
    ::= { audioSignalFormat 13 }

-- 4.2 Audio transport format definitions

audioTransportFormat OBJECT IDENTIFIER ::= { audioFormat transport(2) }

unspecifiedTransport OBJECT-IDENTITY
    STATUS          current
    DESCRIPTION      "Unspecified transport."
    ::= { audioTransportFormat 0 }

analogue OBJECT-IDENTITY
    STATUS          current
    DESCRIPTION      "Analogue transport."
```

```
 ::= { audioTransportFormat 1 }

aes3 OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "AES3 transport."
  ::= { audioTransportFormat 2 }

aes10 OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "AES10 transport."
  ::= { audioTransportFormat 3 }

aes50 OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "AES50 transport."
  ::= { audioTransportFormat 4 }

-- Metadata

audioMetadataFormat OBJECT IDENTIFIER ::= { audioFormat 3 }

unspecifiedMetadata OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "Unspecified metadata."
  ::= { audioMetadataFormat 0 }

END
```

IECNORM.COM : Click to view the full PDF of IEC 62379-2:2008

Annex B (informative)

Machine-readable audio block definitions

This annex provides a machine-readable version of the audio block definitions which is intended to be interpretable by standard MIB browsing software tools. It does not express all the requirements of the standard, for instance where access to an object is restricted at certain privilege levels. If there is any inconsistency between this annex and Clause 5, Clause 5 takes precedence.

The format used to describe the MIB objects conforms to IETF STD 58 (SMIv2).

```
IEC62379-2-MIB DEFINITIONS ::= BEGIN

IMPORTS

    MODULE-IDENTITY, OBJECT-TYPE
        FROM SNMPv2-SMI

    OBJECT-GROUP, MODULE-COMPLIANCE
        FROM SNMPv2-CONF

    TEXTUAL-CONVENTION, TruthValue
        FROM SNMPv2-TC

    IntegerNumber, CardinalNumber, IndexNumber, Utf8String, BlockId,
    MediaFormat, PortDirection, iec62379, OperationId, ObjectName,
    ObjectValue, PrivilegeLevel, OperationState
        FROM IEC62379-1-MIB;

audioMIB MODULE-IDENTITY
    LAST-UPDATED "200804301500Z"
    ORGANIZATION "IEC PT62379"
    CONTACT-INFO "Not specified"
    DESCRIPTION "The MIB module for managing audio functions in IEC 62379
        compliant equipment."
    REVISION "200804301500Z"
    DESCRIPTION "Edition 1.0 final draft."
    ::= { audio 1 }

audio OBJECT IDENTIFIER ::= { iec62379 2 }

-- Object identifier values for module compliance statements
audioMIBCompliance OBJECT IDENTIFIER ::= { audioMIB 0 }

-- Object identifier values for MIB object groups
-- the following definitions are also used as BlockType values

audioPort          OBJECT IDENTIFIER ::= { audioMIB 1 }

audioMixer          OBJECT IDENTIFIER ::= { audioMIB 2 }

audioCrosspoint     OBJECT IDENTIFIER ::= { audioMIB 3 }

audioClipPlayer     OBJECT IDENTIFIER ::= { audioMIB 4 }

audioLimiter         OBJECT IDENTIFIER ::= { audioMIB 5 }

audioConverter       OBJECT IDENTIFIER ::= { audioMIB 6 }

audioLevelAlarm     OBJECT IDENTIFIER ::= { audioMIB 7 }
```

-- 5.2 Type definitions

-- 5.2.1 Textual conventions

```
AudioTransportType ::= TEXTUAL-CONVENTION
    STATUS          current
    DESCRIPTION     "A reference to the transport used for an audio connection."
    SYNTAX          OBJECT IDENTIFIER

AudioLevel ::= TEXTUAL-CONVENTION
    STATUS          current
    DESCRIPTION     "An absolute or relative audio level in units of 0.01dB."
    SYNTAX          INTEGER (-20000..20000)
                    -- {
                    --   mInfinity (-20000),
                    --   fullScale (0),
                    --   pInfinity (20000)
                    -- } (mInfinity..pInfinity)

AudioPhase ::= TEXTUAL-CONVENTION
    STATUS          current
    DESCRIPTION     "An absolute or relative phase value in units of 0.01 degree."
    SYNTAX          INTEGER (-18000..18000)

AudioQuality ::= TEXTUAL-CONVENTION
    STATUS          current
    DESCRIPTION     "An enumeration identifying an audio signal processing quality
                    level. Semantics are equipment specific. Equipment that
                    supports selectable quality levels shall as a minimum support
                    the values low and high."
    SYNTAX          INTEGER (1..127)
                    -- {
                    --   worst (1),
                    --   low (32),
                    --   high (96),
                    --   best (127)
                    -- } (worst..best)

AudioChannel ::= TEXTUAL-CONVENTION
    STATUS          current
    DESCRIPTION     "An enumeration identifying an audio channel. Stereo equipment
                    shall accept the values left and right. Multi-channel equipment
                    should accept a contiguous range of values starting from 1 and
                    may disregard the association of the values 1 and 2 with left
                    and right."
    SYNTAX          INTEGER (1..240)
                    -- {
                    --   left (1),
                    --   right (2)
                    -- } (1..240)

Aes3ChannelData ::= TEXTUAL-CONVENTION
    STATUS          current
    DESCRIPTION     "The channel status data associated with an AES3 channel."
    SYNTAX          OCTET STRING (SIZE(24))

Aes3UserData ::= TEXTUAL-CONVENTION
    STATUS          current
    DESCRIPTION     "The user status data associated with an AES3 channel."
    SYNTAX          OCTET STRING (SIZE(24))

AudioClipSelector ::= TEXTUAL-CONVENTION
    STATUS          current
    DESCRIPTION     "An enumeration identifying an audio clip. Semantics are
                    equipment specific."
    SYNTAX          INTEGER (1..2147483647)

AudioClipActivation ::= TEXTUAL-CONVENTION
    STATUS          current
```

```
DESCRIPTION  "An enumeration for audio clip playback control."
SYNTAX       INTEGER {
                play      (1),
                stop      (2),
                stopAtEnd (3)
            } -- (play..stopAtEnd)
```

AudioClipTrackControl ::= TEXTUAL-CONVENTION

```
STATUS       current
DESCRIPTION   "An enumeration for audio clip track control."
SYNTAX       INTEGER {
                previous(1),
                next    (2)
            } -- (previous..next)
```

AudioClipRepeat ::= TEXTUAL-CONVENTION

```
STATUS       current
DESCRIPTION   "An enumeration for audio clip repeat control."
SYNTAX       INTEGER {
                none  (1),
                track (2),
                all   (3)
            } -- (none..all)
```

AudioRecoveryMode ::= TEXTUAL-CONVENTION

```
STATUS       current
DESCRIPTION   "An enumeration identifying an audio limiter recovery mode.
                Semantics are equipment specific."
SYNTAX       INTEGER {
                auto (1),
                slow (2),
                fast (3)
            } -- (auto..fast)
```

AudioAlarmStatus ::= TEXTUAL-CONVENTION

```
STATUS       current
DESCRIPTION   "An enumeration representing an alarm's status"
SYNTAX       INTEGER {
                ok      (1),
                warning (2),
                failure (3)
            } -- (ok..failure)
```

AudioLevelAlarmType ::= TEXTUAL-CONVENTION

```
STATUS       current
DESCRIPTION   "An enumeration representing the trigger condition for an audio
                level alarm. If lower, the alarm is triggered by the audio
                level being less than the threshold; if higher, the alarm is
                triggered by the audio level being greater than the threshold"
SYNTAX       INTEGER {
                lower (1),
                higher (2)
            } -- (lower..higher)
```

-- 5.2.3 Sequences

APortEntry ::= SEQUENCE {

```
    aPortBlockId    BlockId,
    aPortDirection   PortDirection,
    aPortFormat       MediaFormat,
    aPortTransport    AudioTransportType,
    aPortName         Utf8String
}
```

Aes3DataEntry ::= SEQUENCE {

```
    aes3BlockId      BlockId,
    aes3ChannelId     AudioChannel,
    aes3ChannelData   Aes3ChannelData,
    aes3UserData      Aes3UserData,
```

```

    aes3ValidityError TruthValue
}

APhantomEntry ::= SEQUENCE {
    aPhantomBlockId BlockId,
    aPhantomEnabled TruthValue,
    aPhantomLevel CardinalNumber
}

ALockedEntry ::= SEQUENCE {
    aLockedBlockId BlockId,
    aLockedTime CardinalNumber,
    aLockedSamplesInserted CardinalNumber,
    aLockedSamplesDropped CardinalNumber
}

AMixerBlockEntry ::= SEQUENCE {
    aMixerBlockId BlockId,
    aMixerFadeDuration CardinalNumber,
    aMixerFadeNow TruthValue
}

AMixerInputEntry ::= SEQUENCE {
    aMixerInputBlockId BlockId,
    aMixerInputNumber IndexNumber,
    aMixerInputLevel AudioLevel,
    aMixerInputFadeToLevel AudioLevel,
    aMixerInputDelay CardinalNumber
}

ACrosspointBlockEntry ::= SEQUENCE {
    aCrosspointBlockId BlockId,
    aCrosspointConfigure TruthValue,
    aCrosspointCopy BlockId
}

ACrosspointPathEntry ::= SEQUENCE {
    aCrosspointPathBlockId BlockId,
    aCrosspointPathSrc AudioChannel,
    aCrosspointPathDst AudioChannel,
    aCrosspointPathGain AudioLevel,
    aCrosspointPathNewGain AudioLevel,
    aCrosspointPathPhase AudioPhase,
    aCrosspointPathNewPhase AudioPhase
}

AClipPlayerBlockEntry ::= SEQUENCE {
    acpBlockId BlockId,
    acpSelection AudioClipSelector,
    acpMainFaderLevelNow AudioLevel,
    acpMainFaderLevelStart AudioLevel,
    acpMainFaderLevelStop AudioLevel,
    acpMainFaderDuration CardinalNumber,
    acpClipFaderLevelNow AudioLevel,
    acpClipFaderLevelStart AudioLevel,
    acpClipFaderLevelStop AudioLevel,
    acpClipFaderDuration CardinalNumber,
    acpStartDelay IntegerNumber,
    acpActivation AudioClipActivation,
    acpTrackControl AudioClipTrackControl,
    acpTrackNumber CardinalNumber,
    acpRepeat AudioClipRepeat
}

AClipSequenceEntry ::= SEQUENCE {
    aClipSequenceBlockId BlockId,
    aClipSequenceId AudioClipSelector,
    aClipSequenceName Utf8String
}

```

```
ALimiterBlockEntry ::= SEQUENCE {
    aLimiterBlockId      BlockId,
    aLimiterThreshold    AudioLevel,
    aLimiterAttackTime   CardinalNumber,
    aLimiterGainMakeup   AudioLevel,
    aLimiterRecoveryTime CardinalNumber,
    aLimiterRecoveryMode AudioRecoveryMode
}
```

```
AConverterBlockEntry ::= SEQUENCE {
    aConverterBlockId      BlockId,
    aConverterQuality      AudioQuality,
    aConverterEnabled      TruthValue,
    aConverterDithering    TruthValue,
    aConverterOutputFormat MediaFormat,
    aConverterError        TruthValue
}
```

```
ALevelAlarmBlockEntry ::= SEQUENCE {
    alaBlockId      BlockId,
    alaType         AudioLevelAlarmType,
    alaThreshold    AudioLevel,
    alaWarningTime  CardinalNumber,
    alaFailureTime  CardinalNumber,
    alaCounter      CardinalNumber,
    alaEnabled      TruthValue,
    alaStatus       AudioAlarmStatus
}
```

-- 5.3 Audio port and associated managed object type definitions

-- 5.3.1 Generic port functionality

```
aPortTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF APortEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "See IEC 62379-2 clause 5.3.1."
    ::= { audioPort 1 }
```

```
aPortEntry OBJECT-TYPE
    SYNTAX      APortEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "See IEC 62379-2 clause 5.3.1."
    INDEX       { aPortBlockId }
    ::= { aPortTable 1 }
```

```
aPortBlockId OBJECT-TYPE
    SYNTAX      BlockId
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "See IEC 62379-2 clause 5.3.1."
    ::= { aPortEntry 1 }
```

```
aPortDirection OBJECT-TYPE
    SYNTAX      PortDirection
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "See IEC 62379-2 clause 5.3.1."
    ::= { aPortEntry 2 }
```

```
aPortFormat OBJECT-TYPE
    SYNTAX      MediaFormat
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "See IEC 62379-2 clause 5.3.1."
    ::= { aPortEntry 3 }
```

```

aPortTransport OBJECT-TYPE
    SYNTAX      AudioTransportType
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION  "See IEC 62379-2 clause 5.3.1."
    ::= { aPortEntry 4 }

```

```

aPortName OBJECT-TYPE
    SYNTAX      Utf8String
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION  "See IEC 62379-2 clause 5.3.1."
    ::= { aPortEntry 5 }

```

-- 5.3.2 AES3 ancillary data

```

aes3DataTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Aes3DataEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION  "See IEC 62379-2 clause 5.3.2."
    ::= { audioPort 2 }

```

```

aes3DataEntry OBJECT-TYPE
    SYNTAX      Aes3DataEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION  "See IEC 62379-2 clause 5.3.2."
    INDEX       { aes3BlockId, aes3ChannelId }
    ::= { aes3DataTable 1 }

```

```

aes3BlockId OBJECT-TYPE
    SYNTAX      BlockId
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION  "See IEC 62379-2 clause 5.3.2."
    ::= { aes3DataEntry 1 }

```

```

aes3ChannelId OBJECT-TYPE
    SYNTAX      AudioChannel
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION  "See IEC 62379-2 clause 5.3.2."
    ::= { aes3DataEntry 2 }

```

```

aes3ChannelData OBJECT-TYPE
    SYNTAX      Aes3ChannelData
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION  "See IEC 62379-2 clause 5.3.2."
    ::= { aes3DataEntry 3 }

```

```

aes3UserData OBJECT-TYPE
    SYNTAX      Aes3UserData
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION  "See IEC 62379-2 clause 5.3.2."
    ::= { aes3DataEntry 4 }

```

```

aes3ValidityError OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION  "See IEC 62379-2 clause 5.3.2."
    ::= { aes3DataEntry 5 }

```

-- 5.3.3 Phantom power

aPhantomTable OBJECT-TYPE
 SYNTAX SEQUENCE OF APhantomEntry
 MAX-ACCESS not-accessible
 STATUS current
 DESCRIPTION "See IEC 62379-2 clause 5.3.3."
 ::= { audioPort 3 }

aPhantomEntry OBJECT-TYPE
 SYNTAX APhantomEntry
 MAX-ACCESS not-accessible
 STATUS current
 DESCRIPTION "See IEC 62379-2 clause 5.3.3."
 INDEX { aPhantomBlockId }
 ::= { aPhantomTable 1 }

aPhantomBlockId OBJECT-TYPE
 SYNTAX BlockId
 MAX-ACCESS not-accessible
 STATUS current
 DESCRIPTION "See IEC 62379-2 clause 5.3.3."
 ::= { aPhantomEntry 1 }

aPhantomEnabled OBJECT-TYPE
 SYNTAX TruthValue
 MAX-ACCESS read-write
 STATUS current
 DESCRIPTION "See IEC 62379-2 clause 5.3.3."
 ::= { aPhantomEntry 2 }

aPhantomLevel OBJECT-TYPE
 SYNTAX CardinalNumber
 MAX-ACCESS read-write
 STATUS current
 DESCRIPTION "See IEC 62379-2 clause 5.3.3."
 ::= { aPhantomEntry 3 }

-- 5.3.4 Audio locked to Reference

aLockedTable OBJECT-TYPE
 SYNTAX SEQUENCE OF ALockedEntry
 MAX-ACCESS not-accessible
 STATUS current
 DESCRIPTION "See IEC 62379-2 clause 5.3.4."
 ::= { audioPort 4 }

aLockedEntry OBJECT-TYPE
 SYNTAX ALockedEntry
 MAX-ACCESS not-accessible
 STATUS current
 DESCRIPTION "See IEC 62379-2 clause 5.3.4."
 INDEX { aLockedBlockId }
 ::= { aLockedTable 1 }

aLockedBlockId OBJECT-TYPE
 SYNTAX BlockId
 MAX-ACCESS not-accessible
 STATUS current
 DESCRIPTION "See IEC 62379-2 clause 5.3.4."
 ::= { aLockedEntry 1 }

aLockedTime OBJECT-TYPE
 SYNTAX CardinalNumber
 MAX-ACCESS read-only
 STATUS current
 DESCRIPTION "See IEC 62379-2 clause 5.3.4."
 ::= { aLockedEntry 2 }

aLockedSamplesInserted OBJECT-TYPE
 SYNTAX CardinalNumber

```

MAX-ACCESS      read-write
STATUS          current
DESCRIPTION     "See IEC 62379-2 clause 5.3.4."
::= { aLockedEntry 3 }

```

```

aLockedSamplesDropped OBJECT-TYPE
SYNTAX          CardinalNumber
MAX-ACCESS      read-write
STATUS          current
DESCRIPTION     "See IEC 62379-2 clause 5.3.4."
::= { aLockedEntry 4 }

```

```

aPortGroup OBJECT-GROUP
OBJECTS        { aPortDirection,
                  aPortFormat,
                  aPortTransport,
                  aPortName,
                  aes3ChannelData,
                  aes3UserData,
                  aes3ValidityError,
                  aPhantomEnabled,
                  aPhantomLevel,
                  aLockedTime,
                  aLockedSamplesInserted,
                  aLockedSamplesDropped }
STATUS          current
DESCRIPTION     "The group of objects used to control an audio port."
::= { audioPort 99 }

```

-- 5.4 Other audio block and associated managed object type definitions

-- 5.4.1 Audio mixer blocks

```

aMixerBlockTable OBJECT-TYPE
SYNTAX          SEQUENCE OF AMixerBlockEntry
MAX-ACCESS      not-accessible
STATUS          current
DESCRIPTION     "See IEC 62379-2 clause 5.4.1."
::= { audioMixer 1 }

```

```

aMixerBlockEntry OBJECT-TYPE
SYNTAX          AMixerBlockEntry
MAX-ACCESS      not-accessible
STATUS          current
DESCRIPTION     "See IEC 62379-2 clause 5.4.1."
INDEX          { aMixerBlockId }
::= { aMixerBlockTable 1 }

```

```

aMixerBlockId OBJECT-TYPE
SYNTAX          BlockId
MAX-ACCESS      not-accessible
STATUS          current
DESCRIPTION     "See IEC 62379-2 clause 5.4.1."
::= { aMixerBlockEntry 1 }

```

```

aMixerFadeDuration OBJECT-TYPE
SYNTAX          CardinalNumber
MAX-ACCESS      read-write
STATUS          current
DESCRIPTION     "See IEC 62379-2 clause 5.4.1."
::= { aMixerBlockEntry 2 }

```

```

aMixerFadeNow OBJECT-TYPE
SYNTAX          TruthValue
MAX-ACCESS      read-write
STATUS          current
DESCRIPTION     "See IEC 62379-2 clause 5.4.1."
::= { aMixerBlockEntry 3 }

```

```

aMixerInputTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF AMixerInputEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION  "See IEC 62379-2 clause 5.4.1."
    ::= { audioMixer 2 }

aMixerInputEntry OBJECT-TYPE
    SYNTAX      AMixerInputEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION  "See IEC 62379-2 clause 5.4.1."
    INDEX       { aMixerInputBlockId, aMixerInputNumber }
    ::= { aMixerInputTable 1 }

aMixerInputBlockId OBJECT-TYPE
    SYNTAX      BlockId
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION  "See IEC 62379-2 clause 5.4.1."
    ::= { aMixerInputEntry 1 }

aMixerInputNumber OBJECT-TYPE
    SYNTAX      IndexNumber
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION  "See IEC 62379-2 clause 5.4.1."
    ::= { aMixerInputEntry 2 }

aMixerInputLevel OBJECT-TYPE
    SYNTAX      AudioLevel
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION  "See IEC 62379-2 clause 5.4.1."
    ::= { aMixerInputEntry 3 }

aMixerInputFadeToLevel OBJECT-TYPE
    SYNTAX      AudioLevel
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION  "See IEC 62379-2 clause 5.4.1."
    ::= { aMixerInputEntry 4 }

aMixerInputDelay OBJECT-TYPE
    SYNTAX      CardinalNumber
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION  "See IEC 62379-2 clause 5.4.1."
    ::= { aMixerInputEntry 5 }

audioMixerGroup OBJECT-GROUP
    OBJECTS      { aMixerFadeDuration,
                   aMixerFadeNow,
                   aMixerInputLevel,
                   aMixerInputFadeToLevel,
                   aMixerInputDelay }
    STATUS      current
    DESCRIPTION  "The group of objects used to control an audio mixer block."
    ::= { audioMixer 99 }

```

-- 5.4.2 Audio crosspoint blocks

```

aCrosspointBlockTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF ACrosspointBlockEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION  "See IEC 62379-2 clause 5.4.2."
    ::= { audioCrosspoint 1 }

```

```

aCrosspointBlockEntry OBJECT-TYPE
    SYNTAX      ACrosspointBlockEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION  "See IEC 62379-2 clause 5.4.2."
    INDEX       { aCrosspointBlockId }
    ::= { aCrosspointBlockTable 1 }

aCrosspointBlockId OBJECT-TYPE
    SYNTAX      BlockId
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION  "See IEC 62379-2 clause 5.4.2."
    ::= { aCrosspointBlockEntry 1 }

aCrosspointConfigure OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION  "See IEC 62379-2 clause 5.4.2."
    ::= { aCrosspointBlockEntry 2 }

aCrosspointCopy OBJECT-TYPE
    SYNTAX      BlockId
    MAX-ACCESS  write-only
    STATUS      current
    DESCRIPTION  "See IEC 62379-2 clause 5.4.2."
    ::= { aCrosspointBlockEntry 3 }

aCrosspointPathTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF ACrosspointPathEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION  "See IEC 62379-2 clause 5.4.2."
    ::= { audioCrosspoint 2 }

aCrosspointPathEntry OBJECT-TYPE
    SYNTAX      ACrosspointPathEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION  "See IEC 62379-2 clause 5.4.2."
    INDEX       { aCrosspointPathBlockId,
                  aCrosspointPathSrc,
                  aCrosspointPathDst }
    ::= { aCrosspointPathTable 1 }

aCrosspointPathBlockId OBJECT-TYPE
    SYNTAX      BlockId
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION  "See IEC 62379-2 clause 5.4.2."
    ::= { aCrosspointPathEntry 1 }

aCrosspointPathSrc OBJECT-TYPE
    SYNTAX      AudioChannel
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION  "See IEC 62379-2 clause 5.4.2."
    ::= { aCrosspointPathEntry 2 }

aCrosspointPathDst OBJECT-TYPE
    SYNTAX      AudioChannel
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION  "See IEC 62379-2 clause 5.4.2."
    ::= { aCrosspointPathEntry 3 }

aCrosspointPathGain OBJECT-TYPE
    SYNTAX      AudioLevel

```

```

MAX-ACCESS      read-write
STATUS          current
DESCRIPTION     "See IEC 62379-2 clause 5.4.2."
::= { aCrosspointPathEntry 4 }

aCrosspointPathNewGain OBJECT-TYPE
    SYNTAX      AudioLevel
    MAX-ACCESS   read-write
    STATUS      current
    DESCRIPTION  "See IEC 62379-2 clause 5.4.2."
    ::= { aCrosspointPathEntry 5 }

aCrosspointPathPhase OBJECT-TYPE
    SYNTAX      AudioPhase
    MAX-ACCESS   read-write
    STATUS      current
    DESCRIPTION  "See IEC 62379-2 clause 5.4.2."
    ::= { aCrosspointPathEntry 6 }

aCrosspointPathNewPhase OBJECT-TYPE
    SYNTAX      AudioPhase
    MAX-ACCESS   read-write
    STATUS      current
    DESCRIPTION  "See IEC 62379-2 clause 5.4.2."
    ::= { aCrosspointPathEntry 7 }

audioCrosspointGroup OBJECT-GROUP
    OBJECTS      { aCrosspointConfigure,
                    aCrosspointCopy,
                    aCrosspointPathGain,
                    aCrosspointPathNewGain,
                    aCrosspointPathPhase,
                    aCrosspointPathNewPhase }
    STATUS      current
    DESCRIPTION  "The group of objects used to control an audio crosspoint
                    block."
    ::= { audioCrosspoint 99 }

-- 5.4.3 Audio clip player blocks

aClipPlayerBlockTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF AClipPlayerBlockEntry
    MAX-ACCESS   not-accessible
    STATUS      current
    DESCRIPTION  "See IEC 62379-2 clause 5.4.3."
    ::= { audioClipPlayer 1 }

aClipPlayerBlockEntry OBJECT-TYPE
    SYNTAX      AClipPlayerBlockEntry
    MAX-ACCESS   not-accessible
    STATUS      current
    DESCRIPTION  "See IEC 62379-2 clause 5.4.3."
    INDEX       { acpBlockId }
    ::= { aClipPlayerBlockTable 1 }

acpBlockId OBJECT-TYPE
    SYNTAX      BlockId
    MAX-ACCESS   not-accessible
    STATUS      current
    DESCRIPTION  "See IEC 62379-2 clause 5.4.3."
    ::= { aClipPlayerBlockEntry 1 }

acpSelection OBJECT-TYPE
    SYNTAX      AudioClipSelector
    MAX-ACCESS   read-write
    STATUS      current
    DESCRIPTION  "See IEC 62379-2 clause 5.4.3."
    ::= { aClipPlayerBlockEntry 2 }

```

```
acpMainFaderLevelNow OBJECT-TYPE
    SYNTAX      AudioLevel
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION  "See IEC 62379-2 clause 5.4.3."
    ::= { aClipPlayerBlockEntry 3 }

acpMainFaderLevelStart OBJECT-TYPE
    SYNTAX      AudioLevel
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION  "See IEC 62379-2 clause 5.4.3."
    ::= { aClipPlayerBlockEntry 4 }

acpMainFaderLevelStop OBJECT-TYPE
    SYNTAX      AudioLevel
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION  "See IEC 62379-2 clause 5.4.3."
    ::= { aClipPlayerBlockEntry 5 }

acpMainFaderDuration OBJECT-TYPE
    SYNTAX      CardinalNumber
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION  "See IEC 62379-2 clause 5.4.3."
    ::= { aClipPlayerBlockEntry 6 }

acpClipFaderLevelNow OBJECT-TYPE
    SYNTAX      AudioLevel
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION  "See IEC 62379-2 clause 5.4.3."
    ::= { aClipPlayerBlockEntry 7 }

acpClipFaderLevelStart OBJECT-TYPE
    SYNTAX      AudioLevel
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION  "See IEC 62379-2 clause 5.4.3."
    ::= { aClipPlayerBlockEntry 8 }

acpClipFaderLevelStop OBJECT-TYPE
    SYNTAX      AudioLevel
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION  "See IEC 62379-2 clause 5.4.3."
    ::= { aClipPlayerBlockEntry 9 }

acpClipFaderDuration OBJECT-TYPE
    SYNTAX      CardinalNumber
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION  "See IEC 62379-2 clause 5.4.3."
    ::= { aClipPlayerBlockEntry 10 }

acpStartDelay OBJECT-TYPE
    SYNTAX      IntegerNumber
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION  "See IEC 62379-2 clause 5.4.3."
    ::= { aClipPlayerBlockEntry 11 }

acpActivation OBJECT-TYPE
    SYNTAX      AudioClipActivation
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION  "See IEC 62379-2 clause 5.4.3."
    ::= { aClipPlayerBlockEntry 12 }
```

