INTERNATIONAL ELECTROTECHNICAL COMMISSION

_____

Common Control Interface –

Part 7: Measurements

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 itself does not provide any attestation of conformity. Independent certification bodies provide conformity assessment services and, in some areas, access to IEC marks of conformity. IEC is not responsible for any services carried out by independent certification bodies.
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 XXX has been prepared by subcommittee XX: TITLE, of IEC technical committee XX:

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

FDIS	Report on voting
XX/XX/FDIS	XX/XX/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 stability date[1] 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.

1 The National Committees are requested to note that for this publication the stability date is ....

0 Introduction

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

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

This part of IEC 62379 specifies those aspects that are specific for using the Block structure as defined in IEC 62379-1, for standardising the collection method of audio and video parameters for use by the European Broadcasting Union Expert Communities Networks – IP Measurements (EBU ECN-IPM) Group.

The collection of network related parameters may be outside the scope of this document. These are expected to be collected from the standard IETF MIBS that are generally present in most (if not all) networked equipment. Some specific network parameters are included that are not obtainable from existing standard IETF MIBS.

0.1 Structure of the family of standards

IEC 62379 specifies the Common Control Interface, a protocol for managing networked audiovisual equipment. It is intended to include the following Parts:

1) General
2) Audio
3) Video
4) Data
5) Transmission over networks
6) Packet transfer service
7) Measurement

Part 1 specifies aspects which are common to all equipment.

Parts 2 to 4 specify control of internal functions specific to equipment carrying particular types of live media. Part 4 does not refer to packet data such as the control messages themselves.

Part 5 specifies control of transmission of these media over each individual network technology. It includes network specific management interfaces along with network specific control elements that integrate into the control framework.

Part 6 specifies carriage of control and status messages and non-audiovisual data over transports that do not support audio and video, such as RS232 serial links, with (as with Part 5) a separate subpart for each technology.

Part 7 (this document) specifies those aspects that are specific to the measurement requirements of the EBU ECN-IPM Group.

An introduction to the Common Control Interface is given in IEC 62739-1.

0.2 Description, aims and requirements of EBU ECN-IPM Group

In recent years, EBU Members have been increasingly adopting IP networks for the contribution of audio and video in real-time. It is well known that although IP networks are of lower cost and provide more flexibility compared with circuit switched networks, they suffer from longer delays and have much larger jitter, while broadcasters' tolerance to these variables is much less than that of normal business IT traffic.

To respond to Members' use of IP the EBU set up two groups, ECN-ACIP (Audio contribution over IP) and ECN-VCIP (Video contribution over IP) with the tasks of drawing up recommended codes of practice[2].

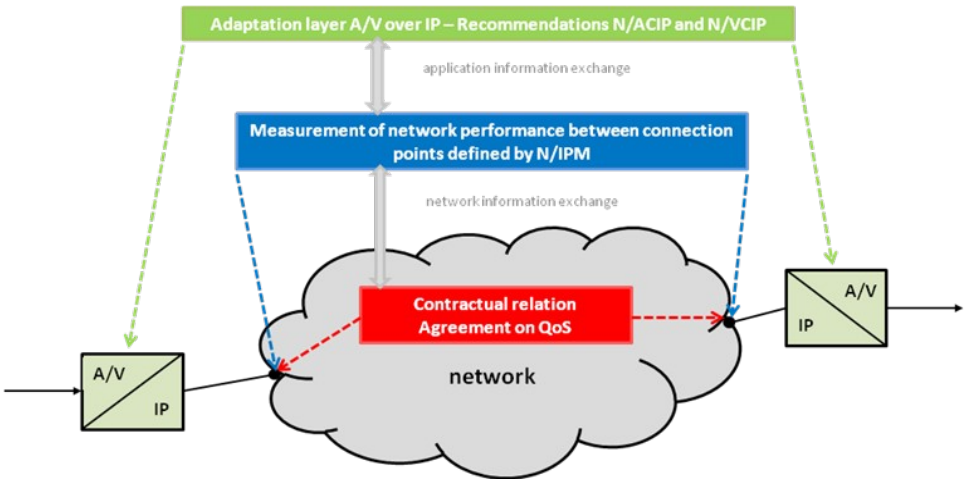


Figure 1 – Relationships between ECN groups ACIP, VCIP and IPM

It was also recognised that there would be a strong demand for tools that would enable broadcasters to measure and manage their IP networks properly to suit the many time-critical broadcast applications they would be subjected to. To this end, the ECN-IPM (IP measurement) group was set up. The relationships between these three groups are shown in the following figure.

The goals of ECN-IPM group were:

– Define a Quality of Service classification to achieve requested A/V transmission quality for broadcast applications
– Standardise network information exchange between EBU Members and Telecom suppliers
– Propose a method of collecting end-to-end performance information for management purposes

In achieving these goals the ECN-IPM Group has specified a set of parameters that are important for broadcasters when using IP networks for audio and video transmission and has developed a software mechanism to probe a network for device and topology discovery, physical path tracing for both end-to-end communication and multicast streams, with the potential for multilayer monitoring for streams on a multi-vendor network with fully media-specific parameters.

The specified parameters cover both the network layer and application layer (for video and audio). SNMP is employed to collect information on the status of networked devices, such as the transmission rate, error rate, the codec used and multicast streams status.

To ensure that all the parameters can be recovered from a variety of different manufacturers' IP equipment, the group has designed a MIB (Management Information Base). Although many MIB files have been published over the years, especially on the network side, very little standardisation work has been done on A/V codec MIB files. The EBU ECN-IPM group has therefore proposed a new standard, based upon IEC 62379 (Common Control Interface for Networked Audio and Video Systems) to address this issue.

Two EBU technical publications have been produced by the ECN-IPM group:

2 ECN-ACIP and ECN-VCIP were formally known as N/ACIP and N/VCIP respectively.

The parameters and new MIB information may be found in EBU-Tech 3345, End-to-End IP Network Measurement for Broadcast Applications - Parameters & Management Information Base (MIB), Geneva, July 2011.

A description of the software mechanism, EisStream[3], may be found in EBU Tech 3346, End-to-End IP Network Measurement for Broadcast Applications - EisStream Software package description, Geneva, July 2011. The software is written in Java and it provides physical path tracing for IP traffic using SNMP.

This document, IEC 62379-7 and other related parts of IEC 62379, is the standard upon which Section 3 of EBU Tech 3345 is based.

If there is any inconsistency between this document and Section 3 of EBU Tech 3345, then this document, IEC 62379-7 and other related parts of IEC 62379, takes precedence.

3 EBU Integrated Monitoring Solution for Media Streams on IP Networks, http://eisstream.sourceforge.net/

Common Control Interface –

Part 7: Measurements

1 Scope

This International Standard specifies aspects of the Common Control Interface of IEC 62379-1 that are specific to the measurement of the service experienced by audio and video streams and in particular to the requirements of EBU ECN-IPM Measurements Group.

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.

IEC 62379-1:2007 Common Control Interface for Networked Audio and Video Products – Part 1: General

IEC 62379-2:2008 Common Control Interface for Networked Audio and Video Products – Part 2: Audio

IEC 62379-3 Ed.1: Common Control Interface for Networked Audio and Video Products – Part 3: Video

EBU-TECH 3345 End-to-End IP Network Measurement for Broadcast Applications - Parameters & Management Information Base (MIB), Geneva, July 2011

3 Terms, definitions and abbreviations

For the purposes of this International Standard, the terms and definitions given in IEC 62379-1, IEC 62379-2, IEC 62379-3 and the following apply.

3.1 Abbreviations

Internet Group Management Protocol
IGMP

Time Stamped Delay Factor
TS-DF

Programme ID
PID

Standard Definition
SD

High Definition
HD

Forward Error Correction
FEC

Asynchronous Serial Interface
ASI

Session Initiation Protocol
SIP

Media delivery Index
MDI

Real-time Protocol
RTP

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 as defined and specified in IEC 62379-2 or an object identifier defined elsewhere.

NOTE 1 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

NOTE 2 The audio signal format definitions specified in IEC 62379-2 are used in a common manner by both audio only units and the one or more audio components associated with a video flow

5 Video format definitions

5.1 Video signal format definitions

At any point in the video signal chain, the video 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 as defined and specified in IEC 62379-3 Ed.1 or an object identifier defined elsewhere.

NOTE Permitting video 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

6 MIB definitions for measurement information blocks

6.1 General

This clause defines a set of managed object types for representing measurement information functions in network controlled audio/video equipment for the purposes of standardising the collection method of audio, video and some network parameters for use by the EBU ECN-IPM Group.

The format of the definitions is as specified in IEC 62379-1.

For measurement purposes, a piece of audio/video equipment shall be modelled as a number of discrete measurement blocks, as specified in IEC 62379-1. Each measurement block shall have zero or more inputs and zero or more outputs.

NOTE 1 Information is transferred into these objects internally within the equipment utilising this standard, from existing MIB objects, or elsewhere, within the equipment. A management station can access the measurement information using the standard OIDs defined in this standard independent of the manufacturer.

Each measurement block shall be modelled either by one of the standard measurement block types defined in this standard or by a measurement 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 measurement 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 6.7.1.11 contains a worked examples of the use of the measurement block structure.

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 representing managed object values.

6.2.1 Textual conventions

```
NetworkType ::= INTEGER {
  ipv4 (1),
  ipv6 (2),
  asi  (3)
} (ipv4..asi)
-- An enumeration identifying a network type of over which the
-- media is flowing.

TransportType ::= INTEGER {
  notApplicable (0),
  rtp (1)
} (notApplicable.. rtp)
-- An enumeration identifying a transport type of over which the
-- media is flowing.

-- Note that the values for this textual convention are NOT the same
-- as the numbers used in the protocol field of IPv4 packets and
-- the Next Header Field of IPv6 packets.
-- See http://www.iana.org/assignments/protocol-numbers

AudioFECType ::= INTEGER {
  none(0),
  smpte2021(1),
  smpte2022(2),
  rfc2733(3),
  proprietary(4),
} (none..proprietary)
-- An enumeration identifying the FEC type applied, if present.

VideoFECType ::= INTEGER {
  none        (0),
```

```
  smpte2022   (2),
  proprietary (4)
} (none..proprietary)
-- An enumeration identifying the FEC type applied,if present.

BufferSize ::= Unsigned32
-- A type to indicate the current total size of the receive buffer in
-- msecs.

BufferOcpncyTime ::= Gauge32
-- A type to report the amount of data, expressed in msecs,
-- occupying the receive buffer.

BufferOcpncyPercent ::= INTEGER (1..100)
-- A type to report the amount of data, expressed as a
-- percentage of the total receive buffer size, occupying the
-- receive buffer.

TemperatureLocn ::= OCTET STRING (0..80)
-- A type to indicate the location where the temperature is measured.

TemperatureTrend ::= Gauge32
-- A type to report the current temperature at the measured location.
-- The use of this type will allow changes (either up or down)
-- to be reported.

TemperatureStatus ::= INTEGER {
  undetermined   (0),
..other          (1),
  unknown        (2),
  ok             (3),
..warning        (4),
  critical       (5),
  nonRecoverable (6)
} (undetermined.. nonRecoverable)
-- An enumeration identifying the temperature status levels.
-- Semantics are equipment specific.

BitRateType ::= INTEGER {
  unspecified (0),
  vbr         (1),
  cbr         (2)
} (unspecified..cbr)
-- An enumeration identifying the video bit rate type applied
--
-- vbr = variable bit rate
-- cbr = constant bit rate
```

6.2.2 Sequences

```
nMtEntry ::= SEQUENCE {
  nMtBlockId        BlockId,
  nMtIfIndex        InterfaceIndex,
  nMtTxRxPoint      TruthValue,
  nMtNetworkType    NetworkType,
  nMtTransportType  TransportType,
  nMtTxRxAddr       TAddress,
  nMtPortNumber     CardinalNumber,
  nMtIGMPVersion    CardinalNumber,
  nMtSIPServerAddr  TAddress
}

aMtBlockEntry ::= SEQUENCE {
```

```
  aMtBlockId                BlockId,
  aMtAudioComponentNumber   IndexNumber,
  aMtNetworkBlockId         BlockId,
  aMtAudioStatus            TruthValue,
  aMtAudioSignalFormat      MediaFormat,
  aMtAudioPID               CardinalNumber,
  aMtIfIndex                InterfaceIndex,
  aMtFECType                AudioFECType,
  aMtFECLengthDimension     IntegerNumber
}

vMtEntry ::= SEQUENCE {
  vMtBlockId              BlockId,
  vMtAudioBlockId         BlockId,
  vMtNetworkBlockId       BlockId,
  vMtVideoStatus          TruthValue,
  vMtVideoSourceFormat    MediaFormat,
  vMtVideoCodingType      MediaFormat,
  vMtVideoBitRateType     BitRateType,
  vMtVideoBitRate         CardinalNumber,
  vMtAspectRatio          MediaFormat,
  vMtFECType              VideoFECType,
  vMtFECLengthDimension   IntegerNumber,
  vMtTrickModeSupport     TruthValue
}

rxPointEntry ::= SEQUENCE {
  rxPointBlockId             BlockId,
  rxPointNetworkBlockId      BlockId
  rxPointBufferSize          BufferSize,
  rxPointBufferOcpancyTime   BufferOcpncyTime,
  rxPointBufferOcpncyPcnt    BufferOcpncyPercent,
  rxPointMDI                 Utf8String,
  rxPointTSDF                CardinalNumber
}
```

[Should we make a note here, or elsewhere, that temperature may possibly need to be put into Part 1?]

```
temperatureEntry ::= SEQUENCE {
  temperatureBlockId        BlockId,
  temperatureLocnNumber     IndexNumber,
  temperatureLocation       TemperatureLocn,
  temperatureTrend          TemperatureTrend,
  temperatureStatus         TemperatureStatus,
  temperatureLowWarning     IntegerNumber,
  temperatureHighWarning    IntegerNumber,
  temperatureLowCritical    IntegerNumber,
  temperatureHighCritical   IntegerNumber
}
```

6.3 Network measurement information blocks

6.3.1 Generic network measurement functionality

A network measurement information block shall have the following structure where c is the number of channels on a connection:

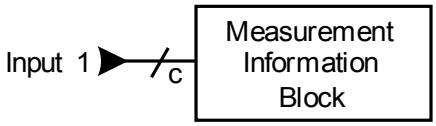


Figure 2 – Network Measurement Information block

A network measurement information block shall consist of a generic measurement information block containing information specific to network parameters that are to be available for use by a management agent.

NOTE This is a special type of block that has no internal functionality. It is merely a means to gather information from within the unit from, for example, other tables and objects or direct from the media stream and for it to be presented and used in a standardised way to a management system. This specification does not define where the information for the content of these blocks is sourced from.

The group of objects in Table 1 shall be implemented by all compliant equipment that has a management model that incorporates one or more measurement information block(s). The root node for these objects shall be

{ iso(1) standard(0) iec62379 measurement(7) measurementMIB(1) networkMeasurement(1) }

This node shall be used as the block type identifier for network measurement information blocks.

Table 1 - Managed objects for network measurement information blocks

Identifier	Syntax	Index	Readable	Writable	Volatile	Status
nMtBlockTable(1)	SEQUENCE OF NMtBlockEntry		none	none	no	m
└nMtBlockEntry(1)	NMtBlockEntry		none	none	no	m
├nMtBlockId(1)	BlockId	yes	none	none	no	m
├nMtIfIndex(2)	InterfaceIndex		listener	none	maybe	m
├nMtTxRxPoint(3)	TruthValue		listener	supervisor	no	m
├nMtNetworkType(4)	NetworkType		listener	none	maybe	m
├nMtTransportType(5)	TransportType		listener	none	maybe	m
├nMtTxRxAddr(6)	TAddress		listener	none	maybe	m
├nMtPortNumber(7)	CardinalNumber		listener	none	maybe	m
├nMtIGMPVersion(8)	CardinalNumber		listener	none	maybe	o
└nMtSIPServerAddr(9)	TAddress		listener	supervisor	maybe	o

6.3.1.1 nMtBlockTable

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

6.3.1.2 nMtBlockEntry

An entry in the network measurement block table.

6.3.1.3 nMtBlockId

The block identifier for this block. Used as an index when accessing the network measurement block table.

6.3.1.4 nMtIfIndex

The identifier for the associated network interfaces object within MIB-II. This object shall be a copy of `ifIndex` (actually Textual Convention `InterfaceIndex` - see below from IF-MIB) from RFC1213-MIB (MIB-II) (1.3.6.1.2.1.2.2.1.1) so as to provide a link between this network measurement block and the network interface it is associated with.

If `ifIndex` is not used within the unit, then an equivalent number to identify the interface should be used. This shall be greater than zero

NOTE This is a note about and Description of Textual Convention `InterfaceIndex` from IF-MIB. `InterfaceIndex` contains the semantics of `ifIndex` and should be used for any objects defined in other MIB modules that need these semantics. A unique value, greater than zero, for each interface or interface sub-layer in the managed system. It is recommended that values are assigned contiguously starting from 1. The value for each interface sub-layer must remain constant at least from one re-initialization of the entity's network management system to the next re-initialization.

6.3.1.5 nMtTxRxPoint

This represents the measurement position in the media chain; whether the unit is a transmitter or receiver of the media. In the case of bi-directional media flow, then this value should be set to indicate the position from where the measurement is required to be made; from a media transmission or reception point of view.

- Transmitter (Tx) = `false`
- Receiver (Rx) = `true`

If the value of this object is `true` (Rx), then the receiver table (rxPointTable, clause 6.6.1.1) shall exist, otherwise its existence is optional.

6.3.1.6 nMtNetworkType

A description of the current network type over which the media flow is flowing.

If the network type is `asi(3)`, then the remaining entries of

- nMtTransportType
- nMtTxRxAddr
- nMtPortNumber
- nMtIGMPVersion
- nMtSIPServerAddr

in this table are not required.

6.3.1.7 nMtTransportType

A description of the transport type being used to convey the media.

If `nMtNetworkType` is `asi(3)`, then `notApplicable(0)` shall be returned.

6.3.1.8 nMtTxRxAddr

The network address of either the send or receive point network interface over which the media is flowing. Which it is, is dependent on the measurement position in the media chain; whether the unit is a transmitter or receiver of the media. The value of `nMtTxRxPoint` shall determine which end is being referred to.

If `nMtNetworkType` is `asi(3)`, then `NULL` shall be returned.

6.3.1.9 `nMt`PortNumber

The port number assigned to the port over which the media stream or other application of interest is being measured.

A port number is a 16-bit unsigned integer, ranging from 0 to 65535.

If `nMtNetworkType` is `asi(3)`, then `zero` shall be returned.