```

acpTrackControl OBJECT-TYPE
    SYNTAX      AudioClipTrackControl
    MAX-ACCESS   write-only
    STATUS       current
    DESCRIPTION  "See IEC 62379-2 clause 5.4.3."
    ::= { aClipPlayerBlockEntry 13 }

acpTrackNumber OBJECT-TYPE
    SYNTAX      CardinalNumber
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION  "See IEC 62379-2 clause 5.4.3."
    ::= { aClipPlayerBlockEntry 14 }

acpRepeat OBJECT-TYPE
    SYNTAX      AudioClipRepeat
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION  "See IEC 62379-2 clause 5.4.3."
    ::= { aClipPlayerBlockEntry 15 }

aClipSequenceTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF AClipSequenceEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION  "See IEC 62379-2 clause 5.4.3."
    ::= { audioClipPlayer 2 }

aClipSequenceEntry OBJECT-TYPE
    SYNTAX      AClipSequenceEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION  "See IEC 62379-2 clause 5.4.3."
    INDEX       { aClipSequenceBlockId, aClipSequenceId }
    ::= { aClipSequenceTable 1 }

aClipSequenceBlockId OBJECT-TYPE
    SYNTAX      BlockId
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION  "See IEC 62379-2 clause 5.4.3."
    ::= { aClipSequenceEntry 1 }

aClipSequenceId OBJECT-TYPE
    SYNTAX      AudioClipSelector
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION  "See IEC 62379-2 clause 5.4.3."
    ::= { aClipSequenceEntry 2 }

aClipSequenceName OBJECT-TYPE
    SYNTAX      Utf8String
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION  "See IEC 62379-2 clause 5.4.3."
    ::= { aClipSequenceEntry 3 }

audioClipPlayerGroup OBJECT-GROUP
    OBJECTS      { acpSelection,
                    acpMainFaderLevelNow,
                    acpMainFaderLevelStart,
                    acpMainFaderLevelStop,
                    acpMainFaderDuration,
                    acpClipFaderLevelNow,
                    acpClipFaderLevelStart,
                    acpClipFaderLevelStop,
                    acpClipFaderDuration,
                    acpStartDelay,
                    acpActivation,
                    acpTrackControl,

```

```

        acpTrackNumber,
        acpRepeat,
        aClipSequenceName }
STATUS      current
DESCRIPTION  "The group of objects used to control an audio clip player
              block."
::= { audioClipPlayer 99 }

-- 5.4.4 Audio limiter blocks

aLimiterBlockTable OBJECT-TYPE
SYNTAX      SEQUENCE OF ALimiterBlockEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION "See IEC 62379-2 clause 5.4.4."
::= { audioLimiter 1 }

aLimiterBlockEntry OBJECT-TYPE
SYNTAX      ALimiterBlockEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION "See IEC 62379-2 clause 5.4.4."
INDEX       { aLimiterBlockId }
::= { aLimiterBlockTable 1 }

aLimiterBlockId OBJECT-TYPE
SYNTAX      BlockId
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION "See IEC 62379-2 clause 5.4.4."
::= { aLimiterBlockEntry 1 }

aLimiterThreshold OBJECT-TYPE
SYNTAX      AudioLevel
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION "See IEC 62379-2 clause 5.4.4."
::= { aLimiterBlockEntry 2 }

aLimiterAttackTime OBJECT-TYPE
SYNTAX      CardinalNumber
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION "See IEC 62379-2 clause 5.4.4."
::= { aLimiterBlockEntry 3 }

aLimiterGainMakeup OBJECT-TYPE
SYNTAX      AudioLevel
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION "See IEC 62379-2 clause 5.4.4."
::= { aLimiterBlockEntry 4 }

aLimiterRecoveryTime OBJECT-TYPE
SYNTAX      CardinalNumber
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION "See IEC 62379-2 clause 5.4.4."
::= { aLimiterBlockEntry 5 }

aLimiterRecoveryMode OBJECT-TYPE
SYNTAX      AudioRecoveryMode
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION "See IEC 62379-2 clause 5.4.4."
::= { aLimiterBlockEntry 6 }

audioLimiterGroup OBJECT-GROUP
OBJECTS       { aLimiterThreshold,

```

```

        aLimiterAttackTime,
        aLimiterGainMakeup,
        aLimiterRecoveryTime,
        aLimiterRecoveryMode }
STATUS      current
DESCRIPTION "The group of objects used to control an audio limiter block."
::= { audioLimiter 99 }

```

-- 5.4.5 Audio converter blocks

```

aConverterBlockTable OBJECT-TYPE
SYNTAX      SEQUENCE OF AConverterBlockEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION "See IEC 62379-2 clause 5.4.5."
::= { audioConverter 1 }

```

```

aConverterBlockEntry OBJECT-TYPE
SYNTAX      AConverterBlockEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION "See IEC 62379-2 clause 5.4.5."
INDEX       { aConverterBlockId }
::= { aConverterBlockTable 1 }

```

```

aConverterBlockId OBJECT-TYPE
SYNTAX      BlockId
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION "See IEC 62379-2 clause 5.4.5."
::= { aConverterBlockEntry 1 }

```

```

aConverterQuality OBJECT-TYPE
SYNTAX      AudioQuality
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION "See IEC 62379-2 clause 5.4.5."
::= { aConverterBlockEntry 2 }

```

```

aConverterEnabled OBJECT-TYPE
SYNTAX      TruthValue
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION "See IEC 62379-2 clause 5.4.5."
::= { aConverterBlockEntry 3 }

```

```

aConverterDithering OBJECT-TYPE
SYNTAX      TruthValue
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION "See IEC 62379-2 clause 5.4.5."
::= { aConverterBlockEntry 4 }

```

```

aConverterOutputFormat OBJECT-TYPE
SYNTAX      MediaFormat
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION "See IEC 62379-2 clause 5.4.5."
::= { aConverterBlockEntry 5 }

```

```

aConverterError OBJECT-TYPE
SYNTAX      TruthValue
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION "See IEC 62379-2 clause 5.4.5."
::= { aConverterBlockEntry 6 }

```

```

audioConverterGroup OBJECT-GROUP
OBJECTS        { aConverterQuality,

```

```

        aConverterEnabled,
        aConverterDithering,
        aConverterOutputFormat,
        aConverterError}
STATUS      current
DESCRIPTION  "The group of objects used to control an audio converter block."
::= { audioConverter 99 }

```

-- 5.4.6 Audio level alarm block

```

aLevelAlarmBlockTable OBJECT-TYPE
SYNTAX      SEQUENCE OF ALevelAlarmBlockEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION  "See IEC 62379-2 clause 5.4.6."
::= { audioLevelAlarm 1 }

```

```

aLevelAlarmBlockEntry OBJECT-TYPE
SYNTAX      ALevelAlarmBlockEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION  "See IEC 62379-2 clause 5.4.6."
INDEX       { alaBlockId }
::= { aLevelAlarmBlockTable 1 }

```

```

alaBlockId OBJECT-TYPE
SYNTAX      BlockId
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION  "See IEC 62379-2 clause 5.4.6."
::= { aLevelAlarmBlockEntry 1 }

```

```

alaType OBJECT-TYPE
SYNTAX      AudioLevelAlarmType
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION  "See IEC 62379-2 clause 5.4.6."
::= { aLevelAlarmBlockEntry 2 }

```

```

alaThreshold OBJECT-TYPE
SYNTAX      AudioLevel
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION  "See IEC 62379-2 clause 5.4.6."
::= { aLevelAlarmBlockEntry 3 }

```

```

alaWarningTime OBJECT-TYPE
SYNTAX      CardinalNumber
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION  "See IEC 62379-2 clause 5.4.6."
::= { aLevelAlarmBlockEntry 4 }

```

```

alaFailureTime OBJECT-TYPE
SYNTAX      CardinalNumber
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION  "See IEC 62379-2 clause 5.4.6."
::= { aLevelAlarmBlockEntry 5 }

```

```

alaCounter OBJECT-TYPE
SYNTAX      CardinalNumber
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION  "See IEC 62379-2 clause 5.4.6."
::= { aLevelAlarmBlockEntry 6 }

```

```

alaEnabled OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION  "See IEC 62379-2 clause 5.4.6."
    ::= { aLevelAlarmBlockEntry 7 }

alaStatus OBJECT-TYPE
    SYNTAX      AudioAlarmStatus
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION  "See IEC 62379-2 clause 5.4.6."
    ::= { aLevelAlarmBlockEntry 8 }

audioLevelAlarmGroup OBJECT-GROUP
    OBJECTS      { alaType,
                    alaThreshold,
                    alaWarningTime,
                    alaFailureTime,
                    alaCounter,
                    alaEnabled,
                    alaStatus }
    STATUS        current
    DESCRIPTION   "The group of objects used to control an audio level alarm
    block."
    ::= { audioLevelAlarm 99 }

-- Compliance statements

audioMIBComplianceV1 MODULE-COMPLIANCE
    STATUS        current
    DESCRIPTION   "The compliance statement for entities that conform to
    IEC 62379-2 (2008)."
```

MODULE -- this module

```

    MANDATORY-GROUPS {}

    GROUP         aPortGroup
    DESCRIPTION   "Mandatory for equipment that contains one or more analogue
    ports."
```

GROUP audioMixerGroup

```

    DESCRIPTION   "Mandatory for equipment that contains one or more audio
    mixer blocks."
```

GROUP audioCrosspointGroup

```

    DESCRIPTION   "Mandatory for equipment that contains one or more audio
    crosspoint blocks."
```

GROUP audioClipPlayerGroup

```

    DESCRIPTION   "Mandatory for equipment that contains one or more audio
    clip player blocks."
```

GROUP audioLimiterGroup

```

    DESCRIPTION   "Mandatory for equipment that contains one or more audio
    limiter blocks."
```

GROUP audioConverterGroup

```

    DESCRIPTION   "Mandatory for equipment that contains one or more audio
    converter blocks."
```

GROUP audioLevelAlarmGroup

```

    DESCRIPTION   "Mandatory for equipment that contains one or more audio
    alarm blocks."
```

MODULE IEC62379-2-STATUS-MIB

GROUP audioFormatsMapGroup

DESCRIPTION "Mandatory for equipment that reports audio format
information in status pages."

::= {audioMIBCompliance 1 }

END

IECNORM.COM : Click to view the full PDF of IEC 62379-2:2008

Annex C (informative)

Machine-readable status page group definitions

This annex provides a machine-readable version of the status page group definitions which is intended to be interpretable by standard MIB browsing software tools. If there is any inconsistency between this annex and Clause 6, Clause 6 takes precedence.

The format used to describe the status page group identifiers conforms to IETF STD 58 (SMIv2).

```
IEC62379-2-STATUS-GROUPS DEFINITIONS ::= BEGIN

IMPORTS

    MODULE-IDENTITY, OBJECT-IDENTITY
        FROM SNMPv2-SMI

    iec62379
        FROM IEC62379-1-MIB;

audioStatusGroup MODULE-IDENTITY
    LAST-UPDATED "200804301500Z"
    ORGANIZATION "IEC PT62379"
    CONTACT-INFO "Not specified"
    DESCRIPTION "The status page group identifiers defined in clause 6.5 of
        IEC 62379-2."
    REVISION "200804301500Z"
    DESCRIPTION "Edition 1.0 final draft."
    ::= { audio 3 }

audio OBJECT IDENTIFIER ::= { iec62379 2 }

-- 6.5 Page groups

audioPorts OBJECT-IDENTITY
    STATUS current
    DESCRIPTION "See clause 6.5.1 of IEC 62379-2."
    ::= { audioStatusGroup 1 }

standardAudioBlocks OBJECT-IDENTITY
    STATUS current
    DESCRIPTION "See clause 6.5.2 of IEC 62379-2."
    ::= { audioStatusGroup 2 }

audioAlarms OBJECT-IDENTITY
    STATUS current
    DESCRIPTION "See clause 6.5.3 of IEC 62379-2."
    ::= { audioStatusGroup 3 }

END
```

Annex D (informative)

Machine-readable status page MIB definitions

This annex provides a machine-readable version of the status page MIB definitions which is intended to be interpretable by standard MIB browsing software tools. If there is any inconsistency between this annex and Clause 6, Clause 6 takes precedence.

The format used to describe the status page group identifiers conforms to IETF STD 58 (SMIv2).

```

IEC62379-2-STATUS-MIB DEFINITIONS ::= BEGIN

IMPORTS

    MODULE-IDENTITY, OBJECT-IDENTITY
        FROM SNMPv2-SMI

    iec62379, IndexNumber, MediaFormat
        FROM IEC62379-1-MIB;

audioStatusMIB MODULE-IDENTITY
    LAST-UPDATED "200804301500Z"
    ORGANIZATION "IEC PT62379"
    CONTACT-INFO "Not specified"
    DESCRIPTION "The MIB module for managing audio status pages in
        IEC 62379-2."
    REVISION "200804301500Z"
    DESCRIPTION "Edition 1.0 final draft."
    ::= { audio 4 }

-- Node definitions

audio OBJECT IDENTIFIER ::= { iec62379 2 }

audioStatusMIBCompliance OBJECT IDENTIFIER ::= { audioStatusMIB 1 }

audioFormatsMap OBJECT IDENTIFIER ::= { audioStatusMIB 2 }

-- 6.2 Application wide type definitions

-- 6.2.1 Textual conventions

AudioConverterStatus ::= TEXTUAL-CONVENTION
    STATUS current
    DESCRIPTION "A set of bits representing the status of an audio converter:
        bit 0 (lsb) = converter enabled (1=enabled, 0=not enabled)
        bit 1      = dithering applied (1=dithering, 0=no dithering)
        bit 2      = conversion ok (1=input converted, 0=converter error)
        bit 3      = reserved
        bit 4      = reserved
        bit 5      = reserved
        bit 6      = reserved
        bit 7 (msb) = reserved"
    SYNTAX OCTET STRING (SIZE(1))

-- 6.2.2 Sequences

AudioFormatsMapEntry ::= SEQUENCE {
    afmNumber    IndexNumber,
    afmFormat    MediaFormat

```

```

}

-- 6.3 Audio formats mapping

audioFormatsMapTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF AudioFormatsMapEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION   "See clause 6.3 of IEC 62379-2."
    ::= { audioFormatsMap 1 }

audioFormatsMapEntry OBJECT-TYPE
    SYNTAX      AudioFormatsMapEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION   "See clause 6.3 of IEC 62379-2."
    INDEX       { afmNumber }
    ::= { audioFormatsMapTable 1 }

afmNumber OBJECT-TYPE
    SYNTAX      IndexNumber
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION   "See clause 6.3 of IEC 62379-2."
    ::= { audioFormatsMapEntry 1 }

afmFormat OBJECT-TYPE
    SYNTAX      MediaFormat
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION   "See clause 6.3 of IEC 62379-2."
    ::= { audioFormatsMapEntry 2 }

audioFormatsMapGroup OBJECT-GROUP
    OBJECTS      { afmFormat }
    STATUS       current
    DESCRIPTION   "The group of objects used to represent audio formats in
                  a status page."
    ::= { audioFormatsMap 99 }

-- Compliance statements

audioStatusMIBComplianceV1 MODULE-COMPLIANCE
    STATUS       current
    DESCRIPTION   "The compliance statement for entities that conform to
                  IEC 62379-2 (2008)."
```

MODULE ~~is~~ this module

```

    GROUP       audioFormatsMapGroup
    DESCRIPTION   "Mandatory for equipment that reports audio format
                  information in status pages."
    ::= { audioStatusMIBCompliance 1 }