6.3.1.10 `nMt`IGMPVersion

The version of Internet Group Management Protocol (IGMP) being used.

If `nMtNetworkType` is `asi(3)`, then `zero` shall be returned.

If IGMP is not supported, then `zero` shall be returned.

6.3.1.11 `nMt`SIPServerAddr

The IP address of the SIP server which the unit may be registered with.

If `nMtNetworkType` is `asi(3)`, then `NULL` shall be returned.

If the unit is not registered with a SIP server, then `NULL` shall be returned.

6.4 Audio measurement information blocks

6.4.1 Generic audio measurement functionality

An audio measurement information block shall have the following structure where c is the number of channels on a connection:

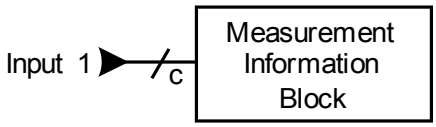


Figure 3 – Audio Measurement Information block

An audio measurement information block shall consist of a generic measurement information block containing information specific to audio parameters that are to be available for use by a management agent.

NOTE This is a special type of block that has no internal functionality. It is merely a means to gather information from within the unit from, for example, other tables and objects or direct from the media stream, through the input and for this information to be used and presented in a standardised way to a management system. This specification does not define where the information for the content of these blocks is sourced from.

The group of objects in Table 2 shall be implemented by all compliant equipment that has a management model that incorporates one or more measurement information block(s). The root node for these objects shall be

{ iso(1) standard(0) iec62379 measurement(7) measurementMIB(1) audioMeasurement(2) }

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

Table 2 - Managed objects for audio measurement information blocks

Identifier	Syntax	Index	Readable	Writable	Volatile	Status
aMtBlockTable(1)	SEQUENCE OF AMtBlockEntry		none	none	no	m
└aMtBlockEntry(1)	AMtBlockEntry		none	none	no	m
├aMtBlockId(1)	BlockId	yes	none	none	no	m
├aMtAudioComponentNumber(2)	IndexNumber	yes	none	none	maybe	m
├aMtNetworkBlockId(3)	BlockId		listener	none	no	m
├aMtAudioStatus(4)	TruthValue		listener	none	yes	m
├aMtAudioSignalFormat(5)	MediaFormat		listener	none	yes	m
├aMtAudioPId(6)	CardinalNumber		listener	none	yes	m
├aMtIfIndex(7)	InterfaceIndex		listener	none	maybe	m
├aMtFECType(8)	AudioFECType		listener	none	yes	o
└aMtFECLengthDimension(9)	IntegerNumber		listener	none	maybe	o

6.4.1.1 aMtBlockTable

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

6.4.1.2 aMtBlockEntry

An entry in the audio measurement block table.

6.4.1.3 aMtBlockId

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

6.4.1.4 aMtAudioComponentNumber

The audio component identifier. The audio component number is the identifier for each audio item, when there are one or more audio items associated with a single video stream/signal.

For an audio only unit, this identifier shall also be used to identify each audio item, when there are one or more audio items present within the unit.

Note An audio item here refers to a single audio stream/signal. Information pertaining to the number of channels within the audio item, such as stereo for two channels, for example, is contained within the 6.4.1.7 aMtAudioSignalFormat object.

Used as an index when accessing the audio measurement block table.

6.4.1.5 aMtNetworkBlockId

The identifier for the associated network measurement block. This provides a link between the audio measurement block and the network measurement block.

6.4.1.6 aMtAudioStatus

If `true`, indicates the audio signal is present. If `false`, indicates the audio signal is not present.

6.4.1.7 aMtAudioSignalFormat

A description of the current audio signal format.

If `aMtAudioPId > 0`, this should be the audio signal format associated with this PID.

If `aMtAudioPId = 0`, such as for an audio only unit that does not use PIDs, this object shall be the audio signal format of a single audio component.

If `aMtAudioStatus` is `true`, but no valid audio format can be identified, then the value `unspecifiedAudio` shall be returned.

If `aMtAudioStaus` is `false`, the value `noAudio` shall be returned.

6.4.1.8 aMtAudioPId

The programme identifier for this particular audio component. For units that do not use PIDs, such as audio only units, the value of `zero` shall be returned.

6.4.1.9 aMtIfIndex

The identifier for the associated network interfaces object within MIB-II. This object shall be a copy of `ifIndex` from RFC1213-MIB (MIB-II) (1.3.6.1.2.1.2.2.1.1) so as to provide a link between this audio component and the network interface on which it is present.

If `ifIndex` is not used within the unit, then an equivalent number to identify the interface should be used. This shall be > 0.

6.4.1.10 aMtFECType

An indication as to the FEC type applied, if present.

6.4.1.11 aMtFECLengthDimension

A description of the number of bytes over which FEC is applied and the matrix size being used. If the value of `aMtFECType` is `zero` (no FEC present), the value of `zero` shall be returned.

The format shall be `xxyy`, where, `xx` represents the number of bytes over which FEC is applied and `yy` represents the matrix size, for example, 2405 for 24 bytes and a 5x5 matrix.

6.5 Video measurement information blocks

6.5.1 Generic video measurement functionality

A video measurement information block shall have the following structure where c is the number of channels on a connection:

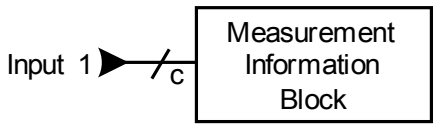


Figure 4 – Video Measurement Information block

A video measurement information block shall consist of a generic measurement information block containing information specific to video parameters that are to be available for use by a management agent.

NOTE This is a special type of block that has no internal functionality. It is merely a means to gather information from within the unit from, for example, other tables and objects or direct from the media stream and for it to be presented and used in a standardised way to a management system. This specification does not define where the information for the content of these blocks is sourced from.

The group of objects in Table 3 shall be implemented by all compliant equipment that has a management model that incorporates one or more measurement information block(s). The root node for these objects shall be

{ iso(1) standard(0) iec62379 measurement(7) measurementMIB(1) videoMeasurement(3) }

This node shall be used as the block type identifier for video measurement information blocks.

Table 3 - Managed objects for video measurement information blocks

Identifier	Syntax	Index	Readable	Writable	Volatile	Status
vMtBlockTable(1)	SEQUENCE OF VMtBlockEntry		none	none	no	m
└vMtBlockEntry(1)	VMtBlockEntry		none	none	no	m
├vMtBlockId(1)	BlockId	yes	none	none	no	m
├vMtAudioBlockId(2)	BlockId		listener	none	maybe	m
├vMtNetworkBlockId(3)	BlockId		listener	none	maybe	m
├vMtVideoStatus(4)	TruthValue		listener	none	yes	m
├vMtVideoSourceFormat(5)	MediaFormat		listener	none	yes	m
├vMtVideoCodingType(6)	MediaFormat		listener	none	yes	m
├vMtVideoBitRateType(7)	BitRateType		listener	none	yes	m
├vMtVideoBitRate(8)	CardinalNumber		listener	none	yes	m
├vMtAspectRatio(9)	MediaFormat		listener	none	yes	m
├vMtFECType(10)	VideoFECType		listener	none	yes	o
├vMtFECLengthDimension(11)	IntegerNumber		listener	none	maybe	o
└vMtTrickModeSupport(12)	TruthValue		listener	none	yes	o

6.5.1.1 vMtBlockTable

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

6.5.1.2 vMtBlockEntry

An entry in the video measurement block table.

6.5.1.3 vMtBlockId

The block identifier for this block. Used as an index when accessing the video measurement block table.

6.5.1.4 vMtAudioBlockId

The identifier for the associated audio measurement block. This object provides a link between the video measurement block and the audio measurement block.

6.5.1.5 vMtNetworkBlockId

The identifier for the associated network measurement block. This object provides a link between the video measurement block and the network measurement block.

6.5.1.6 vMtVideoStatus

The status of the video signal.

If `true`, indicates the video signal is present. If `false`, indicates the video signal is not present.

6.5.1.7 vMtVideoSourceFormat

A description of the structure of the base video source format.

The format is defined in IEC62397-3 Ed.1

It has the following format:

1.0.62379.3.2.1.2.w.x.y.z where

- w is the frame rate in Hz
- x is the definition of the video source type
  - Unspecified (0)
  - SD (1)
  - HD (2)
- y is the number of vertical lines of resolution
- z is the video scan type
  - Unspecified (0)
  - Progressive - P (1)
  - Interlaced - I (2)
  - Progressive Segmented Frame – PSF (3)

6.5.1.8 vMtVideoCodingType

A description of the current video signal coding type.

The format is defined in IEC62397-3 Ed.1

It has the following format:

1.0.62379.3.2.1.3.z where

- z is either uncompressed or the coding type
  - Unspecified (0)
  - Uncompressed (1)
  - MPEG2 (2)
  - H264 (3)
  - JPEG2000 (4)
  - SMPTE VC-2 (5)
  - VP8 (6)
  - H264 Scalable Extension (7)

6.5.1.9 vMtVideoBitRateType

A description of the current video bit rate type applied.

6.5.1.10 vMtVideoBitRate

A description of the current video bit rate in kbit/s.

If `vMtVideoBitRateType = Unspecified (0)`, then this object shall be set to a default value of `Unspecified (0)`

If `vMtVideoBitRateType` = `VBR (1)`, then this object shall be the advertised Maximum Bit Rate

If `vMtVideoBitRateType = CBR (2)`, then this object shall be the advertised (Constant) Bit Rate

6.5.1.11 vMtVideoAspectRatio

A description of the current video aspect ratio.