END
```

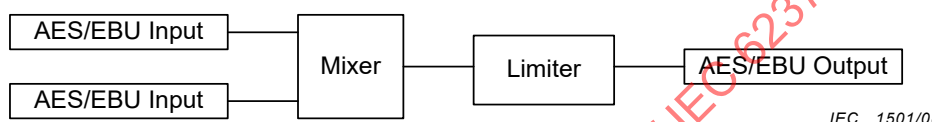
Annex E (informative)

Worked examples

This annex provides two worked examples to show how Parts 1 and 2 of this standard may be used to model audio devices. If there is any inconsistency between this annex and the main clauses of Parts 1 and 2, those clauses take precedence.

E.1 Example 1

To show the relationships between the various tables defined in Parts 1 and 2 of IEC 62379 consider a simple audio device that mixes two inputs and provides a limiter, as represented by the following diagram:



Assume the unit is capable of carrying two channel stereo 24 bit audio at either 44,1k or 48k but that it is currently operating at 48k.

E.1.1 Block table

The first thing that is required is a block table showing all the functional blocks that make up the unit. The mapping of a block to a blockId is arbitrary; what matters is that a particular block is always associated with the same blockId. The connector table will show the structure of the unit by showing the connections between the blocks.

The table is located at *iso.standard.iec67329.general.generalMIB.block.blockTable* (1.0.62379.1.1.2.1)

As there are five blocks in the example, this table will have five rows:

.blockTable.blockEntry (.1.1)	
blockId (1)	blockType (2)
①	1.0.62379.2.1.1 (Audio port)
②	1.0.62379.2.1.1 (Audio port)
③	1.0.62379.2.1.2 (Mixer block)
④	1.0.62379.2.1.5 (Limiter block)
⑤	1.0.62379.2.1.1 (Audio port)

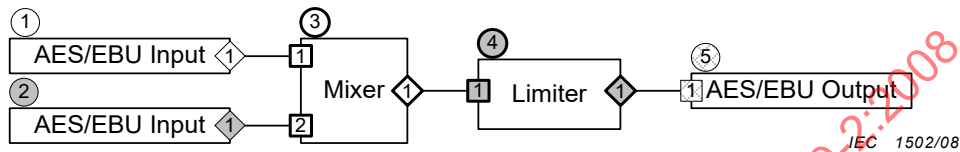
E.1.2 Connector table

To show the structure of the unit, a connector table is required to express all the links that exist between the blocks defined in the block table.

This table is located at *iso.standard.iec67329.general.generalMIB.block.connectorTable* (1.0.62379.1.1.2.2)

.connectorTable.connectorEntry (.2.1)			
connRxBlockId (1)	connRxBlockInput (2)	connTxBlockId (3)	connTxBlockOutput (4)
③	1	①	1
③	2	②	1
④	1	③	1
⑤	1	④	1

Below is the diagram of the unit annotated with each block's blockId (in circles), input numbers (squares) and output numbers (diamonds):



E.1.3 Mode table

The mode table shows all the formats that are valid for each internal output.

This table is located at *iso.standard.iec67329.general.generalMIB.block.modeTable* (1.0.62379.1.1.2.3)

In this example, there are four internal outputs, each capable of running at 44,1k or 48k. There are therefore eight entries in this table:

.modeTable.modeEntry (.3.1)			
mBlockId (1)	mBlockOutput (2)	mMediaFormat (3)	mEnabled (4)
①	1	1.0.62379.2.2.1.3.2.2.24.44100	true (1)
①	1	1.0.62379.2.2.1.3.2.2.24.48000	true (1)
②	1	1.0.62379.2.2.1.3.2.2.24.44100	true (1)
②	1	1.0.62379.2.2.1.3.2.2.24.48000	true (1)
③	1	1.0.62379.2.2.1.3.2.2.24.44100	true (1)
③	1	1.0.62379.2.2.1.3.2.2.24.48000	true (1)
④	1	1.0.62379.2.2.1.3.2.2.24.44100	true (1)
④	1	1.0.62379.2.2.1.3.2.2.24.48000	true (1)

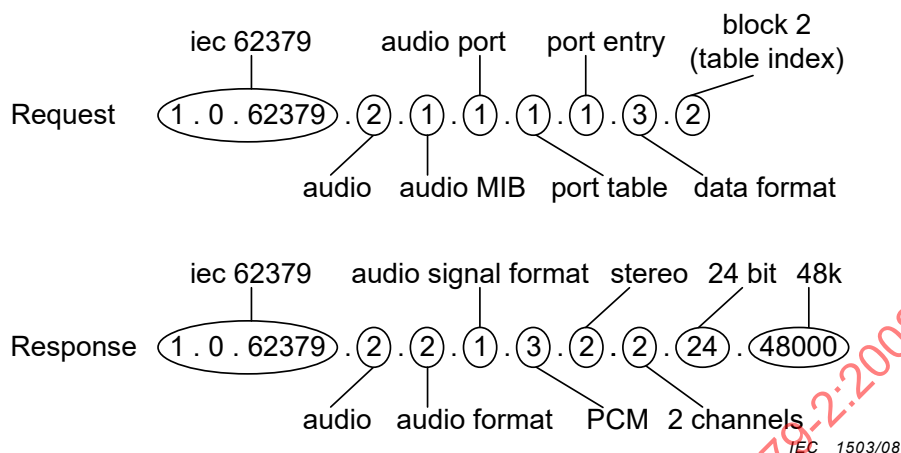
E.1.4 Audio ports

Audio ports are rooted at *iso.standard.iec62379.audio.audioMIB.audioPort* (1.0.62379.2.1.1)

The block consists of a table, *aPortTable*, with entries for all ports in the unit, located at 1.0.62379.2.1.1.1. In this example, there are three entries in this table, one for each port. *aPortDataFormat* shows that all ports are carrying two channel stereo PCM audio, at 24 bits, 48k; *aPortTransport* shows that this audio is carried in an AES3 transport:

.aPortTable.aPortEntry (.1.1)				
aPortBlockId (1)	aPortDirection (2)	aPortDataFormat (3)	aPortTransport (4)	aPortName (5)
①	input (1)	1.0.62379.2.2.1.3.2.2.24.48000	1.0.62379.2.2.2.2	
②	input (1)	1.0.62379.2.2.1.3.2.2.24.48000	1.0.62379.2.2.2.2	
⑤	output (2)	1.0.62379.2.2.1.3.2.2.24.48000	1.0.62379.2.2.2.2	

So, to find the data format of block 2, issue a `GET` on `1.0.62379.2.1.1.1.3.2` to get a `GET RESPONSE` value of `1.0.62379.2.2.1.3.2.2.24.48000`:



The port block also consists of other tables to represent functionality specific to a particular port. As the ports in this example represent AES/EBU audio, AES3 data may be obtained from *aes3DataTable*, located at `1.0.62379.2.1.1.2`. There is an entry here for each channel of each port that supports AES3 data: in this example, this table has six rows:

.aAes3DataTable.aAes3DataEntry (.2.1)				
aes3 BlockId (1)	aes3 ChannelId (2)	aes3 ChannelData (3)	aes3 UserData (4)	aes3 ValidityError (5)
①	1			
①	2			
②	1			
②	2			
⑤	1			
⑤	2			

E.1.5 Mixer block

The mixer block is rooted at *iso.standard.iec62379.audio.audioMIB.audioMixer* (`1.0.62379.2.1.2`)

This block consists of two tables, *aMixerBlockTable*, located at `1.0.62379.2.1.2.1` and *aMixerInputTable*, located at `1.0.62379.2.1.2.2`. There is one entry in *aMixerBlockTable* for each mixer block in a unit and an entry in *aMixerInputTable* for each input of each mixer.

As the example has one mixer block with two inputs, *aMixerBlockTable* has one entry (of type *AMixerBlockEntry*) and *aMixerInputTable* has two entries (of type *AMixerInputEntry*):

.aMixerBlockTable.aMixerBlockEntry (.1.1)		
aMixerBlockId (1)	aMixerFadeDuration (2)	aMixerFadeNow (3)
③		

.aMixerInputTable.aMixerInputEntry (.2.1)				
aMixerInput BlockId (1)	aMixerInput Number (2)	aMixerInput Level (3)	aMixerInput FadeToLevel (4)	aMixerInput Delay (5)
③	①			
③	②			

E.1.6 Limiter block

The limiter block is located at *iso.standard.iec62379.audio.audioMIB.audioLimiter* (1.0.62379.2.1.5).

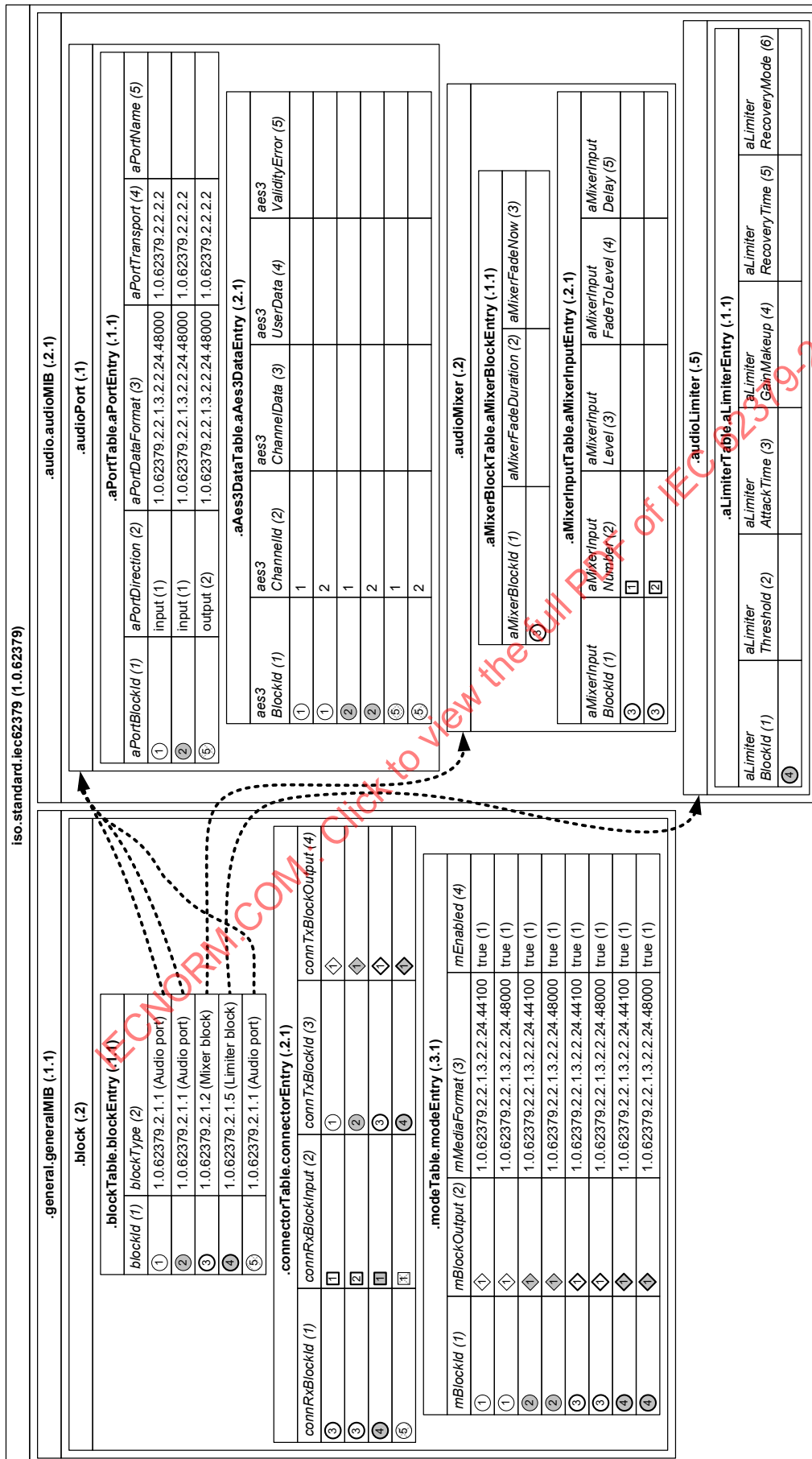
This block consists of a single table, *aLimiterBlockTable*, located at 1.0.62379.2.1.5.1. As there is only one limiter in the example, the limiter table has one entry (of type *ALimiterBlockEntry*):

.aLimiterTable.aLimiterEntry (.1.1)					
<i>aLimiterBlockId</i> (1)	<i>aLimiterThreshold</i> (2)	<i>aLimiterAttackTime</i> (3)	<i>aLimiterGainMakeup</i> (4)	<i>aLimiterRecoveryTime</i> (5)	<i>aLimiterRecoveryMode</i> (6)
④					

To set the limiter threshold to –60dB set 1.0.62379.2.1.5.1.1.2.4 to –6000.

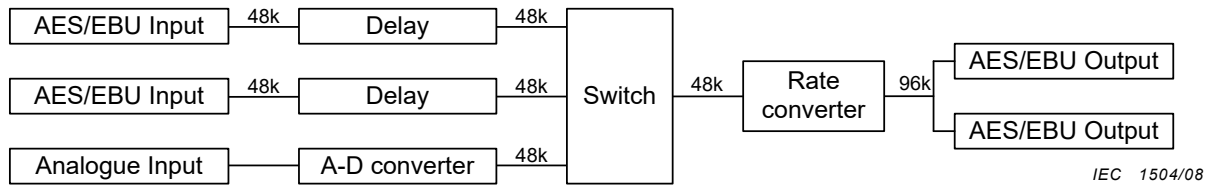
IECNORM.COM : Click to view the full PDF of IEC 62379-2:2008

E.1.7 Summary of tables



E.2 Example 2

Below, another device is shown.



This unit works at 24 bit, 48k until the rate converter, which can output 16 or 24 bit audio at 44,1k, 48k, 88,2k or 96k sampling. It is currently configured to output 24 bit, 96k. The entire unit works with two channel stereo.

E.2.1 Block and connector tables

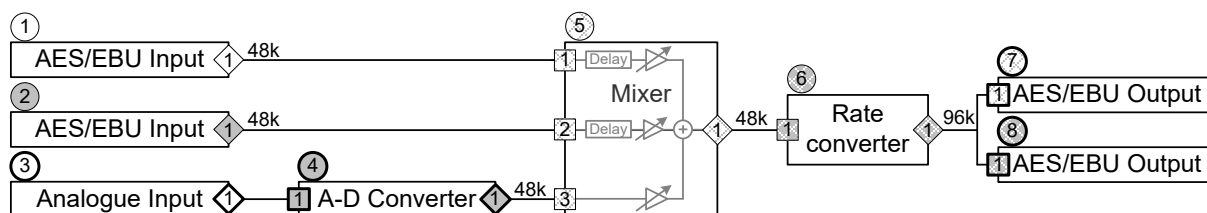
The two delays and the switch can be implemented using a mixer block with three inputs, which leads to the following block table:

.blockTable.blockEntry (.1.1)	
blockId (1)	blockType (2)
①	1.0.62379.2.1.1 (Audio port)
②	1.0.62379.2.1.1 (Audio port)
③	1.0.62379.2.1.1 (Audio port)
④	1.0.62379.2.1.6 (Converter block)
⑤	1.0.62379.2.1.2 (Mixer block)
⑥	1.0.62379.2.1.6 (Converter block)
⑦	1.0.62379.2.1.1 (Audio port)
⑧	1.0.62379.2.1.1 (Audio port)

The two AES/EBU outputs are derived from the same source, the rate converter, as shown in the connector table:

.connectorTable.connectorEntry (.2.1)			
connRxBlockId (1)	connRxBlockInput (2)	connTxBlockId (3)	connTxBlockOutput (4)
④	1	③	1
⑤	1	①	1
⑥	2	②	1
⑦	3	④	1
⑧	1	⑤	1
⑦	1	⑥	1
⑧	1	⑥	1

This diagram shows the unit annotated with block, input and output numbers. It also shows the delays and switch redrawn as a mixer block:



IEC 1505/08

E.2.2 Mode table

The mode table shows blocks 1, 2, 4 and 5 operating at 24 bit, 48k, along with analogue audio in block 3. There are eight entries for the rate converter: one for each combination of 16 or 24 bit and 44,1, 48, 88,2 or 96k. The *mEnabled* column is used by a converter block to specify which of the supported modes should be output: in this case, 24 bit, 96k is selected.

.modeTable.modeEntry (.3.1)			
<i>mBlockId</i> (1)	<i>mBlockOutput</i> (2)	<i>mMediaFormat</i> (3)	<i>mEnabled</i> (4)
①	①	1.0.62379.2.2.1.3.2.2.24.48000	true (1)
②	①	1.0.62379.2.2.1.3.2.2.24.48000	true (1)
③	④	1.0.62379.2.2.1.2.2.2	true (1)
④	④	1.0.62379.2.2.1.3.2.2.24.48000	true (1)
⑤	①	1.0.62379.2.2.1.3.2.2.24.48000	true (1)
⑥	①	1.0.62379.2.2.1.3.2.2.16.44100	false (2)
⑥	①	1.0.62379.2.2.1.3.2.2.16.48000	false (2)
⑥	①	1.0.62379.2.2.1.3.2.2.16.88200	false (2)
⑥	①	1.0.62379.2.2.1.3.2.2.16.96000	false (2)
⑥	①	1.0.62379.2.2.1.3.2.2.24.44100	false (2)
⑥	①	1.0.62379.2.2.1.3.2.2.24.48000	false (2)
⑥	①	1.0.62379.2.2.1.3.2.2.24.88200	false (2)
⑥	①	1.0.62379.2.2.1.3.2.2.24.96000	true (1)

E.2.3 Audio ports

There are four AES/EBU ports: two inputs working at 24 bit, 48k and two output ports, working at 24 bit, 96k. In addition, there is a two channel stereo analogue port too.

.aPortTable.aPortEntry (.1.1)				
<i>aPortBlockId</i> (1)	<i>aPortDirection</i> (2)	<i>aPortDataFormat</i> (3)	<i>aPortTransport</i> (4)	<i>aPortName</i> (5)
①	input (1)	1.0.62379.2.2.1.3.2.2.24.48000	1.0.62379.2.2.2.2	
②	input (1)	1.0.62379.2.2.1.3.2.2.24.48000	1.0.62379.2.2.2.2	
③	input (1)	1.0.62379.2.2.1.2.2.2	1.0.62379.2.2.2.1	
⑦	output (2)	1.0.62379.2.2.1.3.2.2.24.96000	1.0.62379.2.2.2.2	
⑧	output (2)	1.0.62379.2.2.1.3.2.2.24.96000	1.0.62379.2.2.2.2	

There is also an *aes3DataTable* with two entries for each of the four AES/EBU ports.

E.2.4 Mixer block

The mixer block as defined 'comes with' a delay on each input. In this example, there is no delay on the analogue input so the delay entry for input 3 will be fixed at zero.

.aMixerBlockTable.aMixerBlockEntry (.1.1)		
<i>aMixerBlockId</i> (1)	<i>aMixerFadeDuration</i> (2)	<i>aMixerFadeNow</i> (3)
⑤		

.aMixerInputTable.aMixerInputEntry (.2.1)				
<i>aMixerInputBlockId</i> (1)	<i>aMixerInputNumber</i> (2)	<i>aMixerInputLevel</i> (3)	<i>aMixerInputFadeToLevel</i> (4)	<i>aMixerInputDelay</i> (5)
⑤	①	As this block represents a switch, one of these will be at fullscale, the other two at minus infinity		Inputs 1 and 2 can accept a delay value; input 3 has a fixed delay of zero
⑤	②			
⑤	③			

E.2.5 Converter blocks

Both the analogue-digital converter and the sample rate converter can be implemented by a converter block, rooted at *iso.standard.iec62379.audio.audioMIB.converter* (1.0.62379.2.1.6).

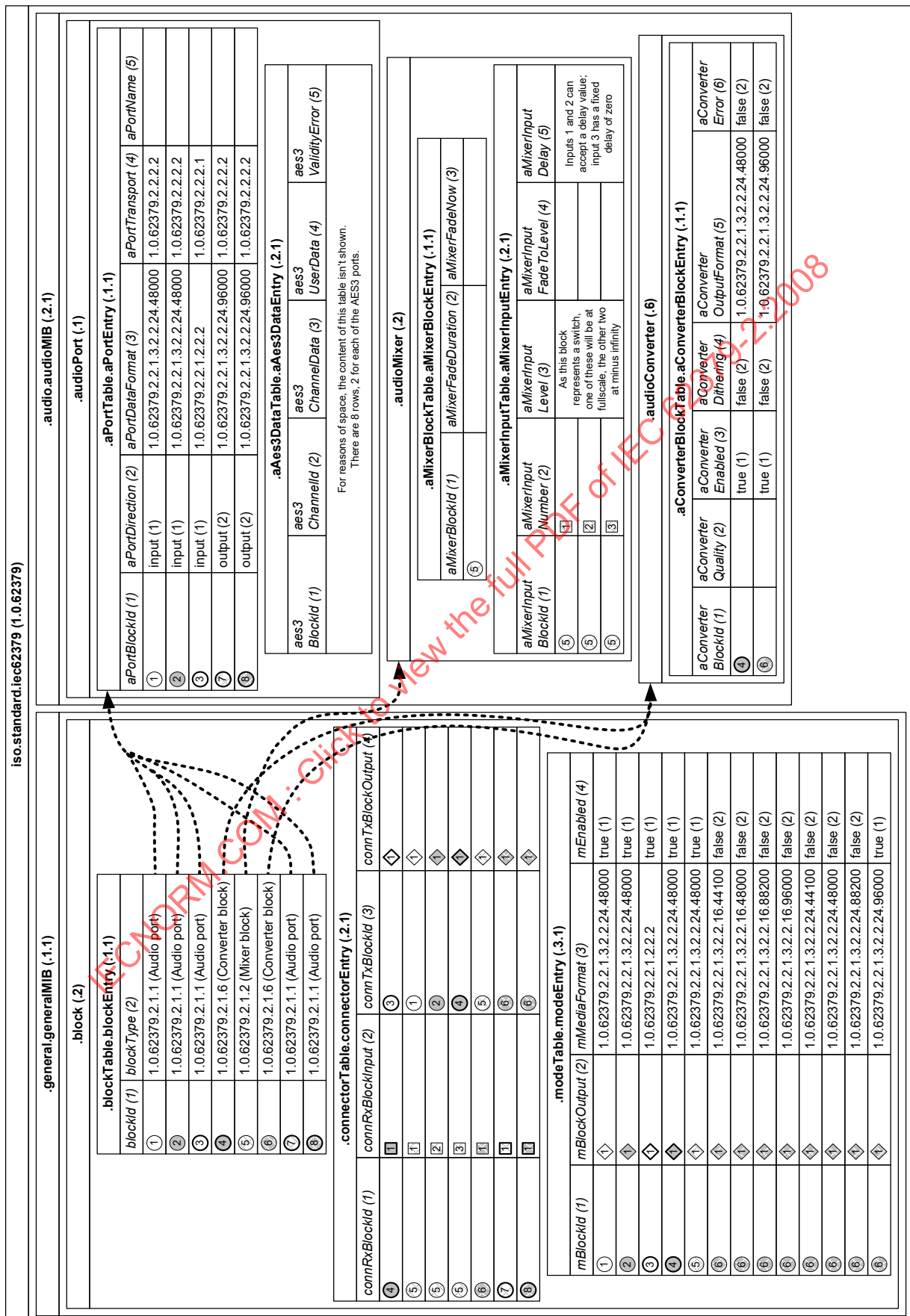
This block consists of a single table, *aConverterBlockTable*, located at 1.0.62379.2.1.6.1, which contains an entry for each converter.

As there are two converters, *aConverterBlockTable* contains two entries. Note that although the sample rate converter output goes to two AES3 ports they both come from the same output of the rate converter.

.aConverterBlockTable.aConverterBlockEntry (.1.1)					
<i>aConverterBlockId</i> (1)	<i>aConverterQuality</i> (2)	<i>aConverterEnabled</i> (3)	<i>aConverterDithering</i> (4)	<i>aConverterOutputFormat</i> (5)	<i>aConverterError</i> (6)
④		true (1)	false (2)	1.0.62379.2.2.1.3.2.2.24.48000	false (2)
⑥		true (1)	false (2)	1.0.62379.2.2.1.3.2.2.24.96000	false (2)

It is the mode table that controls what formats the converter block should output and not *aConverterBlockTable*: *aConverterOutputFormat* is a read only field which shows what format is being output, chosen by the unit's own implementation rules based on which entries in the mode table are enabled. In this example, the analogue to digital converter can only convert from analogue to 24 bit, 48k audio, so there is only one possibility in the mode table. In the case of the sample rate converter, there are eight possible output formats but only one is set to enabled (24 bit, 96k) and so the unit is forced to output that format.

E.2.6 Summary of tables



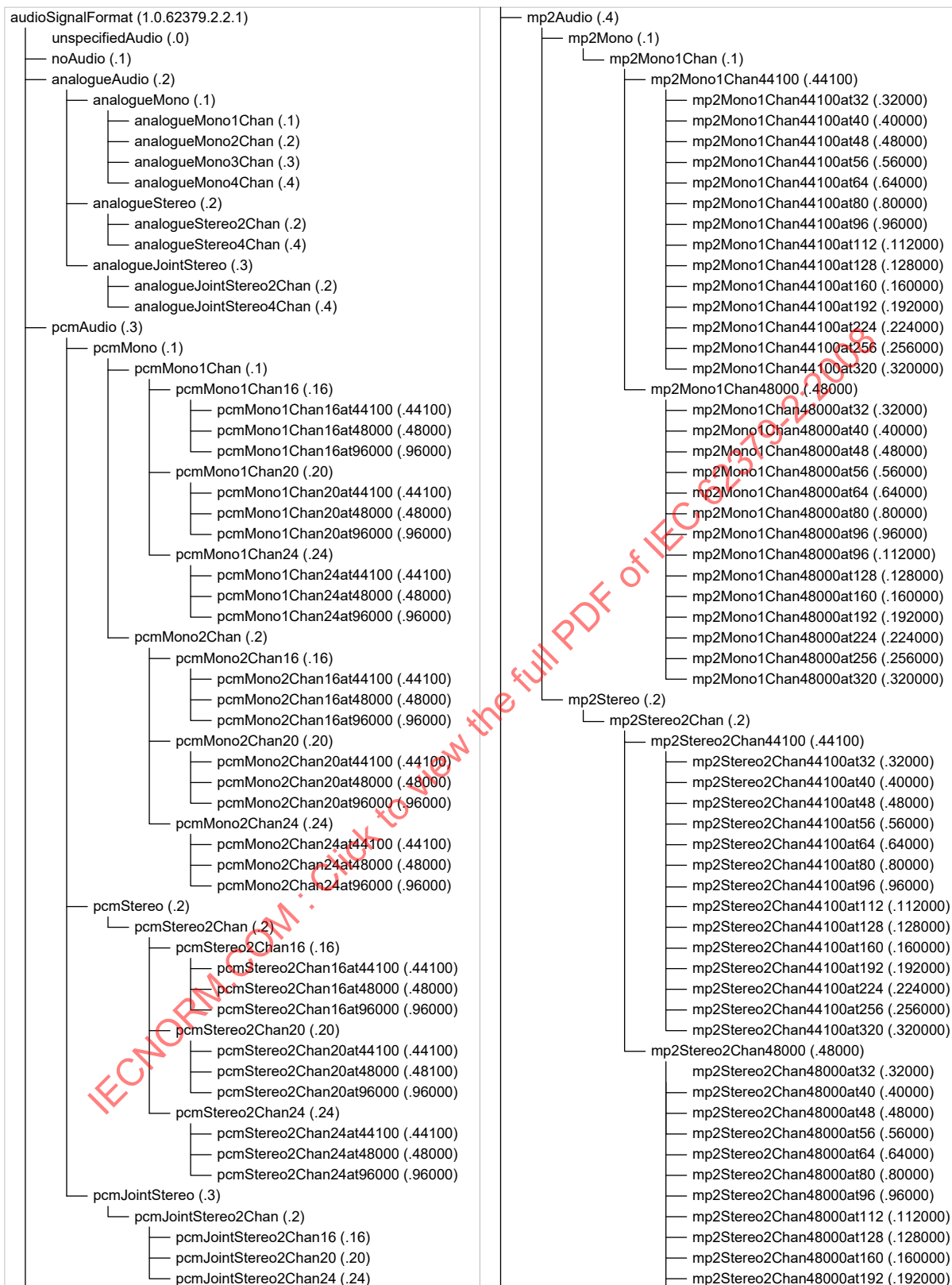
Annex F (informative)

Tree of example audio formats

This annex shows a graphical tree of the example audio formats listed in Annex A. If there is any inconsistency between this annex and Clause 4, Clause 4 takes precedence.

NOTE 1 This annex is not intended to cover every format permitted by the definitions in Clause 4.

IECNORM.COM : Click to view the full PDF of IEC 62379-2:2008



- mp2Stereo2Chan48000at224 (.224000) - mp2Stereo2Chan48000at256 (.256000) - mp2Stereo2Chan48000at320 (.320000) - mp2JointStereo (.3) - mp2JointStereo2Chan (.2) - mp2JointStereo2Chan44100 (.44100) - mp2JointStereo2Chan48000 (.48000) - mp3Audio (.5) - mp3Mono (.1) - mp3Mono1Chan (.1) - mp3Stereo (.2) - mp3Stereo2Chan (.2) - mp3JointStereo (.3) - mp3JointStereo2Chan (.2) - aacAudio (.6) - aacLC (.1) - aacMain (.2) - aacSRS (.3) - aacLTP (.4) - aacLD (.5) - g711Audio (.7) - g711ALaw (.1) - g711MuLaw (.2) - g722Audio (.8) - g722at48 (.48) - g722at56 (.56) - g722at64 (.64) - aptXAudio (.9) - enhAptXAudio (.10) - j41Audio (.11) - j41ALawA (.1) - j41ALawB (.2) - j41Nic (.3) - j57Audio (.12) - j57H11 (.1) - j57H12 (.2) - invalidAudio (.13)	- audioTransportFormat (1.0.62379.2.2.2) - unspecifiedTransport (.0) - analogueAudio (.1) - aes3 (.2) - aes10 (.3) - aes50 (.4) - audioMetadataFormat (1.0.62379.2.2.3) - unspecifiedMetadata (.0)
--	---

IECNORM.COM : Click to view the full PDF of IEC 62379-2:2008

IECNORM.COM : Click to view the full PDF of IEC 62379-2:2008

SOMMAIRE

AVANT-PROPOS	92
INTRODUCTION	94
1 Domaine d'application	95
2 Références normatives	95
3 Termes, définitions et abréviations	95
3.1 Abréviations	95
4 Définitions de format audio	96
4.1 Définitions de formats de signal audio	96
4.1.1 Paramètres audio	96
4.1.2 Formats de signal audio	98
4.2 Définitions de formats de transport audio	100
4.3 Définitions de formats de métadonnées audio	100
5 Définitions MIB pour blocs audio	101
5.1 Généralités	101
5.2 Définitions de types	101
5.2.1 Conventions textuelles	101
5.2.2 Séquences	103
5.3 Définitions d'accès audio et de types d'objets gérés associés	105
5.3.1 Fonctionnalité d'accès générique	105
5.3.2 Données auxiliaires AES3	106
5.3.3 Alimentation fantôme	107
5.3.4 Signal audio verrouillé sur une référence	108
5.4 Autres définitions de blocs audio et de types d'objets gérés associés	109
5.4.1 Blocs de mélangeur audio	109
5.4.2 Blocs de point de connexion audio	111
5.4.3 Blocs de lecteur de clip audio	116
5.4.4 Blocs de limiteur audio	119
5.4.5 Blocs de convertisseur audio	120
5.4.6 Blocs d'alarme de niveau audio	122
6 Diffusions d'état	124
6.1 Généralités	124
6.2 Définitions de types	124
6.2.1 Conventions textuelles	124
6.2.2 Séquences	124
6.3 Mappage de formats audio	124
6.3.1 audioFormatsMapTable	125
6.3.2 audioFormatsMapEntry	125
6.3.3 afmNumber	125
6.3.4 afmFormat	125
6.4 Formats de page	125
6.4.1 Page d'accès audio	125
6.4.2 Page de données auxiliaires AES3	125
6.4.3 Page de mélangeur audio	126
6.4.4 Page de point de connexion audio	126
6.4.5 Page de lecteur de clip audio	127
6.4.6 Page de limiteur audio	127

6.4.7	Page de convertisseur audio	128
6.4.8	Page d'alarme de niveau audio.....	128
6.5	Groupes de pages.....	129
6.5.1	audioPorts.....	129
6.5.2	standardAudioBlocks.....	129
6.5.3	audioAlarms.....	130
Annexe A (informative)	Définitions de formats audio lisibles par une machine.....	131
Annexe B (informative)	Définitions de blocs audio lisibles par une machine.....	147
Annexe C (informative)	Définitions de groupes de pages d'état lisibles par une machine.....	165
Annexe D (informative)	Définitions de MIB de pages d'état lisibles par une machine.....	166
Annexe E (informative)	Exemples pratiques.....	168
Annexe F (informative)	Arborescence des exemples de formats audio.....	181
Figure 1	– Blocs d'accès audio	105
Figure 2	– Bloc de mélangeur audio.....	109
Figure 3	– Bloc de point de connexion audio.....	112
Figure 4	– Bloc de lecteur de clip audio	116
Figure 5	– Bloc de limiteur audio.....	119
Figure 6	– Bloc de convertisseur audio	121
Figure 7	– Bloc d'alarme de niveau audio.....	122
Tableau 1	– Objets gérés pour accès audio	105
Tableau 2	– Objets gérés pour données auxiliaires AES3	106
Tableau 3	– Objets gérés pour alimentation fantôme	107
Tableau 4	– Objets gérés pour signal audio verrouillé.....	108
Tableau 5	– Objets gérés pour blocs de mélangeur audio.....	110
Tableau 6	– Objets gérés pour blocs de point de connexion audio.....	112
Tableau 7	– Objets gérés pour blocs de lecteur de clip audio.....	116
Tableau 8	– Objets gérés pour blocs de limiteur audio.....	120
Tableau 9	– Objets gérés pour blocs de convertisseur audio.....	121
Tableau 10	– Objets gérés pour blocs d'alarme de niveau audio.....	123
Tableau 11	– Objets gérés pour mappages de formats audio.....	125
Tableau 12	– Entrées d'état pour la page d'accès audio	125
Tableau 13	– Entrées d'état pour la page de données auxiliaires AES3	126
Tableau 14	– Entrées d'état pour la page de mélangeur audio	126
Tableau 15	– Entrées d'état pour la page de point de connexion audio	127
Tableau 16	– Entrées d'état pour la page de lecteur de clip audio.....	127
Tableau 17	– Entrées d'état pour la page de limiteur audio.....	128
Tableau 18	– Entrées d'état pour la page de convertisseur audio.....	128
Tableau 19	– Entrées d'état pour la page d'alarme de niveau audio.....	129

COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

INTERFACE DE COMMANDE COMMUNE POUR PRODUITS AUDIO ET VIDÉO NUMÉRIQUES CONNECTÉS EN RÉSEAU –

Partie 2: Audio

AVANT-PROPOS

- 1) La Commission Electrotechnique Internationale (IEC) est une organisation mondiale de normalisation composée de l'ensemble des comités électrotechniques nationaux (Comités nationaux de l'IEC). L'IEC a pour objet de favoriser la coopération internationale pour toutes les questions de normalisation dans les domaines de l'électricité et de l'électronique. A cet effet, l'IEC – entre autres activités – publie des Normes internationales, des Spécifications techniques, des Rapports techniques, des Spécifications accessibles au public (PAS) et des Guides (ci-après dénommés "Publication(s) de l'IEC"). Leur élaboration est confiée à des comités d'études, aux travaux desquels tout Comité national intéressé par le sujet traité peut participer. Les organisations internationales, gouvernementales et non gouvernementales, en liaison avec l'IEC, participent également aux travaux. L'IEC collabore étroitement avec l'Organisation Internationale de Normalisation (ISO), selon des conditions fixées par accord entre les deux organisations.
- 2) Les décisions ou accords officiels de l'IEC concernant les questions techniques représentent, dans la mesure du possible, un accord international sur les sujets étudiés, étant donné que les Comités nationaux de l'IEC intéressés sont représentés dans chaque comité d'études.
- 3) Les Publications de l'IEC se présentent sous la forme de recommandations internationales et sont agréées comme telles par les Comités nationaux de l'IEC. Tous les efforts raisonnables sont entrepris afin que l'IEC s'assure de l'exactitude du contenu technique de ses publications; l'IEC ne peut pas être tenue responsable de l'éventuelle mauvaise utilisation ou interprétation qui en est faite par un quelconque utilisateur final.
- 4) Dans le but d'encourager l'uniformité internationale, les Comités nationaux de l'IEC s'engagent, dans toute la mesure possible, à appliquer de façon transparente les Publications de l'IEC dans leurs publications nationales et régionales. Toutes divergences entre toutes Publications de l'IEC et toutes publications nationales ou régionales correspondantes doivent être indiquées en termes clairs dans ces dernières.
- 5) L'IEC elle-même ne fournit aucune attestation de conformité. Des organismes de certification indépendants fournissent des services d'évaluation de conformité et, dans certains secteurs, accèdent aux marques de conformité de l'IEC. L'IEC n'est responsable d'aucun des services effectués par les organismes de certification indépendants.
- 6) Tous les utilisateurs doivent s'assurer qu'ils sont en possession de la dernière édition de cette publication.
- 7) Aucune responsabilité ne doit être imputée à l'IEC, à ses administrateurs, employés, auxiliaires ou mandataires, y compris ses experts particuliers et les membres de ses comités d'études et des Comités nationaux de l'IEC, pour tout préjudice causé en cas de dommages corporels et matériels, ou de tout autre dommage de quelque nature que ce soit, directe ou indirecte, ou pour supporter les coûts (y compris les frais de justice) et les dépenses découlant de la publication ou de l'utilisation de cette Publication de l'IEC ou de toute autre Publication de l'IEC, ou au crédit qui lui est accordé.
- 8) L'attention est attirée sur les références normatives citées dans cette publication. L'utilisation de publications référencées est obligatoire pour une application correcte de la présente publication.
- 9) L'attention est attirée sur le fait que certains des éléments de la présente Publication de l'IEC peuvent faire l'objet de droits de brevet. L'IEC ne saurait être tenue pour responsable de ne pas avoir identifié de tels droits de brevets et de ne pas avoir signalé leur existence.

La Norme internationale IEC 62379-2 a été établie par le domaine technique 4: Interfaces du système numérique et protocoles, du comité d'études 100 de l'IEC: Systèmes et équipements audio, vidéo et services de données.

La présente version bilingue (2021-03) correspond à la version anglaise monolingue publiée en 2008-09.

La version française de cette norme n'a pas été soumise au vote.

Cette publication a été rédigée selon les Directives ISO/IEC, Partie 2.

Le comité a décidé que le contenu de cette publication ne sera pas modifié avant la date de maintenance indiquée sur le site web de l'IEC sous "<http://webstore.iec.ch>" dans les données relatives à la publication recherchée. A cette date, la publication sera

- reconduite,
- supprimée,
- remplacée par une édition révisée, ou
- amendée.

IECNORM.COM : Click to view the full PDF of IEC 62379-2:2008

INTRODUCTION

L'IEC 62379 spécifie l'interface de commande commune, un protocole de gestion de l'équipement qui achemine les données audio et/ou vidéo sur les réseaux numériques.

La présente partie de l'IEC 62379 précise les aspects spécifiques à l'équipement audio.

Une introduction à l'interface de commande commune est donnée dans l'IEC 62379-1.

IECNORM.COM : Click to view the full PDF of IEC 62379-2:2008

INTERFACE DE COMMANDE COMMUNE POUR PRODUITS AUDIO ET VIDÉO NUMÉRIQUES CONNECTÉS EN RÉSEAU –

Partie 2: Audio

1 Domaine d'application

La présente partie de l'IEC 62379 précise les aspects de l'interface de commande commune de l'IEC 62379-1 qui sont spécifiques à l'équipement audio.

2 Références normatives

Les documents suivants sont cités dans le texte de sorte qu'ils constituent, pour tout ou partie de leur contenu, des exigences du présent document. Pour les références datées, seule l'édition citée s'applique. Pour les références non datées, la dernière édition du document de référence s'applique (y compris les éventuels amendements).

AES3-2003, *AES standard for digital audio – Digital input-output interfacing – Serial transmission format for two-channel linearly represented digital audio data* (disponible en anglais seulement)

AES10-2003, *AES recommended practice for digital audio engineering – Serial multichannel audio digital interface (MADI)* (disponible en anglais seulement)

AES50-2005, *AES standard for digital audio engineering – High-resolution multi-channel audio interconnection (HRMAI)* (disponible en anglais seulement)

IEC 62379-1:2007, *Common control interface for networked audio and video products – Part 1: General* (disponible en anglais seulement)

Recommandation UIT-T G.711, *Modulation par impulsions et codage (MIC) des fréquences vocales*

Recommandation UIT-T G.722, *Codage audiofréquence à 7 kHz à un débit inférieur ou égal à 64 kbit/s*

Recommandation UIT-T J.41, *Caractéristiques des équipements de codage de signaux radiophoniques analogiques de haute qualité pour la transmission sur des voies à 384 kbit/s*

Recommandation UIT-T J.57, *Transmission de signaux audio avec la qualité studio numérique sur des canaux H1*

3 Termes, définitions et abréviations

Pour les besoins du présent document, les termes et définitions donnés dans l'IEC 62379-1 s'appliquent avec les abréviations suivantes.

3.1 Abréviations

3.1.1

modulation par impulsions et codage

MIC

3.1.2 groupe d'experts en images animées MPEG (motion pictures expert group)

3.1.3 codage audio avancé AAC (advanced audio coding)

4 Définitions de format audio

4.1 Définitions de formats de signal audio

En tout point de la chaîne de signaux audio, les données audio sont sous un format particulier. A des fins de gestion, le format doit être identifié par un identificateur d'objet, qui est soit un identificateur d'objet "common control interface standard" défini dans la présente norme, soit un identificateur d'objet défini ailleurs.

NOTE La possibilité de définir les identificateurs de format audio en dehors de la présente norme permet l'utilisation de formats propriétaires dans le protocole normalisé ainsi que l'émergence de formats normalisés de l'industrie qui peuvent éventuellement être incorporés dans de futures révisions de la présente norme.

4.1.1 Paramètres audio

4.1.1.1 Généralités

Les définitions données en 4.1.2 font référence aux paramètres qui fournissent des informations supplémentaires sur le format. Ces paramètres doivent être mappés vers les valeurs de sous-identificateur spécifiées dans les autres paragraphes du 4.1.1. Tout paramètre peut être non spécifié.

Les valeurs de sous-identificateur doivent être annexées aux identificateurs d'objet en tant qu'arcs supplémentaires, dans l'ordre dans lequel les paramètres sont répertoriés dans le paragraphe applicable du 4.1.2; toutefois, si un paramètre est non spécifié et qu'il s'agit du dernier paramètre ou que tous les paramètres suivants sont également non spécifiés, alors il doit être omis.

NOTE Pour tous les paramètres, "non spécifié" est codé par zéro; cette règle assure donc que l'OID ne se termine pas par un arc zéro.

EXEMPLE: Si les deux derniers paramètres sont la profondeur binaire et la fréquence d'échantillonnage, alors 16 bits et 48 kHz est codé 16.48000; 16 bits (avec une fréquence d'échantillonnage non spécifiée) est codé .16 et 48 kHz (avec une profondeur binaire non spécifiée) est codé .0.48000.

4.1.1.2 Configuration de canaux

Le sous-identificateur de la configuration de canaux doit être une valeur du type suivant:

```
ChannelArrangement ::= INTEGER {
    unspecified          (0),
    discreteMono         (1),
    stereo               (2),
    jointStereo          (3),
    surround             (4),
    surroundWithDownmix (5)
} (unspecified.. surroundWithDownmix)
```

discreteMono doit indiquer que chaque canal (en présence de plusieurs) est un signal audio distinct.

Il convient d'utiliser `stereo` uniquement avec un nombre pair de canaux; il doit indiquer que chaque paire de canaux (en présence de plusieurs) est un signal audio stéréo distinct, le premier canal de la paire étant le canal gauche.

Il convient d'utiliser `jointStereo` uniquement avec un nombre pair de canaux; il doit indiquer que chaque paire de canaux (en présence de plusieurs) est un signal audio stéréo distinct, le premier canal de la paire étant le canal M, et le second le canal S.

Il convient d'utiliser `surround` et `surroundWithDownmix` uniquement avec 3 à 8 canaux (inclus); chaque cas doit indiquer que les six premiers canaux correspondent respectivement aux canaux gauche, droit, centre, à effets basses fréquences, surround gauche et surround droit. `surround` doit indiquer que les deux canaux suivants correspondent (respectivement) aux canaux gauche et droit d'un signal stéréo adapté au décodage matriciel. `surroundWithDownmix` doit indiquer que les deux canaux suivants correspondent (respectivement) aux canaux gauche et droit d'un mixage réducteur stéréo.

Lorsque plusieurs configurations peuvent être utilisées pour décrire un format, il convient d'utiliser la valeur applicable la plus faible.

EXEMPLE 1 Une paire stéréo simple (2 canaux) peut être décrite par les valeurs `stereo` (2), `surround` (4) ou `surroundWithDownmix` (5). Il convient d'utiliser la plus faible de ces valeurs, c'est-à-dire 2.

EXEMPLE 2 Le son surround sans signal stéréo associé (6 canaux) peut être décrit par les valeurs `surround` (4) ou `surroundWithDownmix` (5). Il convient d'utiliser la plus faible de ces valeurs, c'est-à-dire la 4.

4.1.1.3 Nombre de canaux

Le sous-identificateur du nombre de canaux doit être une valeur du type suivant:

```
NumberChannels ::= INTEGER
-- Entier qui représente le nombre de canaux audio.
-- Une valeur de zéro doit indiquer "non spécifié".
```

4.1.1.4 Profondeur binaire

Le sous-identificateur de la profondeur binaire doit être une valeur du type suivant:

```
BitDepth ::= INTEGER
-- Entier qui représente la profondeur binaire audio en bits par
-- échantillon.
-- Une valeur de zéro doit indiquer "non spécifié".
```

4.1.1.5 Fréquence d'échantillonnage

Le sous-identificateur de la fréquence d'échantillonnage doit être une valeur du type suivant:

```
SamplingFrequency ::= INTEGER
-- Entier qui représente la fréquence d'échantillonnage audio en Hz.
-- Une valeur de zéro doit indiquer "non spécifié".
```

4.1.1.6 Débit binaire

Le sous-identificateur du débit binaire doit être une valeur du type suivant:

```
BitRate ::= INTEGER
-- Entier qui représente le débit binaire du signal codé en bits par
-- seconde.
-- Une valeur de zéro doit indiquer "non spécifié".
```

4.1.2 Formats de signal audio

La racine des formats de signal audio doit se trouver à l'emplacement suivant dans l'arborescence MIB:

```
iec62379          OBJECT IDENTIFIER ::= { iso(1) standard(0) 62379 }

audioFormat       OBJECT IDENTIFIER ::= { iec62379 audio(2) format(2) }

audioSignalFormat OBJECT IDENTIFIER ::= { audioFormat signal(1) }
```

Les définitions suivantes doivent être utilisées pour identifier les formats spécifiés.

NOTE L'Annexe A contient un exemple d'ensemble de formats définis par la présente norme.

4.1.2.1 Signal audio non spécifié

```
unspecifiedAudio OBJECT IDENTIFIER ::=
                                { audioSignalFormat unspecified(0) }
-- caractère générique - tout format pris en charge admis
```

4.1.2.2 Sans audio

```
noAudio          OBJECT IDENTIFIER ::= { audioSignalFormat none(1) }
-- indique que la sortie est muette
```

4.1.2.3 Signal audio analogique

```
analogueAudio    OBJECT IDENTIFIER ::=
                                { audioSignalFormat analogue(2) }
-- signal audio analogique
```

L'identificateur audio analogique doit comporter deux paramètres. Le premier doit être la configuration de canaux, et le second doit être le nombre de canaux.

4.1.2.4 Signal audio MIC

```
pcmAudio         OBJECT IDENTIFIER ::= { audioSignalFormat pcm(3) }
-- signal audio MIC linéaire
```

L'identificateur de format MIC linéaire doit comporter quatre paramètres. Le premier doit être la configuration de canaux, le deuxième doit être le nombre de canaux, le troisième doit être la profondeur binaire et le quatrième doit être la fréquence d'échantillonnage.

4.1.2.5 Signal audio MPEG-1, couche 2

```
mp2Audio         OBJECT IDENTIFIER ::= { audioSignalFormat mp2(4) }
-- signal audio MPEG-1, couche 2
```

L'identificateur de format MP2 doit comporter quatre paramètres. Le premier doit être la configuration de canaux, le deuxième doit être le nombre de canaux, le troisième doit être la fréquence d'échantillonnage et le quatrième doit être le débit binaire.

4.1.2.6 Signal audio MPEG-1, couche 3

```
mp3Audio         OBJECT IDENTIFIER ::= { audioSignalFormat mp3(5) }
-- signal audio MPEG-1, couche 3
```

L'identificateur de format MP3 doit comporter quatre paramètres. Le premier doit être la configuration de canaux, le deuxième doit être le nombre de canaux, le troisième doit être la fréquence d'échantillonnage et le quatrième doit être le débit binaire.

4.1.2.7 Signal audio AAC

```

aacAudio          OBJECT IDENTIFIER ::= { audioSignalFormat aac(6) }
-- signal audio AAC

aacLC              OBJECT IDENTIFIER ::= { aacAudio aacLC(1) }
-- signal audio AAC avec profil à faible complexité

aacMain            OBJECT IDENTIFIER ::= { aacAudio aacMain(2) }
-- signal audio AAC avec profil principal

aacSRS             OBJECT IDENTIFIER ::= { aacAudio aacSRS(3) }
-- signal audio AAC avec profil évolutif selon la fréquence
-- d'échantillonnage

aacLTP             OBJECT IDENTIFIER ::= { aacAudio aacLTP(4) }
-- signal audio AAC avec profil de prévision à long terme

aacLD              OBJECT IDENTIFIER ::= { aacAudio aacLD(5) }
-- signal audio AAC avec profil à faible latence

```

L'identificateur de format AAC doit comporter quatre paramètres. Le premier doit être la configuration de canaux, le deuxième doit être le nombre de canaux, le troisième doit être la fréquence d'échantillonnage et le quatrième doit être le débit binaire.

4.1.2.8 Signal audio conforme à la Recommandation UIT-T G.711

```

g711Audio          OBJECT IDENTIFIER ::= { audioSignalFormat g711(7) }
-- signal audio G711

g711ALaw           OBJECT IDENTIFIER ::= { g711 aLaw(1) }
-- signal audio G711, codé par loi A

g711MuLaw          OBJECT IDENTIFIER ::= { g711 muLaw(2) }
-- signal audio G711, codé par loi mu

```

4.1.2.9 Signal audio conforme à la Recommandation UIT-T G.722

```

g722Audio          OBJECT IDENTIFIER ::= { audioSignalFormat g722(8) }
-- signal audio G722

```

L'identificateur G722 doit comporter un paramètre. Il doit s'agir du débit binaire.

4.1.2.10 Signal audio APT-X

```

aptXAudio          OBJECT IDENTIFIER ::= { audioSignalFormat aptX(9) }
-- signal audio APT-X

```

L'identificateur de format APT-X doit comporter cinq paramètres. Le premier doit être la configuration de canaux, le deuxième doit être le nombre de canaux, le troisième doit être la profondeur binaire, le quatrième doit être la fréquence d'échantillonnage et le cinquième doit être le débit binaire.

4.1.2.11 Signal audio APT-X amélioré

```

enhancedAptXAudio  OBJECT IDENTIFIER ::= { audioSignalFormat enhAptX(10) }
-- signal audio APT-X amélioré

```

L'identificateur de format APT-X amélioré doit comporter cinq paramètres. Le premier doit être la configuration de canaux, le deuxième doit être le nombre de canaux, le troisième doit être la profondeur binaire, le quatrième doit être la fréquence d'échantillonnage et le cinquième doit être le débit binaire.

4.1.2.12 Signal audio conforme à la Recommandation UIT-T J.41

```
j41Audio          OBJECT IDENTIFIER ::= { audioSignalFormat j41(11) }
-- signal audio J41

j41ALawA          OBJECT IDENTIFIER ::= { j41 aLawA(1) }
-- signal audio J41 qui utilise une compression-expansion de loi A,
-- variante A

j41ALawB          OBJECT IDENTIFIER ::= { j41 aLawB(2) }
-- signal audio J41 qui utilise une compression-expansion de loi A,
-- variante B

j41Nic            OBJECT IDENTIFIER ::= { j41 nic(3) }
-- signal audio J41 qui utilise une compression-expansion quasi instantanée
```

4.1.2.13 Signal audio conforme à la Recommandation UIT-T J.57

```
j57Audio          OBJECT IDENTIFIER ::= { audioSignalFormat j57(12) }
-- signal audio J57

j57H11           OBJECT IDENTIFIER ::= { j57 h11(1) }
-- signal audio J57 qui utilise un canal H11

j57H12           OBJECT IDENTIFIER ::= { j57 h12(2) }
-- signal audio J57 qui utilise un canal H12
```

4.1.2.14 Signal audio non valide

```
invalidAudio      OBJECT IDENTIFIER ::= { audioSignalFormat invalid(13)
}
-- indique une erreur, telle que l'incapacité à décoder un signal en amont
-- dans la chaîne
```

4.2 Définitions de formats de transport audio

La racine des formats de transport audio doit se trouver à l'emplacement suivant dans l'arborescence MIB:

```
audioTransportFormat OBJECT IDENTIFIER ::= { audioFormat transport(2) }
```

Les définitions suivantes doivent être utilisées pour identifier les formats de transport spécifiés.

```
unspecifiedTransport OBJECT IDENTIFIER ::=
{ audioTransportFormat unspecified(0) }

analogue          OBJECT IDENTIFIER ::= { audioTransportFormat analogue(1) }

AES3              OBJECT IDENTIFIER ::= { audioTransportFormat aes3(2) }

AES10             OBJECT IDENTIFIER ::= { audioTransportFormat aes10(3) }

AES50             OBJECT IDENTIFIER ::= { audioTransportFormat aes50(4) }
```

4.3 Définitions de formats de métadonnées audio

La racine des formats de métadonnées audio doit se trouver à l'emplacement suivant dans l'arborescence MIB:

```
audioMetadataFormat OBJECT IDENTIFIER ::= { audioFormat metadata(3) }
```

Les définitions suivantes doivent être utilisées pour identifier les formats de métadonnées spécifiés.

```
unspecifiedMetadata OBJECT IDENTIFIER ::=
    { audioMetadataFormat unspecified(0) }
```

5 Définitions MIB pour blocs audio

5.1 Généralités

Le présent article définit un ensemble de types d'objets gérés pour représenter les fonctions de commande dans les équipements audio commandés par réseau. Le format des définitions est conforme à l'IEC 62379-1.

A des fins de gestion, un équipement audio doit être modélisé par plusieurs blocs et connecteurs audio distincts, comme spécifié dans l'IEC 62379-1. Chaque bloc audio peut avoir zéro ou plusieurs entrées et zéro ou plusieurs sorties, et chaque entrée ou sortie peut comporter un ou plusieurs canaux. Chaque connecteur audio doit connecter une sortie de bloc audio à une entrée de bloc audio avec un mappage biunivoque des canaux entre les blocs.

NOTE 1 Un équipement peut être à fonction fixe, auquel cas le nombre de blocs audio présents et les connexions entre eux sont immuables, ou il peut être programmable, auquel cas le nombre de blocs audio présents et/ou les connexions entre eux peuvent être modifiés par l'utilisateur.

Chaque bloc audio doit être modélisé soit par l'un des types de blocs audio normalisés définis dans la présente norme, soit par un type de bloc audio défini ailleurs. A chaque type de bloc défini doit être associé un groupe (éventuellement vide) de types d'objets gérés, qui représente les fonctions de commande du bloc. Un type de bloc doit être identifié par le nœud dans l'arborescence d'identificateurs d'objets qui correspond au nœud racine du groupe de types d'objets gérés associé à ce type de bloc.

NOTE 2 La possibilité de définir les types de blocs audio en dehors de la présente norme permet la commande de fonctions propriétaires à l'aide du protocole normalisé ainsi que l'émergence de types de blocs normalisés de l'industrie qui peuvent éventuellement être incorporés dans de futures révisions de la présente norme.

NOTE 3 Il est admis de disposer d'un groupe vide de types d'objets gérés, afin de permettre l'existence de blocs qui n'ont aucune fonction de commande associée.

NOTE 4 L'Annexe E contient des exemples pratiques de structures de bloc.

5.2 Définitions de types

En plus des types définis dans l'IEC 62379-1, les types suivants sont utilisés pour spécifier la syntaxe des structures de données abstraites qui représentent les valeurs d'objets gérés.

5.2.1 Conventions textuelles

```
AudioTransportType ::= OBJECT IDENTIFIER
-- Référence au transport utilisé pour une connexion audio.
-- La valeur peut être définie en 4.2 ou dans une sous-partie de
-- l'IEC 62379-5, ou ailleurs.
```

```
AudioLevel ::= INTEGER {
    mInfinity (-20000),
    fullScale (0),
    pInfinity (20000)
} (mInfinity..pInfinity)
-- Niveau audio absolu ou relatif en unités de 0,01 dB.
```

```
AudioPhase ::= INTEGER (-18000..18000)
```

```
-- Valeur de phase absolue ou relative en unités de 0,01 degré.

AudioQuality ::= INTEGER {
    worst (1),
    low   (32),
    high  (96),
    best  (127)
} (worst..best)
-- Énumération qui identifie un niveau de qualité de traitement du signal
-- audio. La sémantique est spécifique à l'équipement. Les équipements qui
-- prennent en charge les niveaux de qualité sélectionnables doivent
-- au moins prendre en charge les valeurs "low" et "high".

AudioChannel ::= INTEGER {
    left  (1),
    right (2)
} (1..240)
-- Énumération qui identifie un canal audio. L'équipement stéréo doit
-- accepter les valeurs "left" et "right". Il convient qu'un équipement
-- multicanal accepte une plage contiguë de valeurs à partir de 1; il peut
-- ignorer l'association des valeurs 1 et 2 avec "left" et "right".

Aes3ChannelData ::= OCTET STRING (SIZE(24))
-- Données d'état de canal associées à un canal AES3.

Aes3UserData ::= OCTET STRING (SIZE(24))
-- Données utilisateur associées à un canal AES3.

AudioClipSelector ::= INTEGER (1..2147483647)
-- Énumération qui identifie un clip audio. La sémantique est spécifique
-- à l'équipement.

AudioClipActivation ::= INTEGER {
    play      (1),
    stop      (2),
    stopAtEnd (3)
} (play..stopAtEnd)
-- Énumération pour la commande de lecture de clip audio.

AudioClipTrackControl ::= INTEGER {
    previous (1),
    next     (2)
} (previous..next)
-- Énumération pour la commande de piste de clip audio.

AudioClipRepeat ::= INTEGER {
    none  (1),
    track (2),
    all   (3)
} (none..all)
-- Énumération pour la commande de répétition de clip audio.

AudioRecoveryMode ::= INTEGER {
    auto (1),
    slow (2),
    fast (3)
} (auto..fast)
-- Énumération qui identifie un mode de récupération du limiteur audio. La
-- sémantique est spécifique à l'équipement.

AudioAlarmStatus ::= INTEGER {
    ok      (1),
```

```

    warning (2),
    failure (3)
} (ok..failure)
-- Enumération qui représente l'état d'une alarme audio.

```

```

AudioLevelAlarmType ::= INTEGER {
    lower (1),
    higher (2)
} (lower..higher)
-- Enumération qui représente la condition d'activation de l'alarme de
-- niveau audio. Si la valeur est "lower", l'alarme est activée lorsque le
-- niveau audio est inférieur au seuil; si la valeur est "higher", l'alarme
-- est activée lorsque le niveau audio est supérieur au seuil.

```

5.2.2 Séquences

```

APortEntry ::= SEQUENCE {
    aPortBlockId      BlockId,
    aPortDirection    PortDirection,
    aPortFormat        MediaFormat,
    aPortTransport     AudioTransportType,
    aPortName          Utf8String
}

Aes3DataEntry ::= SEQUENCE {
    aes3BlockId        BlockId,
    aes3ChannelId       AudioChannel,
    aes3ChannelData     Aes3ChannelData,
    aes3UserData         Aes3UserData,
    aes3ValidityError   TruthValue
}

APhantomEntry ::= SEQUENCE {
    aPhantomBlockId    BlockId,
    aPhantomEnabled     TruthValue,
    aPhantomLevel       CardinalNumber
}

ALockedEntry ::= SEQUENCE {
    aLockedBlockId      BlockId,
    aLockedTime          CardinalNumber,
    aLockedSamplesInserted CardinalNumber,
    aLockedSamplesDropped CardinalNumber
}

AMixerBlockEntry ::= SEQUENCE {
    aMixerBlockId        BlockId,
    aMixerFadeDuration    CardinalNumber,
    aMixerFadeNow         TruthValue
}

AMixerInputEntry ::= SEQUENCE {
    aMixerInputBlockId    BlockId,
    aMixerInputNumber      IndexNumber,
    aMixerInputLevel       AudioLevel,
    aMixerInputFadeToLevel AudioLevel,
    aMixerInputDelay       CardinalNumber
}

ACrosspointBlockEntry ::= SEQUENCE {
    aCrosspointBlockId    BlockId,
    aCrosspointConfigure   TruthValue,
    aCrosspointCopy        BlockId
}

```

}

```
ACrosspointPathEntry ::= SEQUENCE {
    aCrosspointPathBlockId  BlockId,
    aCrosspointPathSrc      AudioChannel,
    aCrosspointPathDst      AudioChannel,
    aCrosspointPathGain     AudioLevel,
    aCrosspointPathNewGain  AudioLevel,
    aCrosspointPathPhase    AudioPhase,
    aCrosspointPathNewPhase AudioPhase
}
```

```
AClipPlayerBlockEntry ::= SEQUENCE {
    acpBlockId          BlockId,
    acpSelection         AudioClipSelector,
    acpMainFaderLevelNow AudioLevel,
    acpMainFaderLevelStart AudioLevel,
    acpMainFaderLevelStop AudioLevel,
    acpMainFaderDuration CardinalNumber,
    acpClipFaderLevelNow AudioLevel,
    acpClipFaderLevelStart AudioLevel,
    acpClipFaderLevelStop AudioLevel,
    acpClipFaderDuration CardinalNumber,
    acpStartDelay       IntegerNumber,
    acpActivation        AudioClipActivation,
    acpTrackControl      AudioClipTrackControl,
    acpTrackNumber       CardinalNumber,
    acpRepeat            AudioClipRepeat
}
```

```
AClipSequenceEntry ::= SEQUENCE {
    aClipSequenceBlockId BlockId,
    aClipSequenceId      AudioClipSelector,
    aClipSequenceName    Utf8String
}
```

```
ALimiterBlockEntry ::= SEQUENCE {
    aLimiterBlockId      BlockId,
    aLimiterThreshold    AudioLevel,
    aLimiterAttackTime   CardinalNumber,
    aLimiterGainMakeup   AudioLevel,
    aLimiterRecoveryTime CardinalNumber,
    aLimiterRecoveryMode AudioRecoveryMode
}
```

```
AConverterBlockEntry ::= SEQUENCE {
    aConverterBlockId      BlockId,
    aConverterQuality       AudioQuality,
    aConverterEnabled       TruthValue,
    aConverterDithering     TruthValue,
    aConverterOutputFormat  MediaFormat,
    aConverterError         TruthValue
}
```