The format is defined in IEC62397-3 Ed.1

It has the following format:

1.0.62379.3.2.1.5.y.z where

- y is the source aspect ratio
  - Unspecified (0)
  - 4:3 (43)
  - 16:9 (169)
  - 2.21:1 (221)

Note: Unspecified (0) uses only the undefined AFD code 0000.

- z is the active format description code for the source aspect ratio
  - The codes are from 0000-1111
  - See SMPTE ST 2016-1:2009 for code descriptions.

6.5.1.12 vMtFECType

An indication as to the FEC type applied, if present.

6.5.1.13 vMtFECLengthDimension

A description of the number of bytes over which FEC is applied and the matrix size being used. If the value of `vMtFECType` is `zero` (no FEC present), the value of `zero` shall be returned.

The format shall be `xxyy`, where, `xx` represents the number of bytes over which FEC is applied and `yy` represents the matrix size, for example, 2405 for 24 bytes and a 5x5 matrix.

6.5.1.14 vMtTrickModeSupport

An indication as to whether trick mode is supported or not.

If `true`, indicates trick mode is supported. If `false`, indicates trick mode is not supported.

6.6 Receiver Point measurement information block

6.6.1 Generic receiver measurement functionality

A receiver point measurement information block shall have the following structure where c is the number of channels on a connection:

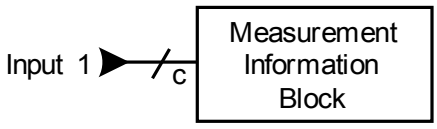


Figure 5 – Receiver Measurement Information Block

A receiver measurement information block shall consist of a generic measurement information block containing information specific to receiver parameters that are to be available for use by a management agent.

NOTE 1 This is a special type of block that has no internal functionality. It is merely a means to gather information from within the unit from, for example, other tables and objects or direct from the media stream and for it to be presented and used in a standardised way to a management system. This specification does not define where the information for the content of these blocks is sourced from.

The group of objects in Table 4shall be implemented by all compliant equipment that has a management model that incorporates one or more measurement information block(s). The root node for these objects shall be

{ iso(1) standard(0) iec62379 measurement(7) measurementMIB(1) receiverMeasurement(4) }

NOTE 2 This group of objects shall only exist if the value of `nMtTxRxPoint` is `true` in the corresponding row in the `nMtTable`.

This node shall be used as the block type identifier for receiver measurement information blocks.

Table 4 - Managed objects for receiver measurement information blocks

Identifier	Syntax	Index	Readable	Writable	Volatile	Status
`rxPointTable(2)`	`SEQUENCE OF RxPointEntry`		none	none	maybe	m
`└ rxPointEntry(1)`	`RxPointEntry`		none	none	maybe	m
`├ rxPointBlockId(1)`	`BlockId`	yes	none	none	maybe	m
`├ rxPointNetworkBlockId(2)`	`BlockId`	yes	none	none	maybe	m
`├ rxPointBufferSize(3)`	`BufferSize`		listener	operator	maybe	m
`├ rxPointBufferOcpancyTime(4)`	`BufferOcpncyTime`		listener	none	maybe	m
`├ rxPointBufferOcpncyPcnt(5)`	`BufferOcpncyPercent`		listener	none	maybe	m
`├ rxPointMDI(6)`	`Utf8String`		listener	none	maybe	o
`├ rxPointTSDF(7)`	`CardinalNumber`		listener	none	maybe	o

6.6.1.1 rxPointTable

A table of receiver point measurement block descriptors for this unit. Each receiver point measurement block in the unit has a corresponding entry in this table.

Entries in this table shall only exist if the value of `nMtTxRxPoint` is `true` in the corresponding row in the `nMtTable`.

The number of entries is between `zero` and the number of entries in the `nMtTable`. Since an entry in the `nMtTable` may not have the value of `true` for `nMtTxRxPoint`, there may not be an entry in this table for each entry in the `nMtTable`.

In addition to the index of `rxPointBlockId`, this table also uses the same index of `nMtBlockId` as the `nMtTable`.

6.6.1.2 rxPointEntry

An entry in the receiver point measurement block table.

6.6.1.3 rxPointBlockId

The block identifier for this block. Used as an index when accessing the receiver point measurement block table.

6.6.1.4 rxPointNetworkBlockId

The index value (`nMtBlockId`) from the Network Measurement table (`nMtTable`) used as an index when accessing the `rxPoint` measurement block table.

6.6.1.5 rxPointBufferSize

A description of the current total size of the receive buffer in ms.

6.6.1.6 rxPointBufferOcpancyTime

A description of the amount of data, in ms, occupying the receive buffer. This ranges between `zero` and `rxPointBufferSize`.

6.6.1.7 rxPointBufferOcpncyPcnt

A description of the amount of data, expressed as a percentage of the total receive buffer size, occupying the receive buffer.

6.6.1.8 rxPointMDI

A description of the measured and calculated Media Delivery Index (MDI). The MDI has two components, the Delay Factor (DF) and the Media Loss Rate (MLR).

MDI is expressed as simple pairs of values in the form of DF:MLR, e.g. 06:10

The measurement units of DF and MLR are ms.

6.6.1.9 rxPointTSDF

A description of the Time-stamped Delay Factor (TS-DF).

Time-stamped Delay Factor is calculated as:

TS-DF = D(Max) - D(Min)

The measurement units of TS and DF are ms.

For further details of this, see EBU TECH3337

6.7 Temperature measurement information block

6.7.1 Generic temperature measurement functionality

A temperature measurement information block shall have the following structure where c is the number of channels on a connection:

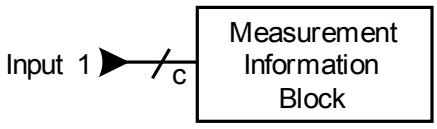


Figure 6 – Temperature Measurement Information Block

A temperature measurement information block shall consist of a generic measurement information block containing information specific to temperature parameters that are to be available for use by a management agent.

NOTE 1 This is a special type of block that has no internal functionality. It is merely a means to gather information from within the unit from, for example, other tables and objects or direct from the media stream and for it to be presented and used in a standardised way to a management system. This specification does not define where the information for the content of these blocks is sourced from.

The group of objects in Table 5 shall be implemented by all compliant equipment that has a management model that incorporates one or more measurement information block(s). The root node for these objects shall be:

{ iso(1) standard(0) iec62379 measurement(7) measurementMIB(1) temperatureMeasurement(5) }

Note 2 This group of objects are optional, but mandatory if temperature sensors are present within the unit.

Table 5 - Managed objects conveying temperature information about the unit

Identifier	Syntax	Index	Readable	Writable	Volatile	Status
temperatureTable(1)	SEQUENCE OF TemperatureEntry		none	none	no	m
└ temperatureEntry(1)	TemperatureEntry		none	none	no	m
├ temperatureBlockId(1)	BlockId	yes	none	none	no	m
├ temperatureLocnNumber(2)	IndexNumber	yes	none	none	no	m
├ temperatureLocn(3)	TemperatureLocn		listener	none	no	m
├ temperatureTrend(4)	TemperatureTrend		listener	none	yes	m
├ temperatureStatus(5)	TemperatureStatus		listener	none	yes	m
├ temperatureLowWarning(6)	IntegerNumber		listener	none	yes	m
├ temperatureHighWarning(7)	IntegerNumber		listener	none	yes	m
├ temperatureLowCritical(8)	IntegerNumber		listener	none	yes	m
└ temperatureHighCritical(9)	IntegerNumber		listener	none	yes	m

6.7.1.1 temperatureTable

A table of temperature measurement descriptors. Each temperature measurement point in the unit has an entry in this table.

6.7.1.2 temperatureEntry

An entry in the temperature measurement table.

6.7.1.3 temperatureBlockId

The block identifier for this block. Used as an index when accessing the temperature measurement block table.

6.7.1.4 temperatureLocnNumber

The identifier where the temperature is measured. Used as an index when accessing the unit temperature measurement table. Each temperature measurement point in a unit has a unique number.

NOTE Temperature measurement location numbers should be allocated sequentially, starting from 1.

6.7.1.5 temperatureLocation

A text description of the location of where the temperature is being measured.

6.7.1.6 temperatureTrend

The current temperature measured at this location. Unit is Degrees Celsius.

6.7.1.7 temperatureStatus

The current status of this particular temperature measurement.

– `undetermined` indicates the temperature cannot be determined
– `other` indicates the temperature is in some other state
– `unknown` indicates the temperature is unknown
– `ok` indicates the temperature is within acceptable limits
– `warning` indicates the temperature has reached the warning level for the unit
– `critical` indicates the temperature has reached the critical level for the unit
– `non-recoverable` indicates the temperature level for the unit is no longer recoverable

6.7.1.8 temperatureLowWarning

The low warning limit for the measured temperature. Unit is Degrees Celsius.

6.7.1.9 temperatureHighWarning

The high warning limit for the measured temperature. Unit is Degrees Celsius.

6.7.1.10 temperatureLowCritical

The low critical limit for the measured temperature. Unit is Degrees Celsius.

6.7.1.11 temperatureHighCritical

The high critical limit for the measured temperature. Unit is Degrees Celsius.

Annex A
(informative)

Machine-readable measurement block definitions

This annex provides a machine-readable version of the measurement 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 6, clause 6 takes precedence.