```
ALevelAlarmBlockEntry ::= SEQUENCE {
    alaBlockId      BlockId,
    alaType         AudioLevelAlarmType,
    alaThreshold    AudioLevel,
    alaWarningTime  CardinalNumber,
    alaFailureTime  CardinalNumber,
    alaCounter      CardinalNumber,
    alaEnabled      TruthValue,
    alaStatus       AudioAlarmStatus
}
```

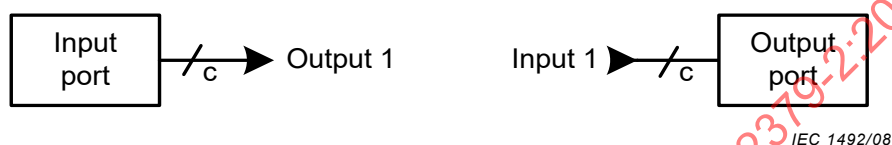
}

5.3 Définitions d'accès audio et de types d'objets gérés associés

5.3.1 Fonctionnalité d'accès générique

Toutes les entrées et sorties audio de l'unité doivent être représentées à l'aide d'un bloc d'accès audio. Le tableau des objets gérés de base fournit une commande commune pour tous les accès; les tableaux d'extension fournissent les fonctionnalités spécifiques à certains types d'accès.

Un bloc d'accès audio doit présenter la structure suivante, où *c* est le nombre de canaux sur une entrée ou une sortie:



Anglais	Français
Input port	Accès d'entrée
Output 1	Sortie 1
Input 1	Entrée 1
Output port	Accès de sortie

Figure 1 – Blocs d'accès audio

Le groupe d'objets du Tableau 1 doit être mis en œuvre par tous les équipements audio conformes qui comportent un ou plusieurs accès audio. Le nœud racine pour ces objets doit être

```
{ iso(1) standard(0) iec62379 audio(2) audioMIB(1) audioPort(1) }
```

Ce nœud doit être utilisé comme identificateur de type de bloc pour les blocs d'accès audio.

Tableau 1 – Objets gérés pour accès audio

Identificateur	Syntaxe	Indice	Lisible	Inscriptible	Volatile	Statut
aPortTable(1)	SEQUENCE OF APortEntry		aucun	aucun	non	o
lAPortEntry(1)	APortEntry		aucun	aucun	non	o
lAPortBlockId(1)	BlockId	oui	aucun	aucun	non	o
lAPortDirection(2)	PortDirection		auditeur	aucun	non	o
lAPortFormat(3)	MediaFormat		auditeur	aucun	oui	o
lAPortTransport(4)	AudioTransportType		auditeur	aucun	non	f
lAPortName(5)	Utf8String		auditeur	superviseur	non	f

5.3.1.1 aPortTable

Tableau des descripteurs d'accès audio pour cette unité. Chaque accès audio de l'unité possède une entrée correspondante dans ce tableau.

5.3.1.2 aPortEntry

Entrée dans le tableau des accès audio.

5.3.1.3 aPortBlockId

Identificateur de bloc pour cet accès. Utilisé comme indice lors de l'accès au tableau des blocs d'accès audio.

5.3.1.4 aPortDirection

Direction (entrée ou sortie) de l'accès.

5.3.1.5 aPortFormat

Format des données audio actuellement reçues ou transmises par l'accès. Si l'accès n'est pas actif, la valeur noAudio doit être renvoyée.

5.3.1.6 aPortTransport

Type de transport utilisé par l'accès.

5.3.1.7 aPortName

Nom attribué à l'accès. Il s'agit d'une chaîne de texte arbitraire attribuée par le gestionnaire du système.

5.3.2 Données auxiliaires AES3

Le groupe d'objets du Tableau 2 doit être mis en œuvre par tous les équipements audio conformes qui permettent l'accès aux données auxiliaires AES3. Le nœud racine pour ces objets doit être

{ iso(1) standard(0) iec62379 audio(2) audioMIB(1) audioPort(1) }

Tableau 2 – Objets gérés pour données auxiliaires AES3

Identificateur	Syntaxe	Indice	Lisible	Inscriptible	Volatile	Statut
aes3DataTable (2)	SEQUENCE OF Aes3DataEntry		aucun	aucun	non	o
l aes3DataEntry (1)	Aes3DataEntry		aucun	aucun	non	o
l aes3BlockId (1)	BlockId	oui	aucun	aucun	non	o
l aes3ChannelId (2)	AudioChannel	oui	aucun	aucun	non	o
l aes3ChannelData (3)	Aes3ChannelData		auditeur	aucun	oui	f
l aes3UserData (4)	Aes3UserData		auditeur	aucun	oui	f
l aes3ValidityError (5)	TruthValue		auditeur	superviseur	oui	f

5.3.2.1 aes3DataTable

Tableau des descripteurs de données auxiliaires AES3 pour cette unité. Chaque canal de chaque accès AES3 de l'unité possède une entrée correspondante dans ce tableau.

5.3.2.2 aes3DataEntry

Entrée dans le tableau des données auxiliaires AES3.

5.3.2.3 aes3BlockId

Identificateur de bloc de l'accès audio associé. Utilisé comme indice lors de l'accès au tableau des données auxiliaires AES3.

5.3.2.4 aes3ChannelId

Numéro de canal. Utilisé comme indice lors de l'accès au tableau des données auxiliaires AES3.

5.3.2.5 aes3ChannelData

Données d'état de canal actuelles (24 octets) pour ce canal.

5.3.2.6 aes3UserData

Données utilisateur actuelles pour ce canal.

5.3.2.7 aes3ValidityError

Indication qui détermine si le bit de validité du canal a été défini dans une sous-trame. Il doit être automatiquement défini sur `true` lorsqu'une sous-trame avec `V = 1` est détectée. Il doit être défini sur `false` uniquement à la suite d'une opération `SET` sur cet objet.

NOTE `V = 1` indique une erreur dans le cas de MIC linéaire; pour les autres formats, il convient de définir `V = 1` dans chaque sous-trame.

5.3.3 Alimentation fantôme

Le groupe d'objets du Tableau 3 doit être mis en œuvre par tous les équipements audio conformes qui fournissent une alimentation fantôme sur un accès audio. Le nœud racine pour ces objets doit être

```
{ iso(1) standard(0) iec62379 audio(2) audioMIB(1) audioPort(1) }
```

Tableau 3 – Objets gérés pour alimentation fantôme

Identificateur	Syntaxe	Indice	Lisible	Inscriptible	Volatile	Statut
aPhantomTable(3)	SEQUENCE OF APhantomEntry		aucun	aucun	non	o
lAPhantomEntry(1)	APhantomEntry		aucun	aucun	non	o
lAPhantomBlockId(1)	BlockId	oui	aucun	aucun	non	o
lAPhantomEnabled(2)	TruthValue		auditeur	opérateur	non	o
lAPhantomLevel(3)	CardinalNumber		auditeur	superviseur	non	f

5.3.3.1 aPhantomTable

Tableau des descripteurs d'alimentation fantôme d'accès audio pour cette unité. Chaque accès audio de l'unité qui prend en charge une alimentation fantôme possède une entrée correspondante dans ce tableau.

5.3.3.2 aPhantomEntry

Entrée dans le tableau des alimentations fantômes.

5.3.3.3 aPhantomBlockId

Identificateur de bloc de l'accès audio associé. Utilisé comme indice lors de l'accès au tableau des alimentations fantômes.

5.3.3.4 aPhantomEnabled

Lorsque la valeur est définie sur `true`, l'alimentation fantôme est fournie par cet accès. Lorsque la valeur est définie sur `false`, l'alimentation fantôme n'est pas fournie par cet accès.

5.3.3.5 aPhantomLevel

Niveau d'alimentation fantôme fourni en millivolts.

5.3.4 Signal audio verrouillé sur une référence

Le groupe d'objets du Tableau 4 doit être mis en œuvre par tous les équipements audio conformes qui fournissent des statistiques concernant les signaux audio verrouillés sur un signal de référence. Le nœud racine pour ces objets doit être

```
{ iso(1) standard(0) iec62379 audio(2) audioMIB(1) audioPort(1) }
```

Tableau 4 – Objets gérés pour signal audio verrouillé

Identificateur	Syntaxe	Indice	Lisible	Inscriptible	Volatile	Statut
aLockedTable(3)	SEQUENCE OF ALockedEntry		aucun	aucun	non	o
lALockedEntry(1)	ALockedEntry		aucun	aucun	non	o
lALockedBlockId(1)	BlockId	oui	aucun	aucun	non	o
lALockedTime(2)	CardinalNumber		auditeur	aucun	oui	o
lALockedSamplesInserted(3)	CardinalNumber		auditeur	superviseur	oui	o
lALockedSamplesDropped(4)	CardinalNumber		auditeur	superviseur	oui	o

5.3.4.1 aLockedTable

Tableau des descripteurs d'accès audio verrouillés pour cette unité. Chaque accès audio de l'unité qui prend en charge la mesure du verrouillage par rapport à une référence possède une entrée correspondante dans ce tableau.

5.3.4.2 aLockedEntry

Entrée dans le tableau des accès verrouillés.

5.3.4.3 aLockedBlockId

Identificateur de bloc de l'accès audio associé. Utilisé comme indice lors de l'accès au tableau des accès verrouillés.

5.3.4.4 aLockedTime

Temps (en secondes) pendant lequel le signal audio a été verrouillé sur cet accès par rapport à l'horloge de référence de l'unité.

5.3.4.5 aLockedSamplesInserted

Nombre d'échantillons audio qui ont été insérés en raison d'une sous-utilisation ou d'une sur-utilisation de la mémoire tampon sur cet accès. La valeur doit être mise à jour tous les 10 s en ajoutant le nombre d'échantillons insérés au cours des 10 dernières secondes à la moitié de la valeur précédente.

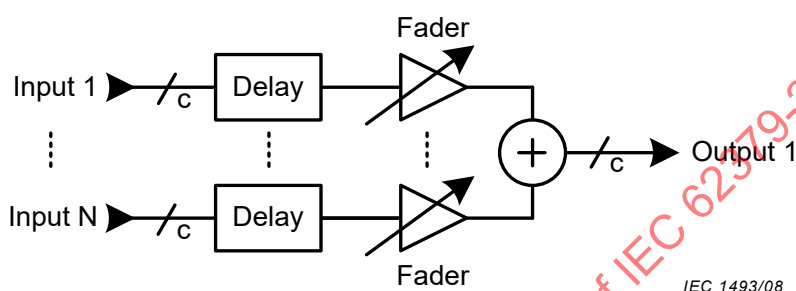
5.3.4.6 aLockedSamplesDropped

Nombre d'échantillons audio qui ont été abandonnés en raison d'une sous-utilisation ou d'une sur-utilisation de la mémoire tampon sur cet accès. La valeur doit être mise à jour tous les 10 s en ajoutant le nombre d'échantillons insérés au cours des 10 dernières secondes à la moitié de la valeur précédente.

5.4 Autres définitions de blocs audio et de types d'objets gérés associés

5.4.1 Blocs de mélangeur audio

Un bloc de mélangeur audio doit avoir la structure suivante:



Anglais	Français
Input 1	Entrée 1
Delay	Latence
Input N	Entrée N
Output 1	Sortie 1

Figure 2 – Bloc de mélangeur audio

où c est le nombre de canaux sur une connexion.

NOTE 1 Un bloc de mélangeur audio peut être utilisé pour représenter un sélecteur commuté simple ou un combineur, en limitant les valeurs admises pour les commandes de niveau de fader à `mInfinity` ou `fullScale`.

NOTE 2 La fonction de latence permet le réaligement temporel des flux audio qui ont traversé différents chemins de traitement ou de transport, soit avec d'autres flux audio, soit avec des flux vidéo associés. Les équipements qui ne prennent pas en charge cette fonctionnalité sont représentés comme ayant une latence fixe de zéro.

Le groupe d'objets du Tableau 5 doit être mis en œuvre par tout équipements audio conforme qui dispose d'un modèle de gestion qui comprend un ou plusieurs blocs de mélangeur audio. Le nœud racine pour ces objets doit être

```
{ iso(1) standard(0) iec62379 audio(2) audioMIB(1) audioMixer(2) }
```

Ce nœud doit être utilisé comme identificateur de type de bloc pour les blocs de mélangeur audio.

Tableau 5 – Objets gérés pour blocs de mélangeur audio

Identificateur	Syntaxe	Indice	Lisible	Inscriptible	Volatile	Statut
aMixerBlockTable (1)	SEQUENCE OF AMixerBlockEntry		aucun	aucun	non	o
LaMixerBlockEntry (1)	AMixerBlockEntry		aucun	aucun	non	o
LaMixerBlockId (1)	BlockId	oui	aucun	aucun	non	o
LaMixerFadeDuration (2)	CardinalNumber		auditeur	opérateur	non	f
LaMixerFadeNow (3)	TruthValue		auditeur	opérateur	oui	f
aMixerInputTable (2)	SEQUENCE OF AMixerInputEntry		aucun	aucun	non	o
LaMixerInputEntry (1)	AMixerInputEntry		aucun	aucun	non	o
LaMixerInputBlockId (1)	BlockId	oui	aucun	aucun	non	o
LaMixerInputNumber (2)	IndexNumber	oui	aucun	aucun	non	o
LaMixerInputLevel (3)	AudioLevel		auditeur	opérateur	non	o
LaMixerInputFadeToLevel (4)	AudioLevel		auditeur	opérateur	non	f
LaMixerInputDelay (5)	CardinalNumber		auditeur	opérateur	non	f

5.4.1.1 aMixerBlockTable

Tableau des descripteurs de bloc de mélangeur audio pour cette unité. Chaque bloc de mélangeur audio de l'unité possède une entrée correspondante dans ce tableau.

5.4.1.2 aMixerBlockEntry

Entrée dans le tableau des blocs de mélangeur audio.

5.4.1.3 aMixerBlockId

Identificateur de bloc pour ce bloc. Utilisé comme indice lors de l'accès au tableau des blocs de mélangeur audio.

5.4.1.4 aMixerFadeDuration

Temps de transition (en millisecondes) pour une opération de fondu lent ou de fondu enchaîné effectuée par ce bloc. Une valeur de zéro indique un basculement brusque.

5.4.1.5 aMixerFadeNow

Lorsque la valeur est définie sur `true`, le bloc effectue une opération de fondu lent ou de fondu enchaîné. A l'aide des valeurs issues du tableau des entrées de mélangeur pour le bloc, l'unité doit simultanément augmenter ou diminuer le niveau de fader pour chaque entrée dans `aMixerInputEntry` de la valeur actuelle vers la valeur spécifiée par `aMixerInputFadeToLevel`. La valeur est automatiquement réinitialisée sur `false` lorsque l'opération est terminée.

5.4.1.6 aMixerInputTable

Tableau des descripteurs d'entrée de mélangeur audio pour cette unité. Chaque entrée de chaque bloc de mélangeur audio de l'unité possède une entrée correspondante dans ce tableau.

5.4.1.7 aMixerInputEntry

Entrée dans le tableau des entrées de mélangeur audio.

5.4.1.8 **aMixerInputBlockId**

Identificateur de bloc du bloc associé. Utilisé comme indice lors de l'accès au tableau des entrées de mélangeur audio.

5.4.1.9 **aMixerInputNumber**

Numéro d'entrée de bloc pour cette entrée. Utilisé comme indice lors de l'accès au tableau des entrées de mélangeur audio.

5.4.1.10 **aMixerInputLevel**

Niveau de fader pour cette entrée. Lorsqu'un niveau est défini, le fader bascule immédiatement sur le niveau spécifié, pendant la durée `aMixerFadeDuration`. Pour les blocs qui prennent uniquement en charge la commutation entre les entrées, les seules valeurs admises sont `mInfinity` et `fullScale`. Les blocs qui basculent automatiquement d'une entrée à l'autre peuvent rejeter les opérations `SET` sur cet objet.

5.4.1.11 **aMixerInputFadeToLevel**

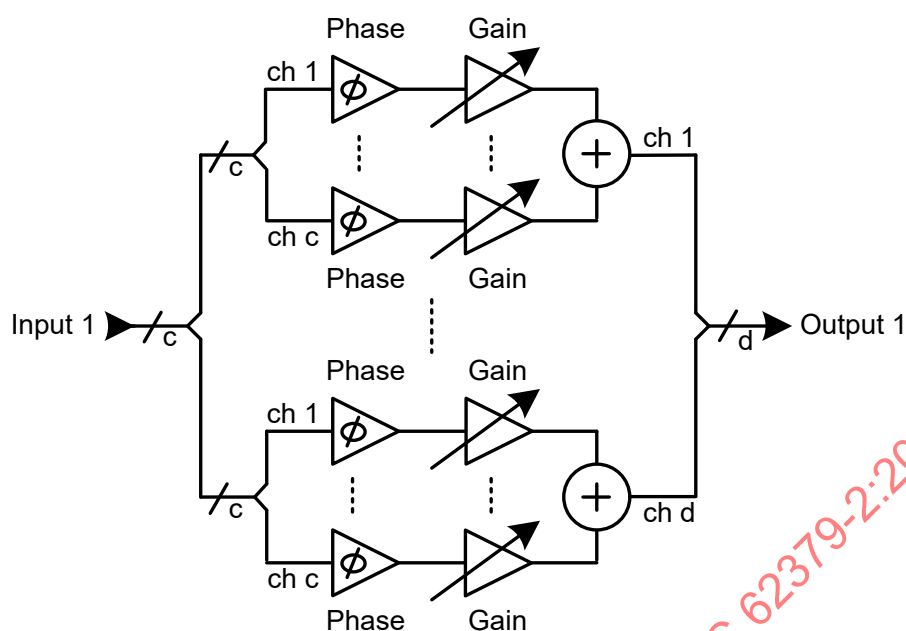
Niveau de fader pour cette entrée, appliqué lorsque `aMixerFadeNow` est défini sur `true`. Pour les blocs qui prennent uniquement en charge la commutation entre les entrées, les seules valeurs admises sont `mInfinity` et `fullScale`. Les blocs qui basculent automatiquement d'une entrée à l'autre peuvent rejeter les opérations `SET` sur cet objet.

5.4.1.12 **aMixerInputDelay**

Latence (en microsecondes) appliquée aux échantillons qui arrivent à cette entrée.

5.4.2 **Blocs de point de connexion audio**

Un bloc de point de connexion audio doit avoir la structure suivante:



IEC 1494/08

Anglais	Français
Input 1	Entrée 1
ch	canal
Output 1	Sortie 1

Figure 3 – Bloc de point de connexion audio

où c représente le nombre de canaux d'entrée, et d le nombre de canaux de sortie.

Le groupe d'objets du Tableau 6 doit être mis en œuvre par tout équipements audio conforme qui dispose d'un modèle de gestion qui comprend un ou plusieurs blocs de point de connexion audio. Le nœud racine pour ces objets doit être

{ iso(1) standard(0) iec62379 audio(2) audioMIB(1) audioCrosspoint(3) }

Ce nœud doit être utilisé comme identificateur de type de bloc pour les blocs de point de connexion audio.

Tableau 6 – Objets gérés pour blocs de point de connexion audio

Identificateur	Syntaxe	Indice	Lisible	Inscriptible	Volatile	Statut
aCrosspointBlockTable(1)	SEQUENCE OF ACrosspointBlockEntry		aucun	aucun	non	o
lACrosspointBlockEntry(1)	ACrosspointBlockEntry		aucun	aucun	non	o
lACrosspointBlockId(1)	BlockId	oui	aucun	aucun	non	o
lACrosspointConfigure(2)	TruthValue		auditeur	opérateur	oui	o
lACrosspointCopy(3)	BlockId		aucun	opérateur	oui	f
aCrosspointPathTable(2)	SEQUENCE OF ACrosspointPathEntry		aucun	aucun	non	o
lACrosspointPathEntry(1)	ACrosspointPathEntry		aucun	aucun	non	o
lACrosspointPathBlockId(1)	BlockId	oui	aucun	aucun	non	o
lACrosspointPathSrc(2)	AudioChannel	oui	aucun	aucun	non	o
lACrosspointPathDst(3)	AudioChannel	oui	aucun	aucun	non	o
lACrosspointPathGain(4)	AudioLevel		auditeur	opérateur	non	o
lACrosspointPathNewGain(5)	AudioLevel		auditeur	opérateur	non	f
lACrosspointPathPhase(6)	AudioPhase		auditeur	opérateur	non	o

Identificateur	Syntaxe	Indice	Lisible	Inscriptible	Volatile	Statut
<code>LaCrosspointPathNewPhase</code> (7)	<code>AudioPhase</code>		auditeur	opérateur	non	f

5.4.2.1 **aCrosspointBlockTable**

Tableau des descripteurs de bloc de point de connexion audio pour cette unité. Chaque bloc de point de connexion audio de l'unité possède une entrée correspondante dans ce tableau.

5.4.2.2 **aCrosspointBlockEntry**

Entrée dans le tableau des blocs de point de connexion audio.

5.4.2.3 **aCrosspointBlockId**

Identificateur de bloc pour ce bloc. Utilisé comme indice lors de l'accès au tableau des blocs de point de connexion audio.

5.4.2.4 **aCrosspointConfigure**

Lorsque la valeur est définie sur `true`, ce bloc est configuré comme spécifié par les entrées `aCrosspointPathNewGain` et `aCrosspointPathNewPhase` correspondantes dans le tableau des chemins de point de connexion. La valeur est automatiquement réinitialisée sur `false` si le bloc prend en charge la configuration latente et si l'une des entrées correspondantes du tableau des chemins de point de connexion est modifiée. La valeur reste en permanence sur `true` si le bloc ne prend pas en charge la configuration latente.

5.4.2.5 **aCrosspointCopy**

Lorsque la valeur est définie sur un identificateur de bloc qui identifie un autre bloc de point de connexion audio dans l'unité avec une structure identique à ce bloc, elle permet de copier les valeurs de `aCrosspointPathGain` et de `aCrosspointPathPhase` pour chaque chemin de point de connexion du bloc identifié vers ce bloc. Si la valeur est définie sur un identificateur de bloc qui n'identifie pas de bloc de point de connexion audio dans l'unité avec une structure identique à ce bloc, l'opération `SET` doit être rejetée.

NOTE Une application possible porte sur la capacité à disposer de configurations communes disponibles sous forme de préréglages en créant des blocs de point de connexion "factices" avec les paramètres exigés, lesquels sont référencés dans le tableau des blocs, mais ne font pas réellement partie du chemin audio.

5.4.2.6 **aCrosspointPathTable**

Tableau des descripteurs de chemin de point de connexion audio pour cette unité. Chaque chemin de chaque bloc de point de connexion audio de l'unité possède une entrée correspondante dans ce tableau.

5.4.2.7 **aCrosspointPathEntry**

Entrée dans le tableau des chemins de point de connexion audio.

5.4.2.8 **aCrosspointPathBlockId**

Identificateur de bloc du bloc associé. Utilisé comme indice lors de l'accès au tableau des chemins de point de connexion audio.

5.4.2.9 **aCrosspointPathSrc**

Canal audio source associé à ce chemin. Utilisé comme indice lors de l'accès au tableau des chemins de point de connexion audio.

5.4.2.10 aCrosspointPathDst

Canal audio de destination associé à ce chemin. Utilisé comme indice lors de l'accès au tableau des chemins de point de connexion audio.

5.4.2.11 aCrosspointPathGain

Gain de signal de la source à la destination de ce chemin. Si cette valeur est définie, la modification se produit immédiatement.

5.4.2.12 aCrosspointPathNewGain

Gain de signal de la source à la destination de ce chemin, appliqué lorsque aCrosspointConfigure est défini sur true.

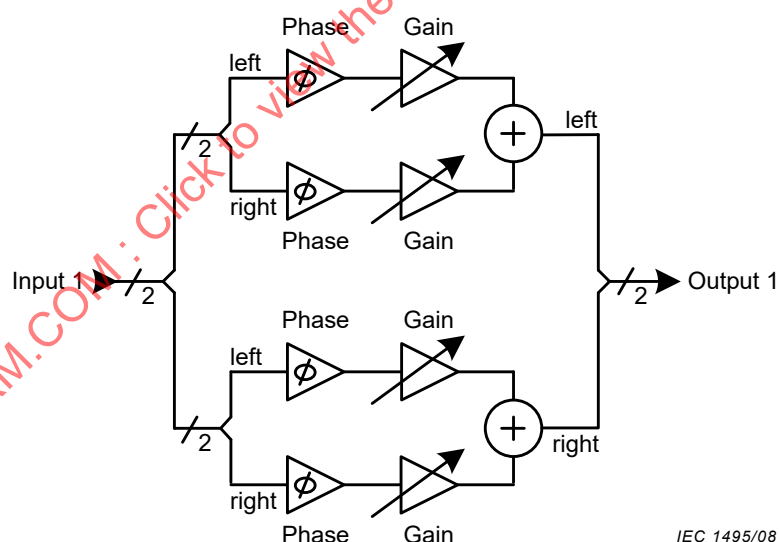
5.4.2.13 aCrosspointPathPhase

Déphasage de la source à la destination de ce chemin. Si cette valeur est définie, la modification se produit immédiatement.

5.4.2.14 aCrosspointPathNewPhase

Déphasage de la source à la destination de ce chemin, appliqué lorsque aCrosspointConfigure est défini sur true.

EXEMPLE: Le bloc de point de connexion permet de modéliser un ensemble de fonctions audio. A titre d'exemple, prendre le point de connexion stéréo suivant:



Anglais	Français
Input 1	Entrée 1
left	gauche
right	droite
Output 1	Sortie 1

Il existe quatre routages par le point de connexion qui représentent l'effet d'un canal d'entrée (la source, représentée par aCrosspointPathSrc) sur un canal de sortie (la destination, représentée par aCrosspointPathDst). Les paires source/destination représentent:

- l'effet du canal gauche de l'entrée sur le canal gauche de la sortie;

- l'effet du canal droit de l'entrée sur le canal gauche de la sortie;
- l'effet du canal gauche de l'entrée sur le canal droit de la sortie;
- l'effet du canal droit de l'entrée sur le canal droit de la sortie.

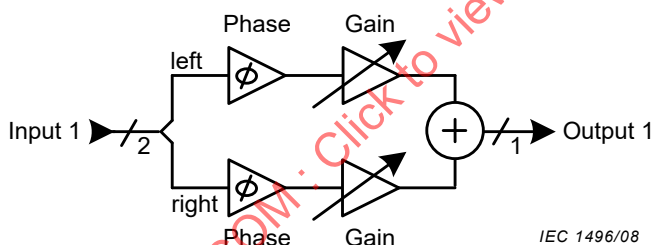
En définissant des valeurs différentes pour la phase et le gain pour chaque paire source/destination, le signal audio peut être manipulé. Par exemple, différentes valeurs de gain peuvent être utilisées pour une permutation mono ou de section du signal audio entrant.

Ce qui suit représente les paramètres de fader (par `aCrosspointPathGain`) pour différents effets; *stereo* transmet l'audio sans altération, *swap* permute les canaux entrants gauche et droite, et *mono* envoie le signal audio mono (M6) sur les deux canaux de sortie.

Src	Dst	Gain		
		stereo	swap	mono
left	left	fullScale	mInfinity	-600
right	left	mInfinity	fullScale	-600
left	right	mInfinity	fullScale	-600
right	right	fullScale	mInfinity	-600

Anglais	Français
stereo	stéréo
swap	permutation
left	gauche
right	droite

Le nombre de canaux d'entrée et de sortie ne doit pas nécessairement être le même, de sorte que ce qui suit modélise un bloc qui utilise une entrée stéréo à deux canaux pour fournir une sortie mono à canal unique:

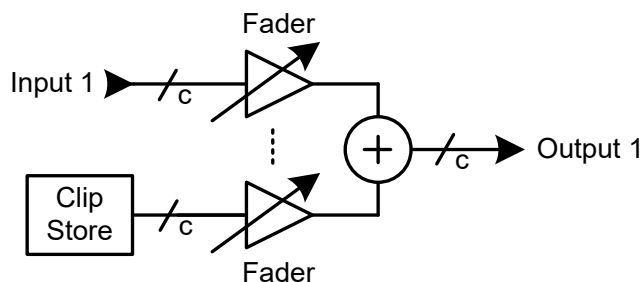


Src	Dest	Gain
left	left	-600
right	left	-600

Anglais	Français
Input 1	Entrée 1
left	gauche
right	droite
Output 1	Sortie 1

5.4.3 Blocs de lecteur de clip audio

Un bloc de lecteur de clip audio doit avoir la structure suivante:



IEC 1497/08

Anglais	Français
Input 1	Entrée 1
Clip Store	Stockage de clips
Output 1	Sortie 1

Figure 4 – Bloc de lecteur de clip audio

où c est le nombre de canaux sur une connexion. Il s'agit d'un bloc qui peut basculer sur une séquence audio préalablement enregistrée choisie ou la mélanger dans un flux audio.

Le groupe d'objets du Tableau 7 doit être mis en œuvre par tout équipement audio conforme qui dispose d'un modèle de gestion qui comprend un ou plusieurs blocs de lecteur de clip audio. Le nœud racine pour ces objets doit être