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

```
   IEC62379-7-IPM-MIB DEFINITIONS ::= BEGIN

        IMPORTS
             iec62379, unitPowerSource, psNumber, psType, psStatus,
             psChargeLevel, psChargeTime, blockId, blockType, BlockId,
             BlockType, CardinalNumber, ClockSource, IndexNumber, IntegerNumber,
             MediaFormat, Utf8String
                  FROM IEC62379-1-MIB
             NetworkType,   TransportType,   TemperatureStatus,   TemperatureLocn,
TemperatureTrend,
             AudioFECType,    VideoFECType,    BufferSize,    BufferOcpncyPercent,
BufferOcpncyTime,
             BitRateType
                  FROM IEC62379-7-IPM-TC-MIB
             InterfaceIndex
                  FROM IF-MIB
             OBJECT-GROUP, MODULE-COMPLIANCE
                  FROM SNMPv2-CONF
             OBJECT-TYPE, MODULE-IDENTITY
                  FROM SNMPv2-SMI
             TAddress, TruthValue
                  FROM SNMPv2-TC;


        -- 1.0.62379.7.1
        measurementMIB MODULE-IDENTITY
             LAST-UPDATED "201106211200Z"        -- June 21, 2011 at 12:00 GMT
             ORGANIZATION
                  "IEC PT62379"
             CONTACT-INFO
                  "Not specified."
             DESCRIPTION
                  "The MIB module for managing measurement functions
                  in IEC 62379 compliant equipment.
                  The contents of this MIB Module have been defined by
                  the EBU ECN-IPM (European Broadcasting Union
                  Expert Communities Networks, IP Measurements) group."
             REVISION "201106211200Z"       -- June 21, 2011 at 12:00 GMT
             DESCRIPTION
                  "Added   additional   coding   types   VP8   and   H.264   Scalable
Extension
                  entries.
                  Added video bit rate type and rate.
                  Addition of Aspect Ratio to the Video Measurement Table.
                  Move FEC information down to accommodate Aspect Ratio."
             ::= { measurement 1 }




   --
```

```
-- Node definitions
--

        -- 1.0.62379.7
        measurement OBJECT IDENTIFIER ::= { iec62379 7 }


        -- 1.0.62379.7.1.0
        measurementMIBCompliance OBJECT IDENTIFIER ::= { measurementMIB 0 }


        -- 1.0.62379.7.1.0.1
        measurementMIBComplianceV1 MODULE-COMPLIANCE
             STATUS current
             DESCRIPTION
                  " The compliance statement for entities that conform to
                  IEC 62379-7 (2010)."
             MODULE -- this module
                  MANDATORY-GROUPS            {            audioMeasurementGroup,
networkMeasurementGroup, receiverMeasurementGroup }
                  GROUP videoMeasurementGroup
                       DESCRIPTION
                            "Mandatory    for    equipment    that    has    video
functionality."
                  GROUP temperatureMeasurementGroup
                       DESCRIPTION
                            "Mandatory   for   equipment   that   has   temperature
measurement functionality."
             ::= { measurementMIBCompliance 1 }


        -- 1.0.62379.7.1.1
        networkMeasurement OBJECT IDENTIFIER ::= { measurementMIB 1 }


        -- 1.0.62379.7.1.1.1
        nMtTable OBJECT-TYPE
             SYNTAX SEQUENCE OF NMtEntry
             MAX-ACCESS not-accessible
             STATUS current
             DESCRIPTION
                  "Description."
             ::= { networkMeasurement 1 }


        -- 1.0.62379.7.1.1.1.1
        nMtEntry OBJECT-TYPE
             SYNTAX NMtEntry
             MAX-ACCESS not-accessible
             STATUS current
             DESCRIPTION
                  "Description."
             INDEX { nMtBlockId }
             ::= { nMtTable 1 }


        NMtEntry ::=
             SEQUENCE {
                  nMtBlockId
                       BlockId,
                  nMtIfIndex
                       InterfaceIndex,
                  nMtTxRxPoint
                       TruthValue,
                  nMtNetworkType
                       NetworkType,
                  nMtTransportType
                       TransportType,
                  nMtTxRxAddr
                       TAddress,
```

```
            nMtPortNumber
                 CardinalNumber,
            nMtIGMPVersion
                 CardinalNumber,
            nMtSIPServerAddr
                 TAddress
       }

-- 1.0.62379.7.1.1.1.1.1
nMtBlockId OBJECT-TYPE
     SYNTAX BlockId
     MAX-ACCESS not-accessible
     STATUS current
     DESCRIPTION
          "The block identifier for this block.
          Used as an index when accessing the network measurement
          block table."
     ::= { nMtEntry 1 }


-- 1.0.62379.7.1.1.1.1.2
nMtIfIndex OBJECT-TYPE
     SYNTAX InterfaceIndex
     MAX-ACCESS read-only
     STATUS current
     DESCRIPTION
          "The identifier for the associated network interfaces object
          within MIB-II.

          This object shall be a copy of ifIndex (actually Textual
          Convention InterfaceIndex - see below from IF-MIB) from
          RFC1213-MIB (MIB-II) (1.3.6.1.2.1.2.2.1.1) so as to provide
          a link between this network measurement block and the
          network interface it is associated with.

          If ifIndex is not used within the unit, then an equivalent
          number to identify the interface should be used.
          This shall be greater than zero.

          (Description of ifIndex from RFC1213-MIB).
          A unique value for each interface.  Its value
          ranges between 1 and the value of ifNumber (1.3.6.1.2.1.2.1).
          The value for each interface must remain constant at
          least from one re-initialization of the entity's
          network management system to the next re-
          initialization.

          (Note about and Description of Textual Convention
          InterfaceIndex from IF-MIB).

          InterfaceIndex contains the semantics of ifIndex and
          should be used for any objects defined in other MIB modules
          that need these semantics.

          A unique value, greater than zero, for each interface or
          interface sub-layer in the managed system.  It is
          recommended that values are assigned contiguously starting
          from 1.  The value for each interface sub-layer must remain
          constant at least from one re-initialization of the entity's
          network management system to the next re-initialization."
     ::= { nMtEntry 2 }


-- 1.0.62379.7.1.1.1.1.3
nMtTxRxPoint OBJECT-TYPE
     SYNTAX TruthValue
     MAX-ACCESS read-write
     STATUS current
     DESCRIPTION
          "This represents the measurement position in the media chain;
```

```
                    whether the unit is a transmitter or receiver of the
                    media.

                    In the case of bi-directional media flow, then this value
                    should be set to indicate the position from where the
                    measurement is required to be made; from a media
                    transmission or reception point of view.

                    Transmitter (Tx) = false
                    Receiver (Rx) = true

                    If the value of this object is true (Rx), then the receiver
table
                    (rxPointTable) shall exist, otherwise its existence is
optional."
               ::= { nMtEntry 3 }


          -- 1.0.62379.7.1.1.1.1.4
          nMtNetworkType OBJECT-TYPE
               SYNTAX NetworkType
               MAX-ACCESS read-only
               STATUS current
               DESCRIPTION
                    "A description of the current network type over which the
                    media flow is flowing.

                    Current Values are:
                    ipv4(1),
                    ipv6(2),
                    asi(3)

                    If the network type is asi(3), then the remaining entries
                    of

                    nMtTransportType
                    nMtTxRxAddr
                    nMtPortNumber
                    nMtIGMPVersion
                    nMtSIPServerAddr

                    in this table are not required."
               ::= { nMtEntry 4 }


          -- 1.0.62379.7.1.1.1.1.5
          nMtTransportType OBJECT-TYPE
               SYNTAX TransportType
               MAX-ACCESS read-only
               STATUS current
               DESCRIPTION
                    "A description of the transport type being used to convey the
                    media.

                    If nMtNetworkType is asi (3), then notApplicable(0) shall be
                    returned.

                    Current Values are:

                    notApplicable(0),
                    rtp(1)"
               ::= { nMtEntry 5 }


          -- 1.0.62379.7.1.1.1.1.6
          nMtTxRxAddr OBJECT-TYPE
               SYNTAX TAddress
               MAX-ACCESS read-only
               STATUS current
               DESCRIPTION
```

```
                    "The network address of either the send or receive point
                    network interface over which the media is flowing.
                    Which it is, is dependent on the measurement position in
                    the media chain; whether the unit is a transmitter or
                    receiver of the media.
                    The value of nMtTxRxPoint shall determine which end is
                    being referred to.

                    If nMtNetworkType is asi (3), then NULL shall be returned."
               ::= { nMtEntry 6 }


          -- 1.0.62379.7.1.1.1.1.7
          nMtPortNumber OBJECT-TYPE
               SYNTAX CardinalNumber
               MAX-ACCESS read-only
               STATUS current
               DESCRIPTION
                    "The port number assigned to the port over which the media
                    stream or other application of interest is being measured.

                    A port number is a 16-bit unsigned integer, ranging from
                    0 to 65535.

                    If nMtNetworkType is asi (3), then zero shall be returned."
               ::= { nMtEntry 7 }


          -- 1.0.62379.7.1.1.1.1.8
          nMtIGMPVersion OBJECT-TYPE
               SYNTAX CardinalNumber
               MAX-ACCESS read-only
               STATUS current
               DESCRIPTION
                    "The version of Internet Group Management Protocol (IGMP)
                    being used.

                    If nMtNetworkType is asi (3), then zero shall be returned.

                    If IGMP is not supported, then zero shall be returned."
               ::= { nMtEntry 8 }


          -- 1.0.62379.7.1.1.1.1.9
          nMtSIPServerAddr OBJECT-TYPE
               SYNTAX TAddress
               MAX-ACCESS read-write
               STATUS current
               DESCRIPTION
                    "The IP address of the SIP server which the unit may be
                    registered with.

                    If nMtNetworkType is asi (3), then NULL shall be returned.

                    If the unit is not registered with a SIP server, then NULL
                    shall be returned."
               ::= { nMtEntry 9 }


          -- 1.0.62379.7.1.1.2
          networkMeasurementGroup OBJECT-GROUP
               OBJECTS     {     nMtIfIndex,     nMtTxRxPoint,     nMtNetworkType,
nMtTransportType, nMtTxRxAddr,
                    nMtPortNumber, nMtIGMPVersion, nMtSIPServerAddr }
               STATUS current
               DESCRIPTION
                    "The group of objects used for network measurements."
               ::= { networkMeasurement 2 }
```

```
-- 1.0.62379.7.1.2
audioMeasurement OBJECT IDENTIFIER ::= { measurementMIB 2 }


-- 1.0.62379.7.1.2.1
aMtBlockTable OBJECT-TYPE
     SYNTAX SEQUENCE OF AMtBlockEntry
     MAX-ACCESS not-accessible
     STATUS current
     DESCRIPTION
          "Description."
     ::= { audioMeasurement 1 }


-- 1.0.62379.7.1.2.1.1
aMtBlockEntry OBJECT-TYPE
     SYNTAX AMtBlockEntry
     MAX-ACCESS not-accessible
     STATUS current
     DESCRIPTION
          "Description."
     INDEX { aMtBlockId, aMtAudioComponentNumber }
     ::= { aMtBlockTable 1 }


AMtBlockEntry ::=
     SEQUENCE {
          aMtBlockId
               BlockId,
          aMtAudioComponentNumber
               IndexNumber,
          aMtNetworkBlockId
               BlockId,
          aMtAudioStatus
               TruthValue,
          aMtAudioSignalFormat
               MediaFormat,
          aMtAudioPId
               CardinalNumber,
          aMtIfIndex
               InterfaceIndex,
          aMtFECType
               AudioFECType,
          aMtFECLengthDimension
               IntegerNumber
      }

-- 1.0.62379.7.1.2.1.1.1
aMtBlockId OBJECT-TYPE
     SYNTAX BlockId
     MAX-ACCESS not-accessible
     STATUS current
     DESCRIPTION
          "The block identifier for this block.
          Used as an index when accessing the audio measurement
          block table."
     ::= { aMtBlockEntry 1 }


-- 1.0.62379.7.1.2.1.1.2
aMtAudioComponentNumber OBJECT-TYPE
     SYNTAX IndexNumber
     MAX-ACCESS not-accessible
     STATUS current
     DESCRIPTION
          "The audio component identifier.
          The audio component number is the identifier for each audio
          item, when there are one or more audio items associated
          with a single video stream/signal.