```
{ iso(1) standard(0) iec62379 audio(2) audioMIB(1) audioClipPlayer(4) }
```

Ce nœud doit être utilisé comme identificateur de type de bloc pour les blocs de lecteur de clip audio.

Tableau 7 – Objets gérés pour blocs de lecteur de clip audio

Identificateur	Syntaxe	Indice	Lisible	Inscriptible	Volatile	Statut
aClipPlayerBlockTable(1)	SEQUENCE OF AClipPlayerBlockEntry		aucun	aucun	non	o
lAcClipPlayerBlockEntry(1)	AClipPlayerBlockEntry		aucun	aucun	non	o
lAcPBlockId(1)	BlockId	oui	aucun	aucun	non	o
lAcPSelection(2)	AudioClipSelector		auditeur	opérateur	non	f
lAcPMainFaderLevelNow(3)	AudioLevel		auditeur	opérateur	non	f
lAcPMainFaderLevelStart(4)	AudioLevel		auditeur	opérateur	non	f
lAcPMainFaderLevelStop(5)	AudioLevel		auditeur	opérateur	non	f
lAcPMainFaderDuration(6)	CardinalNumber		auditeur	opérateur	non	f
lAcPClipFaderLevelNow(7)	AudioLevel		auditeur	opérateur	non	f
lAcPClipFaderLevelStart(8)	AudioLevel		auditeur	opérateur	non	f
lAcPClipFaderLevelStop(9)	AudioLevel		auditeur	opérateur	non	f
lAcPClipFaderDuration(10)	CardinalNumber		auditeur	opérateur	non	f
lAcPStartDelay(11)	IntegerNumber		auditeur	opérateur	non	f
lAcPActivation(12)	AudioClipActivation		auditeur	opérateur	oui	o
lAcPTrackControl(13)	AudioClipTrackControl		aucun	opérateur	oui	f
lAcPTrackNumber(14)	CardinalNumber		auditeur	opérateur	oui	f
lAcPRepeat(15)	AudioClipRepeat		auditeur	opérateur	oui	f

Identificateur	Syntaxe	Indice	Lisible	Inscriptible	Volatile	Statut
aClipSequenceTable(2)	SEQUENCE OF AClipSequenceEntry		aucun	aucun	non	f
LaClipSequenceEntry(1)	AClipSequenceEntry		aucun	aucun	non	o
LaClipSequenceBlockId(1)	BlockId	oui	aucun	aucun	non	o
LaClipSequenceId(2)	AudioClipSelector	oui	aucun	aucun	non	o
LaClipSequenceName(3)	Utf8String		auditeur	opérateur	non	o

5.4.3.1 aClipPlayerBlockTable

Tableau des descripteurs de bloc de lecteur de clip audio pour cette unité. Chaque bloc de lecteur de clip audio de l'unité possède une entrée correspondante dans ce tableau.

5.4.3.2 aClipPlayerBlockEntry

Entrée dans le tableau des blocs de lecteur de clip audio.

5.4.3.3 acpBlockId

Identificateur de bloc pour ce bloc. Utilisé comme indice lors de l'accès au tableau des blocs de lecteur de clip audio.

5.4.3.4 acpSelection

Séquence préalablement enregistrée à lire par ce bloc.

5.4.3.5 acpMainFaderLevelNow

Niveau de fader du chemin de signal principal. Lorsqu'un niveau est défini, le fader bascule immédiatement sur le niveau spécifié, pendant la durée acpMainFaderDuration.

5.4.3.6 acpMainFaderLevelStart

Niveau de fader du chemin de signal principal au démarrage du lecteur de clip. Le fader adopte automatiquement cette valeur pendant la durée acpMainFaderDuration.

5.4.3.7 acpMainFaderLevelStop

Niveau de fader du chemin de signal principal à l'arrêt du lecteur de clip. Le fader adopte automatiquement cette valeur pendant la durée acpMainFaderDuration.

5.4.3.8 acpMainFaderDuration

Temps de transition (en millisecondes) pour une opération de fondu lent effectuée par le fader principal. Une valeur de zéro indique un basculement brusque.

5.4.3.9 acpClipFaderLevelNow

Niveau de fader de la séquence préalablement enregistrée. Lorsqu'un niveau est défini, le fader bascule immédiatement sur le niveau spécifié, pendant la durée acpClipFaderDuration.

5.4.3.10 acpClipFaderLevelStart

Niveau de fader de la séquence préalablement enregistrée au démarrage du lecteur de clip. Le fader adopte automatiquement cette valeur pendant la durée acpClipFaderDuration.

5.4.3.11 **acpClipFaderLevelStop**

Niveau de fader de la séquence préalablement enregistrée à l'arrêt du lecteur de clip. Le fader adopte automatiquement cette valeur pendant la durée `acpClipFaderDuration`.

5.4.3.12 **acpClipFaderDuration**

Temps de transition (en millisecondes) pour une opération de fondu lent effectuée par le fader de clip. Une valeur de zéro indique un basculement brusque.

5.4.3.13 **acpStartDelay**

Décalage temporel (en millisecondes) entre le lecteur de clip en cours d'activation et la séquence préalablement enregistrée en cours de lecture. L'équipement qui prend en charge les opérations programmées peut admettre des valeurs négatives.

5.4.3.14 **acpActivation**

Lorsque la valeur est définie sur `play`, le lecteur de clip commence à lire la séquence préalablement enregistrée choisie. La valeur est automatiquement réinitialisée sur `stop` lorsque la lecture est terminée. L'équipement peut admettre un paramétrage manuel sur `stop` pour abandonner la lecture immédiatement, ou sur `stopAtEnd` pour abandonner la lecture lorsque la piste en cours se termine. Si l'état actuel est `stopAtEnd`, il convient qu'un paramétrage sur `stop` annule la demande d'arrêt en fin de piste et arrête immédiatement la lecture.

5.4.3.15 **acpTrackControl**

Lorsque la valeur est définie sur `next`, le lecteur de clip passe au début de la piste suivante de la séquence choisie. Si le lecteur de clip lit la dernière piste de la séquence choisie, il peut soit passer à la première piste de la séquence, soit rejeter l'opération `SET`.

Lorsque la valeur est définie sur `previous`, le lecteur de clip passe au début de la piste précédente de la séquence choisie. Si le lecteur de clip lit la première piste de la séquence choisie, il peut soit passer à la dernière piste de la séquence, soit rejeter l'opération `SET`.

5.4.3.16 **acpTrackNumber**

Numéro de la piste en cours de lecture. Si le clip choisi n'est pas divisé en pistes, la valeur 1 doit être renvoyée. Si le lecteur de clip est inactif, la valeur 0 doit être renvoyée.

Lorsque la valeur est définie sur un numéro de piste valide pour la séquence choisie, le lecteur de clip passe au début de cette piste. Si la valeur est définie sur un numéro de piste non valide, l'opération `SET` doit être rejetée.

5.4.3.17 **acpTrackRepeat**

Lorsque la valeur est définie sur `none`, le lecteur de clip doit automatiquement arrêter la lecture à la fin de la séquence choisie. Lorsque la valeur est définie sur `all`, le lecteur de clip doit automatiquement revenir au début de la séquence à la fin de la séquence choisie. Lorsque la valeur est définie sur `track`, le lecteur de clip doit automatiquement revenir au début de la piste à la fin d'une piste de la séquence choisie.

5.4.3.18 **aClipSequenceTable**

Tableau des descripteurs de séquence de clip audio pour cette unité. Chaque séquence de clip audio disponible pour chaque bloc de lecteur de clip audio de l'unité possède une entrée correspondante dans ce tableau.

5.4.3.19 aClipSequenceEntry

Entrée dans le tableau des séquences de clip audio.

5.4.3.20 aClipSequenceBlockId

Identificateur de bloc du bloc de lecteur de clip audio associé. Utilisé comme indice lors de l'accès au tableau des séquences de clip audio.

5.4.3.21 aClipSequenceld

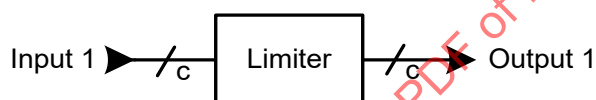
Identificateur de la séquence de clip. Utilisé comme indice lors de l'accès au tableau des séquences de clip audio.

5.4.3.22 aClipSequenceName

Nom attribué à cette séquence de clip. Il s'agit d'une chaîne de texte arbitraire.

5.4.4 Blocs de limiteur audio

Un bloc de limiteur audio doit avoir la structure suivante:



IEC 1498/08

Anglais	Français
Input 1	Entrée 1
Limiter	Limiteur
Output 1	Sortie 1

Figure 5 – Bloc de limiteur audio

où c est le nombre de canaux sur une connexion. Il s'agit d'un bloc qui peut limiter le signal audio entrant à un niveau maximal prédéfini.

Le groupe d'objets du Tableau 8 doit être mis en œuvre par tout équipement audio conforme qui dispose d'un modèle de gestion qui comprend un ou plusieurs blocs de limiteur audio. Le nœud racine pour ces objets doit être

```
{ iso(1) standard(0) iec62379 audio(2) audioMIB(1) audioLimiter(5) }
```

Ce nœud doit être utilisé comme identificateur de type de bloc pour les blocs de limiteur audio.

Tableau 8 – Objets gérés pour blocs de limiteur audio

Identificateur	Syntaxe	Indice	Lisible	Inscriptible	Volatile	Statut
aLimiterBlockTable(1)	SEQUENCE OF ALimiterBlockEntry		aucun	aucun	non	o
LaLimiterBlockEntry(1)	ALimiterBlockEntry		aucun	aucun	non	o
LaLimiterBlockId(1)	BlockId	oui	aucun	aucun	non	o
LaLimiterThreshold(2)	AudioLevel		auditeur	superviseur	non	o
LaLimiterAttackTime(3)	CardinalNumber		auditeur	superviseur	non	f
LaLimiterGainMakeup(4)	AudioLevel		auditeur	superviseur	non	f
LaLimiterRecoveryTime(5)	CardinalNumber		auditeur	superviseur	non	f
LaLimiterRecoveryMode(6)	AudioRecoveryMode		auditeur	superviseur	non	f

5.4.4.1 aLimiterBlockTable

Tableau des descripteurs de bloc de limiteur audio pour cette unité. Chaque bloc de limiteur audio de l'unité possède une entrée correspondante dans ce tableau.

5.4.4.2 aLimiterBlockEntry

Entrée dans le tableau des blocs de limiteur audio.

5.4.4.3 aLimiterBlockId

Identificateur de bloc pour ce bloc. Utilisé comme indice lors de l'accès au tableau des blocs de limiteur audio.

5.4.4.4 aLimiterThreshold

Seuil de limiteur pour ce bloc. Il s'agit du niveau auquel le limiteur commence à atténuer le signal d'entrée.

5.4.4.5 aLimiterAttackTime

Temps d'attaque (en millisecondes) pour ce bloc. Il s'agit du temps nécessaire au limiteur pour commencer l'atténuation lorsque le signal d'entrée dépasse le seuil de limiteur.

5.4.4.6 aLimiterGainMakeup

Compensation de gain pour ce bloc. Il s'agit du volume de gain appliqué au signal après application de la limitation.

5.4.4.7 aLimiterRecoveryTime

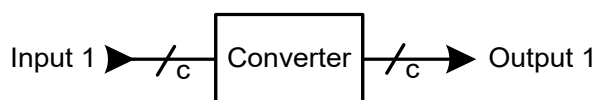
Temps de récupération (en millisecondes) pour ce bloc. Il s'agit du temps nécessaire au limiteur pour arrêter l'atténuation lorsque le signal d'entrée chute au-dessous du seuil de limiteur.

5.4.4.8 aLimiterRecoveryMode

Mode de récupération pour ce bloc. Il s'agit de la façon dont le limiteur effectue une récupération après avoir été activé. Trois descriptions sont actuellement utilisées: *auto*, *slow* and *fast*; toutefois, leur utilisation dépend de la mise en œuvre dans l'équipement.

5.4.5 Blocs de convertisseur audio

Un bloc de convertisseur audio doit avoir la structure suivante:



IEC 1499/08

Anglais	Français
Input 1	Entrée 1
Convertir	Convertisseur
Output 1	Sortie 1

Figure 6 – Bloc de convertisseur audio

où *c* est le nombre de canaux sur une connexion. Il s'agit d'un bloc qui convertit un signal audio entrant dans un format audio donné en un signal audio sortant dans un format audio différent.

NOTE Ce bloc peut être utilisé pour tout type de conversion, y compris le codage et le décodage des formats compressés.

Le tableau des modes du bloc doit être utilisé pour déterminer le format de sortie qu'il convient que le convertisseur génère; si un seul mode est activé, le bloc de convertisseur est alors forcé d'effectuer cette conversion, s'il en est capable. Si plusieurs modes sont activés, il convient que le bloc choisisse le format de sortie en fonction de ses propres règles de mise en œuvre. Si le bloc ne prend en charge aucun des formats de sortie activés, il doit définir `aConverterError` sur `true`.

Le groupe d'objets du Tableau 9 doit être mis en œuvre par tout équipements audio conforme qui dispose d'un modèle de gestion qui comprend un ou plusieurs blocs de convertisseur audio. Le nœud racine pour ces objets doit être

```
{ iso(1) standard(0) iec62379 audio(2) audioMIB(1) audioConverter(6) }
```

Ce nœud doit être utilisé comme identificateur de type de bloc pour les blocs de convertisseur audio.

Tableau 9 – Objets gérés pour blocs de convertisseur audio

Identificateur	Syntaxe	Indice	Lisible	Inscriptible	Volatile	Statut
<code>aConverterBlockTable(1)</code>	SEQUENCE OF <code>AConverterBlockEntry</code>		aucun	aucun	non	o
<code>laConverterBlockEntry(1)</code>	<code>AConverterBlockEntry</code>		aucun	aucun	non	o
<code>laConverterBlockId(1)</code>	<code>BlockId</code>	oui	aucun	aucun	non	o
<code>laConverterQuality(2)</code>	<code>AudioQuality</code>		auditeur	superviseur	non	f
<code>laConverterEnabled(3)</code>	<code>TruthValue</code>		auditeur	superviseur	non	f
<code>laConverterDithering(4)</code>	<code>TruthValue</code>		auditeur	superviseur	non	f
<code>laConverterOutputFormat(5)</code>	<code>MediaFormat</code>		auditeur	aucun	non	o
<code>laConverterError(6)</code>	<code>TruthValue</code>		auditeur	aucun	non	f

5.4.5.1 `aConverterBlockTable`

Tableau des descripteurs de bloc de convertisseur audio pour cette unité. Chaque bloc de convertisseur audio de l'unité possède une entrée correspondante dans ce tableau.

5.4.5.2 `aConverterBlockEntry`

Entrée dans le tableau des blocs de convertisseur audio.

5.4.5.3 aConverterBlockId

Identificateur de bloc pour ce bloc. Utilisé comme indice lors de l'accès au tableau des blocs de convertisseur audio.

5.4.5.4 aConverterQuality

Qualité de la conversion effectuée par ce bloc.

5.4.5.5 aConverterEnabled

La valeur `true` indique que le signal audio entrant peut être converti dans l'un des formats audio activés pour la sortie du bloc dans le tableau des modes. La valeur `false` indique que le signal audio entrant doit être généré en sortie dans le même format qu'en entrée.

5.4.5.6 aConverterDithering

La valeur `true` indique qu'un dithering est appliqué au signal audio converti. La valeur `false` indique qu'aucun dithering n'est appliqué au signal audio converti.

5.4.5.7 aConverterOutputFormat

Description du format de signal audio actuel généré en sortie par le bloc de convertisseur.

5.4.5.8 aConverterError

La valeur `true` indique que le signal audio ne peut pas être converti dans les formats de sortie activés dans le tableau des modes.

5.4.6 Blocs d'alarme de niveau audio

Un bloc d'alarme de niveau audio doit avoir la structure suivante:

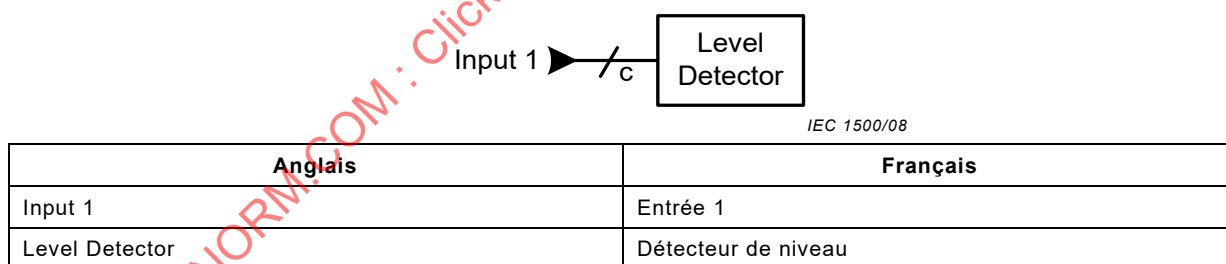


Figure 7 – Bloc d'alarme de niveau audio

où *c* est le nombre de canaux sur une connexion. Il s'agit d'un bloc qui détecte les conditions de défaut du niveau audio dans un flux audio.

NOTE Un bloc d'alarme de niveau audio peut être utilisé pour représenter, par exemple, un détecteur de pertes audio ou un indicateur de surcharge.

Le groupe d'objets du Tableau 10 doit être mis en œuvre par tout équipement audio conforme qui dispose d'un modèle de gestion qui comprend un ou plusieurs blocs d'alarme de niveau audio. Le nœud racine pour ces objets doit être

```
{ iso(1) standard(0) iec62379 audio(2) audioMIB(1) audioLevelAlarm(7) }
```

Ce nœud doit être utilisé comme identificateur de type de bloc pour les blocs d'alarme de niveau audio.

Tableau 10 – Objets gérés pour blocs d'alarme de niveau audio

Identificateur	Syntaxe	Indice	Lisible	Inscriptible	Volatile	Statut
aLevelAlarmBlockTable(1)	SEQUENCE OF ALevelAlarmBlockEntry		aucun	aucun	non	o
lAlevelAlarmBlockEntry(1)	ALevelAlarmBlockEntry		aucun	aucun	non	o
lalaBlockId(1)	BlockId	oui	aucun	aucun	non	o
lalaType(2)	AudioLevelAlarmType		auditeur	superviseur	non	o
lalaThreshold(3)	AudioLevel		auditeur	superviseur	non	o
lalaWarningTime(4)	CardinalNumber		auditeur	superviseur	non	f
lalaFailureTime(5)	CardinalNumber		auditeur	superviseur	non	o
lalaCounter(6)	CardinalNumber		auditeur	superviseur	non	o
lalaEnabled(7)	TruthValue		auditeur	superviseur	non	o
lalaStatus(8)	AudioAlarmStatus		auditeur	aucun	oui	o

5.4.6.1 aLevelAlarmBlockTable

Tableau des descripteurs de bloc d'alarme de niveau audio pour cette unité. Chaque bloc d'alarme de niveau audio de l'unité possède une entrée correspondante dans ce tableau.

5.4.6.2 aLevelAlarmBlockEntry

Entrée dans le tableau des blocs d'alarme de niveau audio.

5.4.6.3 alaBlockId

Identificateur de bloc pour ce bloc. Utilisé comme indice lors de l'accès au tableau des blocs d'alarme de niveau audio.

5.4.6.4 alaType

Type d'alarme de niveau audio. Si la valeur est `lower`, l'alarme est activée lorsque le niveau audio est inférieur au seuil; si la valeur est `higher`, l'alarme est activée lorsque le niveau est supérieur au seuil.

5.4.6.5 alaThreshold

Seuil de détection pour le déclenchement d'une alarme de niveau audio par ce bloc.

5.4.6.6 alaWarningTime

Temps de détection (en secondes) pour le déclenchement d'une alarme d'avertissement de niveau audio par ce bloc; voir 5.4.6.10.

5.4.6.7 alaFailureTime

Temps de détection (en secondes) pour le déclenchement d'une alarme de défaillance de niveau audio par ce bloc; voir 5.4.6.10.

5.4.6.8 alaCounter

Durée (en secondes) pendant laquelle le niveau audio a dépassé (c'est-à-dire au-dessus ou au-dessous, selon `alaType`) le seuil de détection.

Le compteur doit être maintenu à zéro tant que le niveau audio ne dépasse pas le seuil de détection.

Le compteur peut être défini par l'entité de gestion; si, au moment de la demande `SET`, le signal audio dépasse pas le seuil de détection, le compteur doit continuer à partir de la valeur définie.

5.4.6.9 `alaEnabled`

Détermine si l'alarme est amorcée pour fonctionner. Si la valeur est définie sur `true`, les alarmes d'avertissement et de défaillance se déclenchent lorsque le compteur d'alarme de niveau audio dépasse respectivement les temps d'avertissement et de défaillance. Si la valeur est définie sur `false`, l'alarme est désactivée et aucune alarme ne se déclenche.

5.4.6.10 `alaStatus`

Indique le statut du bloc d'alarme. La valeur doit être `failure` si le niveau audio dépasse le seuil de détection et que `alaCounter` est supérieur ou égal à `alaFailureTime`, `warning` si le niveau audio dépasse le seuil de détection et que `alaCounter` est supérieur ou égal à `alaWarningTime`, `ok` dans les autres cas.

6 Diffusions d'état

6.1 Généralités

NOTE Les diffusions d'état sont spécifiées dans l'IEC 62379-1. La présente norme spécifie les pages et les groupes d'état en lien avec les signaux audio.

6.2 Définitions de types

En plus des types définis dans l'IEC 62379-1, les types suivants sont utilisés pour spécifier la syntaxe des structures de données abstraites utilisées dans les pages d'état.

6.2.1 Conventions textuelles

```
AudioConverterStatus ::= OCTET STRING (SIZE(1))
-- ensemble de bits qui représente l'état d'un convertisseur audio
-- bit 0 (lsb) = convertisseur activé (1 = activé, 0 = désactivé)
-- bit 1 = dithering appliqué (1 = avec dithering, 0 = sans
-- dithering)
-- bit 2 = conversion ok (1 = entrée convertie, 0 = erreur de
-- convertisseur)
-- bit 3 = réservé
-- bit 4 = réservé
-- bit 5 = réservé
-- bit 6 = réservé
-- bit 7 (msb) = réservé
```

6.2.2 Séquences