```

```
                    For an audio only unit this identifier shall also be
                    used to identify each audio item, when there are one or more
                    audio items present within the unit.

                    Used as an index when accessing the audio measurement
                    block table."
               ::= { aMtBlockEntry 2 }


          -- 1.0.62379.7.1.2.1.1.3
          aMtNetworkBlockId OBJECT-TYPE
               SYNTAX BlockId
               MAX-ACCESS read-only
               STATUS current
               DESCRIPTION
                    "The identifier for the associated network measurement block.
                    This provides a link between the audio measurement block
                    and the network measurement block."
               ::= { aMtBlockEntry 3 }


          -- 1.0.62379.7.1.2.1.1.4
          aMtAudioStatus OBJECT-TYPE
               SYNTAX TruthValue
               MAX-ACCESS read-only
               STATUS current
               DESCRIPTION
                    "The status of the audio for this component number.

                    If true, indicates the audio signal is present.
                    If false, indicates the audio signal is not present."
               ::= { aMtBlockEntry 4 }


          -- 1.0.62379.7.1.2.1.1.5
          aMtAudioSignalFormat OBJECT-TYPE
               SYNTAX MediaFormat
               MAX-ACCESS read-only
               STATUS current
               DESCRIPTION
                    "A description of the current audio signal format.

                    If aMtAudioPId > 0, this should be the audio signal format
                    associated with this PID.

                    If aMTAudioPID = 0, such as for an audio only device that does
                    not use PIDs, this object shall be the audio signal format of
a
                    single audio component.

                    If aMtAudioStatus is true, but no valid audio format can be
                    identified, then the value unspecifiedAudio shall be returned.

                    If  aMtAudioStaus  is  false,  the  value  noAudio  shall  be
returned."
               ::= { aMtBlockEntry 5 }


          -- 1.0.62379.7.1.2.1.1.6
          aMtAudioPId OBJECT-TYPE
               SYNTAX CardinalNumber
               MAX-ACCESS read-only
               STATUS current
               DESCRIPTION
                    "The programme identifier for this particular audio component.

                    For units that do not use PIDs, such as audio only units,
                    the value of zero shall be returned."
               ::= { aMtBlockEntry 6 }
```

```
        -- 1.0.62379.7.1.2.1.1.7
        aMtIfIndex OBJECT-TYPE
             SYNTAX InterfaceIndex
             MAX-ACCESS read-only
             STATUS current
             DESCRIPTION
                  "The identifier for the associated network interfaces object
                  within MIB-II.

                  This object shall be a copy of ifIndex from RFC1213-MIB (MIB-
II)
                  (1.3.6.1.2.1.2.2.1.1) so as to provide a link between this
                  audio component and the network interface on which
                  it is present.

                  If ifIndex is not used within the unit, then an equivalent
                  number to identify the interface should be used.
                  This shall be > 0.

                  (Description of ifIndex from RFC1213-MIB).
                  A unique value for each interface.  Its value
                  ranges between 1 and the value of ifNumber (1.3.6.1.2.1.2.1).
                  The value for each interface must remain constant at
                  least from one re-initialization of the entity's
                  network management system to the next re-
                  initialization.

                  (Description of InterfaceIndex from IF-MIB).
                  A unique value, greater than zero, for each interface or
                  interface sub-layer in the managed system.  It is
                  recommended that values are assigned contiguously starting
                  from 1.  The value for each interface sub-layer must remain
                  constant at least from one re-initialization of the entity's
                  network management system to the next re-initialization."
             ::= { aMtBlockEntry 7 }


        -- 1.0.62379.7.1.2.1.1.8
        aMtFECType OBJECT-TYPE
             SYNTAX AudioFECType
             MAX-ACCESS read-only
             STATUS current
             DESCRIPTION
                  "An indication as to the FEC type applied, if present.

                  Current Values are:

                  none(0),
                  smpte2021(1),
                  smpte2022(2),
                  rfc2733(3),
                  proprietary(4)"
             ::= { aMtBlockEntry 8 }


        -- 1.0.62379.7.1.2.1.1.9
        aMtFECLengthDimension OBJECT-TYPE
             SYNTAX IntegerNumber
             MAX-ACCESS read-only
             STATUS current
             DESCRIPTION
                  "A description of the number of bytes over which FEC is
                  applied and the matrix size being used.

                  If the value of aMtFECType is zero (no FEC present), the
                  value of zero shall be returned.

                  The format shall be xxyy, where, xx represents the number
                  of bytes over which FEC is applied and yy represents the
```

```
                    matrix size, for example, 2405 for 24 bytes and a 5x5 matrix.
                    "
               ::= { aMtBlockEntry 9 }


          -- 1.0.62379.7.1.2.2
          audioMeasurementGroup OBJECT-GROUP
               OBJECTS { aMtNetworkBlockId, aMtAudioStatus, aMtAudioSignalFormat,
  aMtAudioPId, aMtIfIndex,
                    aMtFECType, aMtFECLengthDimension }
               STATUS current
               DESCRIPTION
                    "The group of objects used for audio measurements."
               ::= { audioMeasurement 2 }


          -- 1.0.62379.7.1.3
          videoMeasurement OBJECT IDENTIFIER ::= { measurementMIB 3 }


          -- 1.0.62379.7.1.3.1
          vMtTable OBJECT-TYPE
               SYNTAX SEQUENCE OF VMtEntry
               MAX-ACCESS not-accessible
               STATUS current
               DESCRIPTION
                    "Description."
               ::= { videoMeasurement 1 }


          -- 1.0.62379.7.1.3.1.1
          vMtEntry OBJECT-TYPE
               SYNTAX VMtEntry
               MAX-ACCESS not-accessible
               STATUS current
               DESCRIPTION
                    "Description."
               INDEX { vMtBlockId }
               ::= { vMtTable 1 }


          VMtEntry ::=
               SEQUENCE {
                    vMtBlockId
                         BlockId,
                    vMtAudioBlockId
                         BlockId,
                    vMtNetworkBlockId
                         BlockId,
                    vMtVideoStatus
                         TruthValue,
                    vMtVideoSourceFormat
                         MediaFormat,
                    vMtVideoCodingType
                         MediaFormat,
                    vMtVideoBitRateType
                         BitRateType,
                    vMtVideoBitRate
                         CardinalNumber,
                    vMtAspectRatio
                         MediaFormat,
                    vMtFECType
                         VideoFECType,
                    vMtFECLengthDimension
                         IntegerNumber,
                    vMtTrickModeSupport
                         TruthValue
                }

          -- 1.0.62379.7.1.3.1.1.1
```

```
        vMtBlockId OBJECT-TYPE
             SYNTAX BlockId
             MAX-ACCESS not-accessible
             STATUS current
             DESCRIPTION
                  "The block identifier for this block.
                  Used as an index when accessing the video measurement
                  block table."
             ::= { vMtEntry 1 }


        -- 1.0.62379.7.1.3.1.1.2
        vMtAudioBlockId OBJECT-TYPE
             SYNTAX BlockId
             MAX-ACCESS read-only
             STATUS current
             DESCRIPTION
                  "The identifier for the associated audio measurement block.
                  This object provides a link between the video measurement
                  block and the audio measurement block."
             ::= { vMtEntry 2 }


        -- 1.0.62379.7.1.3.1.1.3
        vMtNetworkBlockId OBJECT-TYPE
             SYNTAX BlockId
             MAX-ACCESS read-only
             STATUS current
             DESCRIPTION
                  "The identifier for the associated network measurement block.
                  This object provides a link between the video measurement
                  block and the network measurement block."
             ::= { vMtEntry 3 }


        -- 1.0.62379.7.1.3.1.1.4
        vMtVideoStatus OBJECT-TYPE
             SYNTAX TruthValue
             MAX-ACCESS read-only
             STATUS current
             DESCRIPTION
                  "The status of the video signal.

                  If true, indicates the video signal is present.
                  If false, indicates the video signal is not present."
             ::= { vMtEntry 4 }


        -- 1.0.62379.7.1.3.1.1.5
        vMtVideoSourceFormat OBJECT-TYPE
             SYNTAX MediaFormat
             MAX-ACCESS read-only
             STATUS current
             DESCRIPTION
                  "A description of the structure of the base video source
format.

                  The format is defined in IEC62397-3 Ed.1
                  It has the following format:

                  1.0.62379.3.2.1.3.w.x.y.z where

                  w is the frame rate in Hz

                  x is the definition of the video source type
                  Unspecified (0),
                  Standard Definition - SD (1),
                  High Definition - HD (2)

                  y is the number of vertical lines of resolution
```

```
            z is the video scan type
            Unspecified (0),
            Progressive (1),
            Interlaced (2),
            Progressive Segmented Frame - PSF (3)"
      ::= { vMtEntry 5 }


-- 1.0.62379.7.1.3.1.1.6
vMtVideoCodingType OBJECT-TYPE
      SYNTAX MediaFormat
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
            "A description of the current video signal coding type.

            The format is defined in IEC62397-3 Ed.1
            It has the following format:

            1.0.62379.3.2.1.4.z where

            z is either uncompressed or the coding type

            Unspecified (0),
            Uncompressed (1),
            MPEG2 (2),
            H264 (3),
            JPEG2000 (4),
            SMPTE VC-2 (5),
            VP8 (6),
            H264ScaleExtn (7)"
      ::= { vMtEntry 6 }


-- 1.0.62379.7.1.3.1.1.7
vMtVideoBitRateType OBJECT-TYPE
      SYNTAX BitRateType
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
            "A description of the current video bit rate type applied.

            unspecified (0),
            Variable Bit Rate - VBR (1),
            Constant Bite Rate - CBR (2)"
      ::= { vMtEntry 7 }


-- 1.0.62379.7.1.3.1.1.8
vMtVideoBitRate OBJECT-TYPE
      SYNTAX CardinalNumber
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
            "A description of the current video bit rate in kbps.

            If vMtVideoBitRateType = Unspecified (0), then
            this object shall be set to a default value of Unspecified (0)

            If vMtVideoBitRateType = VBR (1),
            then this object shall be the advertised Maximum Bit Rate

            If vMtVideoBitRateType = CBR (2),
            then this object shall be the advertised (Constant) Bit Rate"
      ::= { vMtEntry 8 }


-- 1.0.62379.7.1.3.1.1.9
vMtAspectRatio OBJECT-TYPE
```

```
            SYNTAX MediaFormat
            MAX-ACCESS read-only
            STATUS current
            DESCRIPTION
                 "A description of the current video aspect ratio.

                 The format is defined in IEC62397-3 Ed.1
                 It has the following format:

                 1.0.62379.3.2.1.5.y.z where

                 y is the source aspect ratio
                 4:3  (43),
                 16:9 (169),
                 2.21:1 (221)

                 z is the active format description code for the source aspect
ratio.
                 The codes are from 0000-1111
                 See SMPTE ST 2016-1:2009 for code descriptions."
            ::= { vMtEntry 9 }


        -- 1.0.62379.7.1.3.1.1.10
        vMtFECType OBJECT-TYPE
            SYNTAX VideoFECType
            MAX-ACCESS read-only
            STATUS current
            DESCRIPTION
                 "An indication as to the FEC type applied, if present.

                 Current Values are:

                 none(0),
                 smpte2022(2),
                 proprietary(4)"
            ::= { vMtEntry 10 }


        -- 1.0.62379.7.1.3.1.1.11
        vMtFECLengthDimension OBJECT-TYPE
            SYNTAX IntegerNumber
            MAX-ACCESS read-only
            STATUS current
            DESCRIPTION
                 "A description of the number of bytes over which FEC is
                 applied and the matrix size being used. If the value of
                 vMtFECType is zero (no FEC present), the value of zero
                 shall be returned.

                 The format shall be xxyy, where, xx represents the number
                 of bytes over which FEC is applied and yy represents the
                 matrix size, for example, 2405 for 24 bytes and a 5x5 matrix."
            ::= { vMtEntry 11 }


        -- 1.0.62379.7.1.3.1.1.12
        vMtTrickModeSupport OBJECT-TYPE
            SYNTAX TruthValue
            MAX-ACCESS read-only
            STATUS current
            DESCRIPTION
                 "An indication as to whether trick mode is supported or not."
            ::= { vMtEntry 12 }


        -- 1.0.62379.7.1.3.2
        videoMeasurementGroup OBJECT-GROUP
            OBJECTS   {   vMtAudioBlockId,   vMtNetworkBlockId,   vMtVideoStatus,
vMtVideoSourceFormat, vMtVideoCodingType,
```

```
                    vMtVideoBitRateType,      vMtVideoBitRate,      vMtAspectRatio,
vMtFECType, vMtFECLengthDimension,
                    vMtTrickModeSupport }
               STATUS current
               DESCRIPTION
                    "The group of objects used for video measurements."
               ::= { videoMeasurement 2 }


          -- 1.0.62379.7.1.4
          receiverMeasurement OBJECT IDENTIFIER ::= { measurementMIB 4 }


          -- 1.0.62379.7.1.4.2
          rxPointTable OBJECT-TYPE
               SYNTAX SEQUENCE OF RxPointEntry
               MAX-ACCESS not-accessible
               STATUS current
               DESCRIPTION
                    "Entries in this table shall only exist if the value of
nMtTxRxPoint
                    is TRUE in the corresponding row in the nMtTable.

                    The number of entries is between zero and the number of
entries in the
                    nMtTable. Since an entry in the nMtTable may not have the
value of
                    TRUE for nMtTxRxPoint, there may not be an entry in this
                    table for each entry in the nMtTable.

                    In addition to the index of rxPointBlockId, this table also
uses the
                    same index of nMtBlockId as the nMtTable."
               ::= { receiverMeasurement 2 }


          -- 1.0.62379.7.1.4.2.1
          rxPointEntry OBJECT-TYPE
               SYNTAX RxPointEntry
               MAX-ACCESS not-accessible
               STATUS current
               DESCRIPTION
                    "A row in the receiver point table. Entries cannot
                    be created or deleted with SNMP operations."
               INDEX { rxPointBlockId, rxPointNetworkBlockId }
               ::= { rxPointTable 1 }


          RxPointEntry ::=
               SEQUENCE {
                    rxPointBlockId
                         BlockId,
                    rxPointNetworkBlockId
                         BlockId,
                    rxPointBufferSize
                         BufferSize,
                    rxPointBufferOcpancyTime
                         BufferOcpncyTime,
                    rxPointBufferOcpncyPcnt
                         BufferOcpncyPercent,
                    rxPointMDI
                         Utf8String,
                    rxPointTSDF
                         CardinalNumber
                }

          -- 1.0.62379.7.1.4.2.1.1
          rxPointBlockId OBJECT-TYPE
               SYNTAX BlockId
               MAX-ACCESS not-accessible
```

```
                STATUS current
                DESCRIPTION
                     "The block identifier for this block.
                     Used as an index when accessing the receiver point
                     measurement block table."
                ::= { rxPointEntry 1 }


          -- 1.0.62379.7.1.4.2.1.2
          rxPointNetworkBlockId OBJECT-TYPE
                SYNTAX BlockId
                MAX-ACCESS not-accessible
                STATUS current
                DESCRIPTION
                     "The index value (nMtBlockId) from the Network Measurement
                     table (nMtTable) used as an index when accessing the rxPoint
                     measurement block table."
                ::= { rxPointEntry 2 }


          -- 1.0.62379.7.1.4.2.1.3
          rxPointBufferSize OBJECT-TYPE
                SYNTAX BufferSize
                UNITS "msecs"
                MAX-ACCESS read-only
                STATUS current
                DESCRIPTION
                     "A description of the current total size of the receive buffer
                     in ms."
                ::= { rxPointEntry 3 }


          -- 1.0.62379.7.1.4.2.1.4
          rxPointBufferOcpancyTime OBJECT-TYPE
                SYNTAX BufferOcpncyTime
                UNITS "msecs"
                MAX-ACCESS read-only
                STATUS current
                DESCRIPTION
                     "A description of the amount of data, in ms, occupying the
                     receive buffer.
                     This ranges between zero and rxPointBufferSize."
                ::= { rxPointEntry 4 }


          -- 1.0.62379.7.1.4.2.1.5
          rxPointBufferOcpncyPcnt OBJECT-TYPE
                SYNTAX BufferOcpncyPercent
                MAX-ACCESS read-write
                STATUS current
                DESCRIPTION
                     "A description of the amount of data, expressed as a
                     percentage    of    the    total    receive    buffer    size,
rxPointBufferSize,
                     occupying the receive buffer."
                ::= { rxPointEntry 5 }


          -- 1.0.62379.7.1.4.2.1.6
          rxPointMDI OBJECT-TYPE
                SYNTAX Utf8String
                UNITS "msecs"
                MAX-ACCESS read-only
                STATUS current
                DESCRIPTION
                     "A description of the measured and calculated Media Delivery
                     Index (MDI).
                     The MDI has two components, the Delay Factor (DF) and the
                     Media Loss Rate (MLR).
                     MDI is expressed as simple pairs of absolute values in the
```

```
                    form of DF:MLR, e.g. 06:10"
               ::= { rxPointEntry 6 }


          -- 1.0.62379.7.1.4.2.1.7
          rxPointTSDF OBJECT-TYPE
               SYNTAX CardinalNumber
               UNITS "msecs"
               MAX-ACCESS read-only
               STATUS current
               DESCRIPTION
                    "A description of the Time-stamped Delay Factor (TS-DF).
                    Time-stamped Delay Factor is calculated as:
                    TS-DF = D(Max) - D(Min)
                    For further details of this, see EBU TECH3337"
               ::= { rxPointEntry 7 }


          -- 1.0.62379.7.1.4.3
          receiverMeasurementGroup OBJECT-GROUP
               OBJECTS     {     rxPointBufferSize,     rxPointBufferOcpancyTime,
rxPointBufferOcpncyPcnt, rxPointMDI, rxPointTSDF
                    }
               STATUS current
               DESCRIPTION
                    "The group of objects used for receiver measurements."
               ::= { receiverMeasurement 3 }


          -- 1.0.62379.7.1.5
          temperatureMeasurement OBJECT IDENTIFIER ::= { measurementMIB 5 }


          -- 1.0.62379.7.1.5.1
          temperatureTable OBJECT-TYPE
               SYNTAX SEQUENCE OF TemperatureEntry
               MAX-ACCESS not-accessible
               STATUS current
               DESCRIPTION
                    "Information describing the temperature (if applicable)
                    measured   within   the   unit   at   various   locations   (if
applicable)."
               ::= { temperatureMeasurement 1 }


          -- 1.0.62379.7.1.5.1.1
          temperatureEntry OBJECT-TYPE
               SYNTAX TemperatureEntry
               MAX-ACCESS not-accessible
               STATUS current
               DESCRIPTION
                    "Description."
               INDEX { temperatureBlockId, temperatureLocnNumber }
               ::= { temperatureTable 1 }


          TemperatureEntry ::=
               SEQUENCE {
                    temperatureBlockId
                         BlockId,
                    temperatureLocnNumber
                         IndexNumber,
                    temperatureLocation
                         TemperatureLocn,
                    temperatureTrend
                         TemperatureTrend,
                    temperatureStatus
                         TemperatureStatus,
                    temperatureLowWarning
                         IntegerNumber,
```

```
                    temperatureHighWarning
                         IntegerNumber,
                    temperatureLowCritical
                         IntegerNumber,
                    temperatureHighCritical
                         IntegerNumber
               }

          -- 1.0.62379.7.1.5.1.1.1
          temperatureBlockId OBJECT-TYPE
               SYNTAX BlockId
               MAX-ACCESS not-accessible
               STATUS current
               DESCRIPTION
                    "The block identifier for this block.
                    Used as an index when accessing the temperature
                    measurement block table."
               ::= { temperatureEntry 1 }


          -- 1.0.62379.7.1.5.1.1.2
          temperatureLocnNumber OBJECT-TYPE
               SYNTAX IndexNumber
               MAX-ACCESS not-accessible
               STATUS current
               DESCRIPTION
                    "The identifier where the temperature is measured.
                    Used as an index when accessing the unit temperature
                    measurement table.
                    Each temperature measurement point in a unit has a unique
                    number.
                    Temperature measurement location numbers should be
                    allocated sequentially, starting from 1."
               ::= { temperatureEntry 2 }


          -- 1.0.62379.7.1.5.1.1.3
          temperatureLocation OBJECT-TYPE
               SYNTAX TemperatureLocn
               MAX-ACCESS read-only
               STATUS current
               DESCRIPTION
                    "A text description of the location of where the temperature
is
                    being measured."
               ::= { temperatureEntry 3 }


          -- 1.0.62379.7.1.5.1.1.4
          temperatureTrend OBJECT-TYPE
               SYNTAX TemperatureTrend
               UNITS "Degrees Celsius"
               MAX-ACCESS read-only
               STATUS current
               DESCRIPTION
                    "The current temperature measured at this location.
                    Unit is Degrees Celsius"
               ::= { temperatureEntry 4 }


          -- 1.0.62379.7.1.5.1.1.5
          temperatureStatus OBJECT-TYPE
               SYNTAX TemperatureStatus
               MAX-ACCESS read-only
               STATUS current
               DESCRIPTION
                    "The status of this particular temperature sensor.

                    Current Values are:
```

```
                    undetermined(0),
                    other(1),
                    unknown(2),
                    ok(3),
                    warning(4),
                    critical(5),
                    non-recoverable(6)"
               ::= { temperatureEntry 5 }


          -- 1.0.62379.7.1.5.1.1.6
          temperatureLowWarning OBJECT-TYPE
               SYNTAX IntegerNumber
               UNITS "Degrees Celcius"
               MAX-ACCESS read-only
               STATUS current
               DESCRIPTION
                    "The low warning limit for the measured temperature.
                    Unit is Degrees Celsius."
               ::= { temperatureEntry 6 }


          -- 1.0.62379.7.1.5.1.1.7
          temperatureHighWarning OBJECT-TYPE
               SYNTAX IntegerNumber
               UNITS "Degrees Celcius"
               MAX-ACCESS read-only
               STATUS current
               DESCRIPTION
                    "The high warning limit for the measured temperature.
                    Unit is Degrees Celsius."
               ::= { temperatureEntry 7 }


          -- 1.0.62379.7.1.5.1.1.8
          temperatureLowCritical OBJECT-TYPE
               SYNTAX IntegerNumber
               UNITS "Degrees Celcius"
               MAX-ACCESS read-only
               STATUS current
               DESCRIPTION
                    "The low critical limit for the measured temperature.
                    Unit is Degrees Celsius."
               ::= { temperatureEntry 8 }


          -- 1.0.62379.7.1.5.1.1.9
          temperatureHighCritical OBJECT-TYPE
               SYNTAX IntegerNumber
               UNITS "Degrees Celcius"
               MAX-ACCESS read-only
               STATUS current
               DESCRIPTION
                    "The high critical limit for the measured temperature.
                    Unit is Degrees Celsius."
               ::= { temperatureEntry 9 }


          -- 1.0.62379.7.1.5.2
          temperatureMeasurementGroup OBJECT-GROUP
               OBJECTS { temperatureLocation, temperatureTrend, temperatureStatus,
  temperatureLowWarning, temperatureHighWarning,
                    temperatureLowCritical, temperatureHighCritical }
               STATUS current
               DESCRIPTION
                    "The group of objects used for temperature measurements."
               ::= { temperatureMeasurement 2 }
```

```
END
```