```
AudioFormatsMapEntry ::= SEQUENCE {
    afmNumber IndexNumber,
    afmFormat MediaFormat
}
```

6.3 Mappage de formats audio

NOTE 1 En raison de la longueur variable des identificateurs d'objet, fournir une indication d'un format audio particulier dans une page d'état est problématique. Par conséquent, un tableau qui attribue un identificateur arbitraire à un format audio est défini pour permettre d'associer un code plus court, de longueur fixe, à un format. Le mappage choisi est spécifique à l'équipement et il convient que le logiciel de gestion interroge le tableau avec des demandes SNMP GET-NEXT au démarrage pour apprendre la carte d'un dispositif.

Le groupe d'objets du Tableau 11 doit être mis en œuvre par tous les équipements audio conformes qui rapportent des informations sur le format audio dans les pages d'état. Le nœud racine pour ces objets doit être

{ iso(1) standard(0) iec62379 audio(2) audioStatusMIB(4) }

Tableau 11 – Objets gérés pour mappages de formats audio

Identificateur	Syntaxe	Indice	Lisible	Inscriptible	Volatile	Statut
audioFormatsMapTable(1)	SEQUENCE OF AudioFormatsMapEntry		aucun	aucun	non	o
↳ audioFormatsMapEntry(1)	AudioFormatsMapEntry		aucun	aucun	non	o
↳ afmNumber(1)	IndexNumber	oui	aucun	aucun	non	o
↳ afmFormat(2)	MediaFormat		auditeur	superviseur	oui	o

6.3.1 audioFormatsMapTable

Tableau des formats audio pour les mappages des numéros d'indice pour cette unité. Chaque mappage possède une entrée correspondante dans ce tableau.

6.3.2 audioFormatsMapEntry

Entrée dans le tableau des mappages de formats audio.

6.3.3 afmNumber

Numéro de format audio pour ce mappage.

6.3.4 afmFormat

Format des données audio associées à ce numéro de format.

6.4 Formats de page

6.4.1 Page d'accès audio

Cette page doit être produite par tous les accès audio et doit comporter les entrées du Tableau 12:

Tableau 12 – Entrées d'état pour la page d'accès audio

Octets	Description	Valeur	Note(s)
1..2	Numéro de page		
3..4	Identificateur de bloc	BlockId	1
5..8	Format audio	IndexNumber	2
9..10	Niveau de crête du canal 1	AudioLevel	3
...	répéter l'entrée précédente pour chaque canal restant	...	4
NOTE 1 Codé avec l'identificateur de bloc du bloc.			
NOTE 2 Codé avec le mappage de audioFormatsMapTable pour le format audio de l'accès, indiqué par aPortFormat.			
NOTE 3 Codé avec le niveau de crête audio détecté sur le canal spécifié de l'accès spécifié depuis la dernière diffusion de la page, en unités de 0,01 dB. Une valeur de 0 représente la pleine échelle et les valeurs de 1 ou plus indiquent une surcharge.			
NOTE 4 Le nombre de canaux audio doit être déduit de la longueur de la page.			

6.4.2 Page de données auxiliaires AES3

Cette page doit être produite par tous les accès audio qui permettent l'accès aux données auxiliaires AES3 et doit comporter les entrées du Tableau 13:

Tableau 13 – Entrées d'état pour la page de données auxiliaires AES3

Octets	Description	Valeur	Note(s)
1..2	Numéro de page		
3..4	Identificateur de bloc	BlockId	1
5..28	Données de canal du canal 1	Aes3ChannelData	2
29..52	Données utilisateur du canal 1	Aes3UserData	3
53	Erreur de validité du canal 1	TruthValue	4
...	répéter les 3 entrées précédentes pour chaque canal restant	...	5
NOTE 1 Codé avec l'identificateur de bloc du bloc.			
NOTE 2 Codé avec la valeur actuelle de <code>aes3ChannelData</code> dans l'entrée de tableau associée à ce canal.			
NOTE 3 Codé avec la valeur actuelle de <code>aes3UserData</code> dans l'entrée de tableau associée à ce canal.			
NOTE 4 Codé avec la valeur actuelle de <code>aes3ValidityError</code> dans l'entrée de tableau associée à ce canal.			
NOTE 5 Le nombre de canaux audio doit être déduit de la longueur de la page.			

6.4.3 Page de mélangeur audio

Cette page doit être produite par tous les blocs de mélangeur audio et doit comporter les entrées du Tableau 14:

Tableau 14 – Entrées d'état pour la page de mélangeur audio

Octets	Description	Valeur	Note(s)
1..2	Numéro de page		
3..4	Identificateur de bloc	BlockId	1
5..8	Numéro d'entrée	IndexNumber	2
9..12	Latence	CardinalNumber	3
13..14	Niveau	AudioLevel	4
...	répéter les trois entrées précédentes pour chaque entrée	...	5
NOTE 1 Codé avec l'identificateur de bloc du bloc.			
NOTE 2 Codé avec le numéro de l'entrée actuelle indiquée par <code>aMixerInputNumber</code> dans l'entrée de tableau associée à cette entrée de mélangeur.			
NOTE 3 Codé avec la valeur actuelle de <code>aMixerInputDelay</code> dans l'entrée de tableau associée à cette entrée de mélangeur.			
NOTE 4 Codé avec la valeur actuelle de <code>aMixerInputLevel</code> dans l'entrée de tableau associée à cette entrée de mélangeur.			
NOTE 5 Le nombre d'entrées de mélangeur doit être déduit de la longueur de la page.			

6.4.4 Page de point de connexion audio

Cette page doit être produite par tous les blocs de point de connexion audio et doit comporter les entrées du Tableau 15:

Tableau 15 – Entrées d'état pour la page de point de connexion audio

Octets	Description	Valeur	Note(s)
1..2	Numéro de page		
3..4	Identificateur de bloc	BlockId	1
5..6	Source du chemin de point de connexion	AudioChannel	2
7..8	Destination du chemin de point de connexion	AudioChannel	3
9..10	Phase du chemin de point de connexion	AudioPhase	4
11..12	Gain du chemin de point de connexion	AudioLevel	5
...	répéter les quatre entrées précédentes pour chaque chemin	...	6
NOTE 1 Codé avec l'identificateur de bloc du bloc.			
NOTE 2 Codé avec la source du chemin actuel indiqué par aCrosspointPathSrc dans l'entrée de tableau associée à ce chemin de point de connexion			
NOTE 3 Codé avec la destination du chemin actuel indiqué par aCrosspointPathDst dans l'entrée de tableau associée à ce chemin de point de connexion.			
NOTE 4 Codé avec la valeur actuelle de aCrosspointPathPhase dans l'entrée de tableau associée à ce chemin de point de connexion.			
NOTE 5 Codé avec la valeur actuelle de aCrosspointPathGain dans l'entrée de tableau associée à ce chemin de point de connexion.			
NOTE 6 Le nombre de chemins par le point de connexion doit être déduit de la longueur de la page.			

6.4.5 Page de lecteur de clip audio

Cette page doit être produite par tous les blocs de lecteur de clip audio et doit comporter les entrées du Tableau 16:

Tableau 16 – Entrées d'état pour la page de lecteur de clip audio

Octets	Description	Valeur	Note(s)
1..2	Numéro de page		
3..4	Identificateur de bloc	BlockId	1
5	Activation du lecteur de clip	AudioClipActivation	2
6..9	Choix du lecteur de clip	AudioClipSelector	3
10..13	Numéro de piste du lecteur de clip	CardinalNumber	4
14..15	Niveau du fader principal	AudioLevel	5
16..17	Niveau du fader de lecteur de clip	AudioLevel	6
NOTE 1 Codé avec l'identificateur de bloc du bloc.			
NOTE 2 Codé avec la valeur actuelle de acpActivation dans l'entrée de tableau associée à ce lecteur de clip.			
NOTE 3 Codé avec la valeur actuelle de acpSelection dans l'entrée de tableau associée à ce lecteur de clip.			
NOTE 4 Codé avec la valeur actuelle de acpTrackNumber dans l'entrée de tableau associée à ce lecteur de clip.			
NOTE 5 Codé avec la valeur actuelle de acpMainFaderLevelNow dans l'entrée de tableau associée à ce lecteur de clip.			
NOTE 6 Codé avec la valeur actuelle de acpClipFaderLevelNow dans l'entrée de tableau associée à ce lecteur de clip.			

6.4.6 Page de limiteur audio

Cette page doit être produite par tous les blocs de limiteur audio et doit comporter les entrées du Tableau 17:

Tableau 17 – Entrées d'état pour la page de limiteur audio

Octets	Description	Valeur	Note(s)
1..2	Numéro de page		
3..4	Identificateur de bloc	BlockId	1
5..6	Seuil de limiteur	AudioLevel	2
7..10	Temps d'attaque de limiteur	CardinalNumber	3
11..12	Compensation de gain de limiteur	AudioLevel	4
13..16	Temps de récupération de limiteur	CardinalNumber	5
17	Mode de récupération de limiteur	AudioRecoveryMode	6
NOTE 1 Codé avec l'identificateur de bloc du bloc. NOTE 2 Codé avec la valeur actuelle de <code>aLimiterThreshold</code> dans l'entrée de tableau associée à ce limiteur. NOTE 3 Codé avec la valeur actuelle de <code>aLimiterAttackTime</code> dans l'entrée de tableau associée à ce limiteur. NOTE 4 Codé avec la valeur actuelle de <code>aLimiterGainMakeup</code> dans l'entrée de tableau associée à ce limiteur. NOTE 5 Codé avec la valeur actuelle de <code>aLimiterRecoveryTime</code> dans l'entrée de tableau associée à ce limiteur. NOTE 6 Codé avec la valeur actuelle de <code>aLimiterRecoveryMode</code> dans l'entrée de tableau associée à ce limiteur.			

6.4.7 Page de convertisseur audio

Cette page doit être produite par tous les blocs de convertisseur audio et doit comporter les entrées du Tableau 18:

Tableau 18 – Entrées d'état pour la page de convertisseur audio

Octets	Description	Valeur	Note(s)
1..2	Numéro de page		
3..4	Identificateur de bloc	BlockId	1
5	Etat du convertisseur	AudioConverterStatus	2
6..9	Format de sortie	IndexNumber	3
NOTE 1 Codé avec l'identificateur de bloc du bloc. NOTE 2 Codé avec la valeur actuelle de l'état du convertisseur. NOTE 3 Codé avec le mappage de <code>audioFormatsMapTable</code> pour le format de sortie audio du convertisseur, indiqué par <code>aConverterOutputFormat</code>.			

6.4.8 Page d'alarme de niveau audio

Cette page doit être produite par tous les blocs d'alarme de niveau audio et doit comporter les entrées du Tableau 19:

Tableau 19 – Entrées d'état pour la page d'alarme de niveau audio

Octets	Description	Valeur	Note(s)
1..2	Numéro de page		
3..4	Identificateur de bloc	BlockId	1
5	Alarme activée	TruthValue	2
6	Etat de l'alarme	AudioAlarmStatus	3
7..10	Compteur	CardinalNumber	4
11..12	Seuil audio	AudioLevel	5
13..16	Heure de l'avertissement	CardinalNumber	6
17..20	Temps de défaillance	CardinalNumber	7
NOTE 1 Codé avec l'identificateur de bloc du bloc.			
NOTE 2 Codé avec la valeur actuelle de <code>alaEnabled</code> dans l'entrée de tableau associée à l'alarme de niveau audio.			
NOTE 3 Codé avec la valeur actuelle de <code>alaStatus</code> dans l'entrée de tableau associée à l'alarme de niveau audio.			
NOTE 4 Codé avec la valeur actuelle de <code>alaCounter</code> dans l'entrée de tableau associée à l'alarme de niveau audio.			
NOTE 5 Codé avec la valeur actuelle de <code>alaThreshold</code> dans l'entrée de tableau associée à l'alarme de niveau audio.			
NOTE 6 Codé avec la valeur actuelle de <code>alaWarningTime</code> dans l'entrée de tableau associée à l'alarme de niveau audio.			
NOTE 7 Codé avec la valeur actuelle de <code>alaFailureTime</code> dans l'entrée de tableau associée à l'alarme de niveau audio.			

6.5 Groupes de pages

6.5.1 `audioPorts`

Ce groupe doit être pris en charge par toutes les unités audio avec des accès. Le nœud racine de ce groupe doit être

```
{ iso(1) standard(0) iec62379 audio(2) audioStatusGroup(3) audioPorts(1) }
```

La diffusion de ce groupe doit comprendre les éléments suivants:

- page 1: accès audio, diffusion à la vitesse de page de base;
- page 2: données auxiliaires AES3, diffusion à la moitié de la vitesse de page de base.

La vitesse de page de base par défaut pour ce groupe doit être de 60 pages par minute (1 page par seconde).

6.5.2 `standardAudioBlocks`

Ce groupe doit être pris en charge par toutes les unités audio avec des blocs définis dans la présente norme. Le nœud racine de ce groupe doit être

```
{ iso(1) standard(0) iec62379 audio(2) audioStatusGroup(3) standardAudioBlocks(2) }
```

La diffusion de ce groupe doit comprendre les éléments suivants:

- page 1: mélangeur audio, diffusion à la vitesse de page de base;
- page 2: point de connexion audio, diffusion à la vitesse de page de base;
- page 3: lecteur de clip audio, diffusion à la vitesse de page de base;
- page 4: limiteur audio, diffusion à la vitesse de page de base;
- page 5: convertisseur audio, diffusion à la vitesse de page de base.

La vitesse de page de base par défaut pour ce groupe doit être de 60 pages par minute (1 page par seconde).

6.5.3 audioAlarms

Ce groupe doit être pris en charge par toutes les unités audio avec des alarmes. Le nœud racine de ce groupe doit être

{ iso(1) standard(0) iec62379 audio(2) audioStatusGroup(3) audioAlarms (3) }

La diffusion de ce groupe doit comprendre les éléments suivants:

- page 1: alarme de niveau audio, diffusion à la vitesse de page de base.

La vitesse de page de base par défaut pour ce groupe doit être de 60 pages par minute (1 page par seconde).

IECNORM.COM : Click to view the full PDF of IEC 62379-2:2008

Annexe A (informative)

Définitions de formats audio lisibles par une machine

La présente annexe fournit une version lisible par une machine des définitions de formats de données audio, destinée à être interprétable par des outils logiciels d'exploration de la MIB normalisés. En cas d'incohérence entre la présente annexe et l'Article 4, l'Article 4 prévaut.

Le format utilisé pour décrire les identificateurs de format de données est conforme à l'IETF STD 58 (SMIv2).

NOTE 1 La présente annexe ne vise pas à couvrir tous les formats admis par les définitions de l'Article 4.

```
IEC62379-2-FORMATS DEFINITIONS ::= BEGIN

IMPORTS

    MODULE-IDENTITY, OBJECT-IDENTITY
        FROM SNMPv2-SMI

    iec62379
        FROM IEC62379-1-MIB;

audioFormat MODULE-IDENTITY
    LAST-UPDATED "200804301500Z"
    ORGANIZATION "IEC PT62379"
    CONTACT-INFO "Non spécifié"
    DESCRIPTION "Identificateurs de format audio définis à l'Article 4 de
        l'IEC 62379-2."
    REVISION "200804301500Z"
    DESCRIPTION "Edition 1.0, version finale".
    ::= { audio 2 }

audio OBJECT IDENTIFIER ::= { iec62379 2 }

-- 4.1 Définitions de formats de signal audio
-- 4.1.1 Paramètres audio

ChannelArrangement ::= TEXTUAL-CONVENTION
    STATUS current
    DESCRIPTION "Configuration de canaux audio au format audio."
    SYNTAX INTEGER (0..5)
    -- {
    --     unspecified          (0),
    --     discreteMono         (1),
    --     stereo                (2),
    --     jointStereo           (3),
    --     surround              (4),
    --     surroundWithDownmix   (5)
    -- } (unspecified..surroundWithDownmix)

NumberChannels ::= TEXTUAL-CONVENTION
    STATUS current
    DESCRIPTION "Entier qui représente le nombre de canaux audio."
    SYNTAX INTEGER (0..2147483647)

BitDepth ::= TEXTUAL-CONVENTION
    STATUS current
    DESCRIPTION "Entier qui représente la profondeur binaire audio en bits par
        échantillon."
    SYNTAX INTEGER (0..2147483647)
```

```

SamplingFrequency ::= TEXTUAL-CONVENTION
    STATUS          current
    DESCRIPTION     "Entier qui représente la fréquence d'échantillonnage audio en
                    Hz."
    SYNTAX          INTEGER (0..2147483647)

BitRate ::= TEXTUAL-CONVENTION
    STATUS          current
    DESCRIPTION     "Entier qui représente le débit binaire audio en bits par
                    seconde."
    SYNTAX          INTEGER (0..2147483647)

-- 4.1.2 Formats de signal audio

audioSignalFormat OBJECT IDENTIFIER ::= { audioFormat 1 }

-- 4.1.2.1 Signal audio non spécifié

unspecifiedAudio OBJECT-IDENTITY
    STATUS          current
    DESCRIPTION     "Valeur de caractère générique AudioFormat - tout format pris en
                    charge admis."
    ::= { audioSignalFormat 0 }

-- 4.1.2.2 Sans audio

noAudio OBJECT-IDENTITY
    STATUS          current
    DESCRIPTION     "Indique que la sortie est muette."
    ::= { audioSignalFormat 1 }

-- 4.1.2.3 Signal audio analogique

analogueAudio OBJECT-IDENTITY
    STATUS          current
    DESCRIPTION     "Signal audio analogique".
    ::= { audioSignalFormat 2 }

-- analogique mono distinct

analogueMono OBJECT-IDENTITY
    STATUS          current
    DESCRIPTION     "Signal audio analogique mono distinct."
    ::= { analogueAudio 1 }

analogueMono1Chan OBJECT-IDENTITY
    STATUS          current
    DESCRIPTION     "Signal audio analogique mono distinct avec 1 canal."
    ::= { analogueMono 1 }

analogueMono2Chan OBJECT-IDENTITY
    STATUS          current
    DESCRIPTION     "Signal audio analogique mono distinct avec 2 canaux."
    ::= { analogueMono 2 }

analogueMono3Chan OBJECT-IDENTITY
    STATUS          current
    DESCRIPTION     "Signal audio analogique mono distinct avec 3 canaux."
    ::= { analogueMono 3 }

analogueMono4Chan OBJECT-IDENTITY
    STATUS          current
    DESCRIPTION     "Signal audio analogique mono distinct avec 4 canaux."
    ::= { analogueMono 4 }

```

```
-- stéréo analogique

analogueStereo OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "Signal audio analogique stéréo."
  ::= { analogueAudio 2 }

analogueStereo2Chan OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "Signal audio analogique stéréo avec 2 canaux (1 paire L-R)."
  ::= { analogueStereo 2 }

analogueStereo4Chan OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "Signal audio analogique stéréo avec 4 canaux (2 paires L-R)."
  ::= { analogueStereo 4 }

-- analogique stéréo mixte

analogueJointStereo OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "Signal audio analogique stéréo mixte."
  ::= { analogueAudio 3 }

analogueJointStereo2Chan OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "Signal audio analogique stéréo mixte avec 2 canaux (1 paire
               M-S)."
  ::= { analogueJointStereo 2 }

analogueJointStereo4Chan OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "Signal audio analogique stéréo mixte avec 4 canaux (2 paires
               M-S)."
  ::= { analogueJointStereo 4 }

-- 4.1.2.4 Signal audio MIC

pcmAudio OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "Signal audio MIC linéaire."
  ::= { audioSignalFormat 3 }

-- MIC mono distinct

pcmMono OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "Signal audio MIC linéaire mono distinct."
  ::= { pcmAudio 1 }

pcmMono1Chan OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "Signal audio MIC linéaire mono distinct avec 1 canal."
  ::= { pcmMono 1 }

pcmMono1Chan16 OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "Signal audio MIC linéaire mono distinct de 16 bits avec
               1 canal."
  ::= { pcmMono1Chan 16 }

pcmMono1Chan16at44100 OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "Signal audio MIC linéaire mono distinct de 44,1 kHz et 16 bits
               avec 1 canal."
```

```

::= { pcmMono1Chan16 44100 }

pcmMono1Chan16at48000 OBJECT-IDENTITY
    STATUS      current
    DESCRIPTION  "Signal audio MIC linéaire mono distinct de 48 kHz et 16 bits
                  avec 1 canal."
    ::= { pcmMono1Chan16 48000 }

pcmMono1Chan16at96000 OBJECT-IDENTITY
    STATUS      current
    DESCRIPTION  "Signal audio MIC linéaire mono distinct de 96 kHz et 16 bits
                  avec 1 canal."
    ::= { pcmMono1Chan16 96000 }

pcmMono1Chan20 OBJECT-IDENTITY
    STATUS      current
    DESCRIPTION  "Signal audio MIC linéaire mono distinct de 20 bits avec
                  1 canal."
    ::= { pcmMono1Chan 20 }

pcmMono1Chan20at44100 OBJECT-IDENTITY
    STATUS      current
    DESCRIPTION  "Signal audio MIC linéaire mono distinct de 44,1 kHz et 20 bits
                  avec 1 canal."
    ::= { pcmMono1Chan20 44100 }

pcmMono1Chan20at48000 OBJECT-IDENTITY
    STATUS      current
    DESCRIPTION  "Signal audio MIC linéaire mono distinct de 48 kHz et 20 bits
                  avec 1 canal."
    ::= { pcmMono1Chan20 48000 }

pcmMono1Chan20at96000 OBJECT-IDENTITY
    STATUS      current
    DESCRIPTION  "Signal audio MIC linéaire mono distinct de 96 kHz et 20 bits
                  avec 1 canal."
    ::= { pcmMono1Chan20 96000 }

pcmMono1Chan24 OBJECT-IDENTITY
    STATUS      current
    DESCRIPTION  "Signal audio MIC linéaire mono distinct de 24 bits avec
                  1 canal."
    ::= { pcmMono1Chan 24 }

pcmMono1Chan24at44100 OBJECT-IDENTITY
    STATUS      current
    DESCRIPTION  "Signal audio MIC linéaire mono distinct de 44,1 kHz et 24 bits
                  avec 1 canal."
    ::= { pcmMono1Chan24 44100 }

pcmMono1Chan24at48000 OBJECT-IDENTITY
    STATUS      current
    DESCRIPTION  "Signal audio MIC linéaire mono distinct de 48 kHz et 24 bits
                  avec 1 canal."
    ::= { pcmMono1Chan24 48000 }

pcmMono1Chan24at96000 OBJECT-IDENTITY
    STATUS      current
    DESCRIPTION  "Signal audio MIC linéaire mono distinct de 96 kHz et 24 bits
                  avec 1 canal."
    ::= { pcmMono1Chan24 96000 }

pcmMono2Chan OBJECT-IDENTITY
    STATUS      current
    DESCRIPTION  "Signal audio MIC linéaire mono distinct avec 2 canaux."
    ::= { pcmMono 2 }

pcmMono2Chan16 OBJECT-IDENTITY
    STATUS      current

```

```
DESCRIPTION      "Signal audio MIC linéaire mono distinct de 16 bits avec
                  2 canaux."
::= { pcmMono2Chan 16 }

pcmMono2Chan16at44100 OBJECT-IDENTITY
STATUS           current
DESCRIPTION      "Signal audio MIC linéaire mono distinct de 44,1 kHz et 16 bits
                  avec 2 canaux."
::= { pcmMono2Chan16 44100 }

pcmMono2Chan16at48000 OBJECT-IDENTITY
STATUS           current
DESCRIPTION      "Signal audio MIC linéaire mono distinct de 48 kHz et 16 bits
                  avec 2 canaux."
::= { pcmMono2Chan16 48000 }

pcmMono2Chan16at96000 OBJECT-IDENTITY
STATUS           current
DESCRIPTION      "Signal audio MIC linéaire mono distinct de 96 kHz et 16 bits
                  avec 2 canaux."
::= { pcmMono2Chan16 96000 }

pcmMono2Chan20 OBJECT-IDENTITY
STATUS           current
DESCRIPTION      "Signal audio MIC linéaire mono distinct de 20 bits avec
                  2 canaux."
::= { pcmMono2Chan 20 }

pcmMono2Chan20at44100 OBJECT-IDENTITY
STATUS           current
DESCRIPTION      "Signal audio MIC linéaire mono distinct de 44,1 kHz et 20 bits
                  avec 2 canaux."
::= { pcmMono2Chan20 44100 }

pcmMono2Chan20at48000 OBJECT-IDENTITY
STATUS           current
DESCRIPTION      "Signal audio MIC linéaire mono distinct de 48 kHz et 20 bits
                  avec 2 canaux."
::= { pcmMono2Chan20 48000 }

pcmMono2Chan20at96000 OBJECT-IDENTITY
STATUS           current
DESCRIPTION      "Signal audio MIC linéaire mono distinct de 96 kHz et 20 bits
                  avec 2 canaux."
::= { pcmMono2Chan20 96000 }

pcmMono2Chan24 OBJECT-IDENTITY
STATUS           current
DESCRIPTION      "Signal audio MIC linéaire mono distinct de 24 bits avec
                  2 canaux."
::= { pcmMono2Chan 24 }

pcmMono2Chan24at44100 OBJECT-IDENTITY
STATUS           current
DESCRIPTION      "Signal audio MIC linéaire mono distinct de 44,1 kHz et 24 bits
                  avec 2 canaux."
::= { pcmMono2Chan24 44100 }

pcmMono2Chan24at48000 OBJECT-IDENTITY
STATUS           current
DESCRIPTION      "Signal audio MIC linéaire mono distinct de 48 kHz et 24 bits
                  avec 2 canaux."
::= { pcmMono2Chan24 48000 }

pcmMono2Chan24at96000 OBJECT-IDENTITY
STATUS           current
DESCRIPTION      "Signal audio MIC linéaire mono distinct de 96 kHz et 24 bits
                  avec 2 canaux."
::= { pcmMono2Chan24 96000 }
```

-- MIC stéréo

```

pcmStereo OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "Signal audio MIC linéaire stéréo."
  ::= { pcmAudio 2 }

pcmStereo2Chan OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "Signal audio MIC linéaire stéréo avec 2 canaux (1 paire L-R)."
```

::= { pcmStereo 2 }

```

pcmStereo2Chan16 OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "Signal audio MIC linéaire stéréo de 16 bits avec 2 canaux."
  ::= { pcmStereo2Chan 16 }
```

```

pcmStereo2Chan16at44100 OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "Signal audio MIC linéaire stéréo de 44,1 kHz et 16 bits avec
                2 canaux."
  ::= { pcmStereo2Chan16 44100 }
```

```

pcmStereo2Chan16at48000 OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "Signal audio MIC linéaire stéréo de 48 kHz et 16 bits avec
                2 canaux."
  ::= { pcmStereo2Chan16 48000 }
```

```

pcmStereo2Chan16at96000 OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "Signal audio MIC linéaire stéréo de 96 kHz et 16 bits avec
                2 canaux."
  ::= { pcmStereo2Chan16 96000 }
```

```

pcmStereo2Chan20 OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "Signal audio MIC linéaire stéréo de 20 bits avec 2 canaux."
  ::= { pcmStereo2Chan 20 }
```

```

pcmStereo2Chan20at44100 OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "Signal audio MIC linéaire stéréo de 44,1 kHz et 20 bits avec
                2 canaux."
  ::= { pcmStereo2Chan20 44100 }
```

```

pcmStereo2Chan20at48000 OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "Signal audio MIC linéaire stéréo de 48 kHz et 20 bits avec
                2 canaux."
  ::= { pcmStereo2Chan20 48000 }
```

```

pcmStereo2Chan20at96000 OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "Signal audio MIC linéaire stéréo de 96 kHz et 20 bits avec
                2 canaux."
  ::= { pcmStereo2Chan20 96000 }
```

```

pcmStereo2Chan24 OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "Signal audio MIC linéaire stéréo de 24 bits avec 2 canaux."
  ::= { pcmStereo2Chan 24 }
```

```

pcmStereo2Chan24at44100 OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "Signal audio MIC linéaire stéréo de 44,1 kHz et 24 bits avec
                2 canaux."
  ::= { pcmStereo2Chan24 44100 }
```

```
pcmStereo2Chan24at48000 OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "Signal audio MIC linéaire stéréo de 48 kHz et 24 bits avec
                2 canaux."
  ::= { pcmStereo2Chan24 48000 }

pcmStereo2Chan24at96000 OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "Signal audio MIC linéaire stéréo de 96 kHz et 24 bits avec
                2 canaux."
  ::= { pcmStereo2Chan24 96000 }

-- MIC stéréo mixte

pcmJointStereo OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "Signal audio MIC linéaire stéréo mixte."
  ::= { pcmAudio 3 }

pcmJointStereo2Chan OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "Signal audio MIC linéaire stéréo mixte avec 2 canaux (1 paire
                M-S)."
  ::= { pcmJointStereo 2 }

pcmJointStereo2Chan16 OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "Signal audio MIC linéaire stéréo mixte de 16 bits avec
                2 canaux."
  ::= { pcmJointStereo2Chan 16 }

pcmJointStereo2Chan20 OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "Signal audio MIC linéaire stéréo mixte de 20 bits avec
                2 canaux."
  ::= { pcmJointStereo2Chan 20 }

pcmJointStereo2Chan24 OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "Signal audio MIC linéaire stéréo mixte de 24 bits avec
                2 canaux."
  ::= { pcmJointStereo2Chan 24 }

-- 4.1.2.5 Signal audio MPEG-1, couche 2

mp2Audio OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "Signal audio MPEG-1, couche 2."
  ::= { audioSignalFormat 4 }

-- MP2 mono distinct

mp2Mono OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "Signal audio MP2 mono distinct."
  ::= { mp2Audio 1 }

mp2Mono1Chan OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "Signal audio MP2 mono distinct avec 1 canal."
  ::= { mp2Mono 1 }

-- MP2 mono distinct avec 1 canal à 44 100 Hz

mp2Mono1Chan44100 OBJECT-IDENTITY
  STATUS      current
```

```

DESCRIPTION      "Signal audio MP2 mono distinct de 44,1 kHz avec 1 canal."
::= { mp2Mono1Chan 44100 }

mp2Mono1Chan44100at32 OBJECT-IDENTITY
STATUS            current
DESCRIPTION      "Signal audio MP2 mono distinct de 44,1 kHz à 32 kbit/s avec
                  1 canal."
::= { mp2Mono1Chan44100 32000 }

mp2Mono1Chan44100at40 OBJECT-IDENTITY
STATUS            current
DESCRIPTION      "Signal audio MP2 mono distinct de 44,1 kHz à 40 kbit/s avec
                  1 canal."
::= { mp2Mono1Chan44100 40000 }

mp2Mono1Chan44100at48 OBJECT-IDENTITY
STATUS            current
DESCRIPTION      "Signal audio MP2 mono distinct de 44,1 kHz à 48 kbit/s avec
                  1 canal."
::= { mp2Mono1Chan44100 48000 }

mp2Mono1Chan44100at56 OBJECT-IDENTITY
STATUS            current
DESCRIPTION      "Signal audio MP2 mono distinct de 44,1 kHz à 56 kbit/s avec
                  1 canal."
::= { mp2Mono1Chan44100 56000 }

mp2Mono1Chan44100at64 OBJECT-IDENTITY
STATUS            current
DESCRIPTION      "Signal audio MP2 mono distinct de 44,1 kHz à 64 kbit/s avec
                  1 canal."
::= { mp2Mono1Chan44100 64000 }

mp2Mono1Chan44100at80 OBJECT-IDENTITY
STATUS            current
DESCRIPTION      "Signal audio MP2 mono distinct de 44,1 kHz à 80 kbit/s avec
                  1 canal."
::= { mp2Mono1Chan44100 80000 }

mp2Mono1Chan44100at96 OBJECT-IDENTITY
STATUS            current
DESCRIPTION      "Signal audio MP2 mono distinct de 44,1 kHz à 96 kbit/s avec
                  1 canal."
::= { mp2Mono1Chan44100 96000 }

mp2Mono1Chan44100at112 OBJECT-IDENTITY
STATUS            current
DESCRIPTION      "Signal audio MP2 mono distinct de 44,1 kHz à 112 kbit/s avec
                  1 canal."
::= { mp2Mono1Chan44100 112000 }

mp2Mono1Chan44100at128 OBJECT-IDENTITY
STATUS            current
DESCRIPTION      "Signal audio MP2 mono distinct de 44,1 kHz à 128 kbit/s avec
                  1 canal."
::= { mp2Mono1Chan44100 128000 }

mp2Mono1Chan44100at160 OBJECT-IDENTITY
STATUS            current
DESCRIPTION      "Signal audio MP2 mono distinct de 44,1 kHz à 160 kbit/s avec
                  1 canal."
::= { mp2Mono1Chan44100 160000 }

mp2Mono1Chan44100at192 OBJECT-IDENTITY
STATUS            current
DESCRIPTION      "Signal audio MP2 mono distinct de 44,1 kHz à 192 kbit/s avec
                  1 canal."
::= { mp2Mono1Chan44100 192000 }

mp2Mono1Chan44100at224 OBJECT-IDENTITY

```

```
STATUS          current
DESCRIPTION     "Signal audio MP2 mono distinct de 44,1 kHz à 224 kbit/s avec
                1 canal."
::= { mp2Mono1Chan44100 224000 }

mp2Mono1Chan44100at256 OBJECT-IDENTITY
STATUS          current
DESCRIPTION     "Signal audio MP2 mono distinct de 44,1 kHz à 256 kbit/s avec
                1 canal."
::= { mp2Mono1Chan44100 256000 }

mp2Mono1Chan44100at320 OBJECT-IDENTITY
STATUS          current
DESCRIPTION     "Signal audio MP2 mono distinct de 44,1 kHz à 320 kbit/s avec
                1 canal."
::= { mp2Mono1Chan44100 320000 }

-- MP2 mono distinct avec 1 canal à 48 000 Hz

mp2Mono1Chan48000 OBJECT-IDENTITY
STATUS          current
DESCRIPTION     "Signal audio MP2 mono de 48 kHz avec 1 canal."
::= { mp2Mono1Chan 48000 }

mp2Mono1Chan48000at32 OBJECT-IDENTITY
STATUS          current
DESCRIPTION     "Signal audio MP2 mono distinct de 48 kHz à 32 kbit/s avec
                1 canal."
::= { mp2Mono1Chan48000 32000 }

mp2Mono1Chan48000at40 OBJECT-IDENTITY
STATUS          current
DESCRIPTION     "Signal audio MP2 mono distinct de 8 kHz à 40 kbit/s avec
                1 canal."
::= { mp2Mono1Chan48000 40000 }

mp2Mono1Chan48000at48 OBJECT-IDENTITY
STATUS          current
DESCRIPTION     "Signal audio MP2 mono distinct de 48 kHz à 48 kbit/s avec
                1 canal."
::= { mp2Mono1Chan48000 48000 }

mp2Mono1Chan48000at56 OBJECT-IDENTITY
STATUS          current
DESCRIPTION     "Signal audio MP2 mono distinct de 48 kHz à 56 kbit/s avec
                1 canal."
::= { mp2Mono1Chan48000 56000 }

mp2Mono1Chan48000at64 OBJECT-IDENTITY
STATUS          current
DESCRIPTION     "Signal audio MP2 mono distinct de 48 kHz à 64 kbit/s avec
                1 canal."
::= { mp2Mono1Chan48000 64000 }

mp2Mono1Chan48000at80 OBJECT-IDENTITY
STATUS          current
DESCRIPTION     "Signal audio MP2 mono distinct de 48 kHz à 80 kbit/s avec
                1 canal."
::= { mp2Mono1Chan48000 80000 }

mp2Mono1Chan48000at96 OBJECT-IDENTITY
STATUS          current
DESCRIPTION     "Signal audio MP2 mono distinct de 48 kHz à 96 kbit/s avec
                1 canal."
::= { mp2Mono1Chan48000 96000 }

mp2Mono1Chan48000at112 OBJECT-IDENTITY
STATUS          current
```

```

DESCRIPTION      "Signal audio MP2 mono distinct de 48 kHz à 112 kbit/s avec
                  1 canal."
::= { mp2Mono1Chan48000 112000 }

mp2Mono1Chan48000at128 OBJECT-IDENTITY
STATUS            current
DESCRIPTION      "Signal audio MP2 mono distinct de 48 kHz à 128 kbit/s avec
                  1 canal."
::= { mp2Mono1Chan48000 128000 }

mp2Mono1Chan48000at160 OBJECT-IDENTITY
STATUS            current
DESCRIPTION      "Signal audio MP2 mono distinct de 48 kHz à 160 kbit/s avec
                  1 canal."
::= { mp2Mono1Chan48000 160000 }

mp2Mono1Chan48000at192 OBJECT-IDENTITY
STATUS            current
DESCRIPTION      "Signal audio MP2 mono distinct de 48 kHz à 192 kbit/s avec
                  1 canal."
::= { mp2Mono1Chan48000 192000 }

mp2Mono1Chan48000at224 OBJECT-IDENTITY
STATUS            current
DESCRIPTION      "Signal audio MP2 mono distinct de 48 kHz à 224 kbit/s avec
                  1 canal."
::= { mp2Mono1Chan48000 224000 }

mp2Mono1Chan48000at256 OBJECT-IDENTITY
STATUS            current
DESCRIPTION      "Signal audio MP2 mono distinct de 48 kHz à 256 kbit/s avec
                  1 canal."
::= { mp2Mono1Chan48000 256000 }

mp2Mono1Chan48000at320 OBJECT-IDENTITY
STATUS            current
DESCRIPTION      "Signal audio MP2 mono distinct de 8 kHz à 320 kbit/s avec
                  1 canal."
::= { mp2Mono1Chan48000 320000 }

-- MP2 stéréo

mp2Stereo OBJECT-IDENTITY
STATUS            current
DESCRIPTION      "Signal audio MP2 stéréo."
::= { mp2Audio 2 }

mp2Stereo2Chan OBJECT-IDENTITY
STATUS            current
DESCRIPTION      "Signal audio MP2 stéréo avec 2 canaux (1 paire L-R)."
::= { mp2Stereo 2 }

-- MP2 stéréo avec 2 canaux à 44 100 Hz

mp2Stereo2Chan44100 OBJECT-IDENTITY
STATUS            current
DESCRIPTION      "Signal audio MP2 stéréo de 44,1 kHz."
::= { mp2Stereo2Chan 44100 }

mp2Stereo2Chan44100at32 OBJECT-IDENTITY
STATUS            current
DESCRIPTION      "Signal audio MP2 stéréo de 44,1 kHz à 32 kbit/s."
::= { mp2Stereo2Chan44100 32000 }

mp2Stereo2Chan44100at40 OBJECT-IDENTITY
STATUS            current
DESCRIPTION      "Signal audio MP2 stéréo de 44,1 kHz à 40 kbit/s."
::= { mp2Stereo2Chan44100 40000 }

```

```
mp2Stereo2Chan44100at48 OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "Signal audio MP2 stéréo de 44,1 kHz à 48 kbit/s."
  ::= { mp2Stereo2Chan44100 48000 }

mp2Stereo2Chan44100at56 OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "Signal audio MP2 stéréo de 44,1 kHz à 56 kbit/s."
  ::= { mp2Stereo2Chan44100 56000 }

mp2Stereo2Chan44100at64 OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "Signal audio MP2 stéréo de 44,1 kHz à 64 kbit/s."
  ::= { mp2Stereo2Chan44100 64000 }

mp2Stereo2Chan44100at80 OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "Signal audio MP2 stéréo de 44,1 kHz à 80 kbit/s."
  ::= { mp2Stereo2Chan44100 80000 }

mp2Stereo2Chan44100at96 OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "Signal audio MP2 stéréo de 44,1 kHz à 96 kbit/s."
  ::= { mp2Stereo2Chan44100 96000 }

mp2Stereo2Chan44100at112 OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "Signal audio MP2 stéréo de 44,1 kHz à 112 kbit/s."
  ::= { mp2Stereo2Chan44100 112000 }

mp2Stereo2Chan44100at128 OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "Signal audio MP2 stéréo de 44,1 kHz à 128 kbit/s."
  ::= { mp2Stereo2Chan44100 128000 }

mp2Stereo2Chan44100at160 OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "Signal audio MP2 stéréo de 44,1 kHz à 160 kbit/s."
  ::= { mp2Stereo2Chan44100 160000 }

mp2Stereo2Chan44100at192 OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "Signal audio MP2 stéréo de 44,1 kHz à 192 kbit/s."
  ::= { mp2Stereo2Chan44100 192000 }

mp2Stereo2Chan44100at224 OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "Signal audio MP2 stéréo de 44,1 kHz à 224 kbit/s."
  ::= { mp2Stereo2Chan44100 224000 }

mp2Stereo2Chan44100at256 OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "Signal audio MP2 stéréo de 44,1 kHz à 256 kbit/s."
  ::= { mp2Stereo2Chan44100 256000 }

mp2Stereo2Chan44100at320 OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "Signal audio MP2 stéréo de 44,1 kHz à 320 kbit/s."
  ::= { mp2Stereo2Chan44100 320000 }

-- MP2 stéréo avec 2 canaux à 48 000 Hz

mp2Stereo2Chan48000 OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "Signal audio MP2 stéréo de 48 kHz."
  ::= { mp2Stereo2Chan 48000 }

mp2Stereo2Chan48000at32 OBJECT-IDENTITY
```

```

STATUS          current
DESCRIPTION      "Signal audio MP2 stéréo de 48 kHz à 32 kbit/s."
::= { mp2Stereo2Chan48000 32000 }

mp2Stereo2Chan48000at40 OBJECT-IDENTITY
STATUS          current
DESCRIPTION      "Signal audio MP2 stéréo de 48 kHz à 40 kbit/s."
::= { mp2Stereo2Chan48000 40000 }

mp2Stereo2Chan48000at48 OBJECT-IDENTITY
STATUS          current
DESCRIPTION      "Signal audio MP2 stéréo de 48 kHz à 48 kbit/s."
::= { mp2Stereo2Chan48000 48000 }

mp2Stereo2Chan48000at56 OBJECT-IDENTITY
STATUS          current
DESCRIPTION      "Signal audio MP2 stéréo de 48 kHz à 56 kbit/s."
::= { mp2Stereo2Chan48000 56000 }

mp2Stereo2Chan48000at64 OBJECT-IDENTITY
STATUS          current
DESCRIPTION      "Signal audio MP2 stéréo de 48 kHz à 64 kbit/s."
::= { mp2Stereo2Chan48000 64000 }

mp2Stereo2Chan48000at80 OBJECT-IDENTITY
STATUS          current
DESCRIPTION      "Signal audio MP2 stéréo de 48 kHz à 80 kbit/s."
::= { mp2Stereo2Chan48000 80000 }

mp2Stereo2Chan48000at96 OBJECT-IDENTITY
STATUS          current
DESCRIPTION      "Signal audio MP2 stéréo de 48 kHz à 96 kbit/s."
::= { mp2Stereo2Chan48000 96000 }

mp2Stereo2Chan48000at112 OBJECT-IDENTITY
STATUS          current
DESCRIPTION      "Signal audio MP2 stéréo de 48 kHz à 112 kbit/s."
::= { mp2Stereo2Chan48000 112000 }

mp2Stereo2Chan48000at128 OBJECT-IDENTITY
STATUS          current
DESCRIPTION      "Signal audio MP2 stéréo de 48 kHz à 128 kbit/s."
::= { mp2Stereo2Chan48000 128000 }

mp2Stereo2Chan48000at160 OBJECT-IDENTITY
STATUS          current
DESCRIPTION      "Signal audio MP2 stéréo de 48 kHz à 160 kbit/s."
::= { mp2Stereo2Chan48000 160000 }

mp2Stereo2Chan48000at192 OBJECT-IDENTITY
STATUS          current
DESCRIPTION      "Signal audio MP2 stéréo de 48 kHz à 192 kbit/s."
::= { mp2Stereo2Chan48000 192000 }

mp2Stereo2Chan48000at224 OBJECT-IDENTITY
STATUS          current
DESCRIPTION      "Signal audio MP2 stéréo de 48 kHz à 224 kbit/s."
::= { mp2Stereo2Chan48000 224000 }

mp2Stereo2Chan48000at256 OBJECT-IDENTITY
STATUS          current
DESCRIPTION      "Signal audio MP2 stéréo de 48 kHz à 256 kbit/s."
::= { mp2Stereo2Chan48000 256000 }

mp2Stereo2Chan48000at320 OBJECT-IDENTITY
STATUS          current
DESCRIPTION      "Signal audio MP2 stéréo de 48 kHz à 320 kbit/s."
::= { mp2Stereo2Chan48000 320000 }

```

```
-- MP2 stéréo mixte

mp2JointStereo OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "Signal audio MP2 stéréo mixte."
  ::= { mp2Audio 3 }

mp2JointStereo2Chan OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "Signal audio MP2 stéréo mixte avec 2 canaux (1 paire M-S)."
```

-- 4.1.2.6 Signal audio MPEG-1, couche 3

```
  ::= { mp2JointStereo 2 }

mp2JointStereo2Chan44100 OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "Signal audio MP2 stéréo mixte de 44,1 kHz."
  ::= { mp2JointStereo2Chan 44100 }

mp2JointStereo2Chan48000 OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION  "Signal audio MP2 stéréo mixte de 48 kHz."
  ::= { mp2JointStereo2Chan 48000 }
```

mp3Audio OBJECT-IDENTITY

```
  STATUS      current
  DESCRIPTION  "Signal audio MPEG-1, couche 3."
  ::= { audioSignalFormat 5 }
```

mp3Mono OBJECT-IDENTITY

```
  STATUS      current
  DESCRIPTION  "Signal audio MP3 mono distinct."
  ::= { mp3Audio 1 }
```

mp3Mono1Chan OBJECT-IDENTITY

```
  STATUS      current
  DESCRIPTION  "Signal audio MP3 mono distinct avec 1 canal."
  ::= { mp3Mono 1 }
```

mp3Stereo OBJECT-IDENTITY

```
  STATUS      current
  DESCRIPTION  "Signal audio MP3 stéréo."
  ::= { mp3Audio 2 }
```

mp3Stereo2Chan OBJECT-IDENTITY

```
  STATUS      current
  DESCRIPTION  "Signal audio MP3 stéréo avec 2 canaux (1 paire L-R)."
```

mp3JointStereo OBJECT-IDENTITY

```
  STATUS      current
  DESCRIPTION  "Signal audio MP3 stéréo mixte."
  ::= { mp3Audio 3 }
```

mp3JointStereo2Chan OBJECT-IDENTITY

```
  STATUS      current
  DESCRIPTION  "Signal audio MP3 stéréo mixte avec 2 canaux (1 paire M-S)."
```

-- 4.1.2.7 Signal audio AAC

aacAudio OBJECT-IDENTITY

```
  STATUS      current
  DESCRIPTION  "Signal audio AAC."
  ::= { audioSignalFormat 6 }
```

aacLC OBJECT-IDENTITY

```
  STATUS      current
```

```

DESCRIPTION    "Signal audio AAC avec profil à faible complexité."
::= { aacAudio 1 }

aacMain OBJECT-IDENTITY
STATUS         current
DESCRIPTION    "Signal audio AAC avec profil principal."
::= { aacAudio 2 }

aacSRS OBJECT-IDENTITY
STATUS         current
DESCRIPTION    "Signal audio AAC avec profil évolutif selon la fréquence
               d'échantillonnage."
::= { aacAudio 3 }

aacLTP OBJECT-IDENTITY
STATUS         current
DESCRIPTION    "Signal audio AAC avec profil de prévision à long terme."
::= { aacAudio 4 }

aacLD OBJECT-IDENTITY
STATUS         current
DESCRIPTION    "Signal audio AAC avec profil à faible latence."
::= { aacAudio 5 }

-- 4.1.2.8 Signal audio G711

g711Audio OBJECT-IDENTITY
STATUS         current
DESCRIPTION    "Signal audio G711."
::= { audioSignalFormat 7 }

g711ALaw OBJECT-IDENTITY
STATUS         current
DESCRIPTION    "Signal audio G711 qui indique un codage par loi A."
::= { g711Audio 1 }

g711MuLaw OBJECT-IDENTITY
STATUS         current
DESCRIPTION    "Signal audio G711 qui indique un encodage par loi mu."
::= { g711Audio 2 }

-- 4.1.2.9 Signal audio G722

g722Audio OBJECT-IDENTITY
STATUS         current
DESCRIPTION    "Signal audio G722."
::= { audioSignalFormat 8 }

g722at48 OBJECT-IDENTITY
STATUS         current
DESCRIPTION    "Signal audio G722 à 48 kbit/s."
::= { g722Audio 48 }

g722at56 OBJECT-IDENTITY
STATUS         current
DESCRIPTION    "Signal audio G722 à 56 kbit/s."
::= { g722Audio 56 }

g722at64 OBJECT-IDENTITY
STATUS         current
DESCRIPTION    "Signal audio G722 à 64 kbit/s."
::= { g722Audio 64 }

-- 4.1.2.10 Signal audio APT-X

aptXAudio OBJECT-IDENTITY
STATUS         current

```

```
DESCRIPTION    "Signal audio APT-X."
::= { audioSignalFormat 9 }

-- 4.1.2.11 Signal audio APT-X amélioré

enhAptXAudio OBJECT-IDENTITY
STATUS          current
DESCRIPTION     "Signal audio APT-X amélioré."
::= { audioSignalFormat 10 }

-- 4.1.2.12 Signal audio J41

j41Audio OBJECT-IDENTITY
STATUS          current
DESCRIPTION     "Signal audio J41."
::= { audioSignalFormat 11 }

j41ALawA OBJECT-IDENTITY
STATUS          current
DESCRIPTION     "Signal audio J41 qui utilise une compression-expansion de loi A,
                variante A."
::= { j41Audio 1 }

j41ALawB OBJECT-IDENTITY
STATUS          current
DESCRIPTION     "Signal audio J41 qui utilise une compression-expansion de loi A,
                variante B."
::= { j41Audio 2 }

j41Nic OBJECT-IDENTITY
STATUS          current
DESCRIPTION     "Signal audio J41 qui utilise une compression-expansion quasi
                instantanée."
::= { j41Audio 3 }

-- 4.1.2.13 Signal audio J57

j57Audio OBJECT-IDENTITY
STATUS          current
DESCRIPTION     "Signal audio J57."
::= { audioSignalFormat 12 }

j57H11 OBJECT-IDENTITY
STATUS          current
DESCRIPTION     "J57 qui utilise un canal H11."
::= { j57Audio 1 }

j57H12 OBJECT-IDENTITY
STATUS          current
DESCRIPTION     "J57 qui utilise un canal H12."
::= { j57Audio 2 }

-- 4.1.2.14 Signal audio non valide

invalidAudio OBJECT-IDENTITY
STATUS          current
DESCRIPTION     "Indique une erreur, telle que l'incapacité à décoder un signal
                en amont dans la chaîne."
::= { audioSignalFormat 13 }

-- 4.2 Définitions de formats de transport audio

audioTransportFormat OBJECT IDENTIFIER ::= { audioFormat transport(2) }

unspecifiedTransport OBJECT-IDENTITY
```

```

STATUS          current
DESCRIPTION      "Transport non spécifié."
::= { audioTransportFormat 0 }

analogue OBJECT-IDENTITY
STATUS          current
DESCRIPTION      "Transport analogique."
::= { audioTransportFormat 1 }

aes3 OBJECT-IDENTITY
STATUS          current
DESCRIPTION      "Transport AES3."
::= { audioTransportFormat 2 }

aes10 OBJECT-IDENTITY
STATUS          current
DESCRIPTION      "Transport AES10."
::= { audioTransportFormat 3 }

aes50 OBJECT-IDENTITY
STATUS          current
DESCRIPTION      "Transport AES50."
::= { audioTransportFormat 4 }

-- Métadonnées

audioMetadataFormat OBJECT IDENTIFIER ::= { audioFormat 3 }

unspecifiedMetadata OBJECT-IDENTITY
STATUS          current
DESCRIPTION      "Métadonnées non spécifiées."
::= { audioMetadataFormat 0 }

END

```

IECNORM.COM : Click to view the full PDF of IEC 62379-2:2008

Annexe B (informative)

Définitions de blocs audio lisibles par une machine

La présente annexe fournit une version lisible par une machine des définitions de blocs audio, destinée à être interprétable par des outils logiciels d'exploration de la MIB normalisés. Elle n'exprime pas toutes les exigences de la norme, par exemple lorsque l'accès à un objet est limité à certains niveaux de privilège. En cas d'incohérence entre la présente annexe et l'Article 5, l'Article 5 prévaut.

Le format utilisé pour décrire les objets MIB est conforme à l'IETF STD 58 (SMIv2).

```
IEC62379-2-MIB DEFINITIONS ::= BEGIN

IMPORTS

    MODULE-IDENTITY, OBJECT-TYPE
        FROM SNMPv2-SMI

    OBJECT-GROUP, MODULE-COMPLIANCE
        FROM SNMPv2-CONF

    TEXTUAL-CONVENTION, TruthValue
        FROM SNMPv2-TC

    IntegerNumber, CardinalNumber, IndexNumber, Utf8String, BlockId,
    MediaFormat, PortDirection, iec62379, OperationId, ObjectName,
    ObjectValue, PrivilegeLevel, OperationState
        FROM IEC62379-1-MIB;

audioMIB MODULE-IDENTITY
    LAST-UPDATED "200804301500Z"
    ORGANIZATION "IEC PT62379"
    CONTACT-INFO "Non spécifié"
    DESCRIPTION "Module MIB pour la gestion des fonctions audio dans les
        équipements conformes à l'IEC 62379."
    REVISION "200804301500Z"
    DESCRIPTION "Edition 1.0, version finale".
    ::= { audio 1 }

audio OBJECT IDENTIFIER ::= { iec62379 2 }

-- Valeurs d'identificateur d'objet pour les déclarations de conformité de module
audioMIBCompliance OBJECT IDENTIFIER ::= { audioMIB 0 }

-- Valeurs d'identificateur d'objet pour les groupes d'objets MIB
-- les définitions suivantes sont également utilisées comme valeurs BlockType

audioPort          OBJECT IDENTIFIER ::= { audioMIB 1 }

audioMixer         OBJECT IDENTIFIER ::= { audioMIB 2 }

audioCrosspoint    OBJECT IDENTIFIER ::= { audioMIB 3 }

audioClipPlayer    OBJECT IDENTIFIER ::= { audioMIB 4 }

audioLimiter       OBJECT IDENTIFIER ::= { audioMIB 5 }

audioConverter     OBJECT IDENTIFIER ::= { audioMIB 6 }

audioLevelAlarm    OBJECT IDENTIFIER ::= { audioMIB 7 }
```

-- 5.2 Définitions de types

-- 5.2.1 Conventions textuelles

```

AudioTransportType ::= TEXTUAL-CONVENTION
    STATUS          current
    DESCRIPTION     "Référence au transport utilisé pour une connexion audio."
    SYNTAX          OBJECT IDENTIFIER

AudioLevel ::= TEXTUAL-CONVENTION
    STATUS          current
    DESCRIPTION     "Niveau audio absolu ou relatif en unités de 0,01 dB."
    SYNTAX          INTEGER (-20000..20000)
                    -- {
                    --   mInfinity (-20000),
                    --   fullScale (0),
                    --   pInfinity (20000)
                    -- } (mInfinity..pInfinity)

AudioPhase ::= TEXTUAL-CONVENTION
    STATUS          current
    DESCRIPTION     "Valeur de phase absolue ou relative en unités de 0,01 degré."
    SYNTAX          INTEGER (-18000..18000)

AudioQuality ::= TEXTUAL-CONVENTION
    STATUS          current
    DESCRIPTION     "Énumération qui identifie un niveau de qualité de traitement du
                    signal audio. La sémantique est spécifique à l'équipement. Les
                    équipements qui prennent en charge les niveaux de qualité
                    sélectionnables doivent au moins prendre en charge les valeurs
                    "low" et "high"."
    SYNTAX          INTEGER (1..127)
                    -- {
                    --   worst (1),
                    --   low (32),
                    --   high (96),
                    --   best (127)
                    -- } (worst..best)

AudioChannel ::= TEXTUAL-CONVENTION
    STATUS          current
    DESCRIPTION     "Énumération qui identifie un canal audio. L'équipement stéréo
                    doit accepter les valeurs "left" et "right". Il convient qu'un
                    équipement multicanal accepte une plage contiguë de valeurs à
                    partir de 1; il peut ignorer l'association des valeurs 1 et 2
                    avec "left" et "right"."
    SYNTAX          INTEGER (1..240)
                    -- {
                    --   left (1),
                    --   right (2)
                    -- } (1..240)

Aes3ChannelData ::= TEXTUAL-CONVENTION
    STATUS          current
    DESCRIPTION     "Données d'état de canal associées à un canal AES3."
    SYNTAX          OCTET STRING (SIZE(24))

Aes3UserData ::= TEXTUAL-CONVENTION
    STATUS          current
    DESCRIPTION     "Données d'état utilisateur associées à un canal AES3."
    SYNTAX          OCTET STRING (SIZE(24))

AudioClipSelector ::= TEXTUAL-CONVENTION
    STATUS          current
    DESCRIPTION     "Énumération qui identifie un clip audio. La sémantique est
                    spécifique à l'équipement."
    SYNTAX          INTEGER (1..2147483647)

AudioClipActivation ::= TEXTUAL-CONVENTION

```