Annex B
(informative)

Machine-readable textual conventions definitions

This annex provides a machine-readable version of the textual conventions definitions specific to this standard which is intended to be interpretable by standard MIB browsing software tools. If there is any inconsistency between this annex and clause 6.2.1, clause 6.2.1 takes precedence.

The format used to describe the textual conventions definitions conforms to IETF STD 58 (SMIv2).

```
IEC62379-7-IPM-TC-MIB DEFINITIONS ::= BEGIN

      IMPORTS
            iec62379
                  FROM IEC62379-1-MIB
            Unsigned32, Gauge32, MODULE-IDENTITY
                  FROM SNMPv2-SMI
            TEXTUAL-CONVENTION
                  FROM SNMPv2-TC;


      -- 1.0.62379.7.3
      measurementTCMIB MODULE-IDENTITY
            LAST-UPDATED "201106211200Z"        -- June 21, 2011 at 12:00 GMT
            ORGANIZATION
                  "IEC PT62379"
            CONTACT-INFO
                  "Not specified."
            DESCRIPTION
                  "The Textual Conventions MIB module for managing
                  measurement functions in IEC 62379 compliant
                  equipment.
                  The contents of this MIB Module have been defined by
                  the EBU ECN-IPM (European Broadcasting Union
                  Expert Communities Networks, IP Measurements group)."
            REVISION "201106101200Z"       -- June 10, 2011 at 12:00 GMT
            DESCRIPTION
                  "Added BitRateType textual convention."
            ::= { measurement 3 }



--
-- Textual conventions
--

      NetworkType ::= TEXTUAL-CONVENTION
            STATUS current
            DESCRIPTION
                  "An enumeration identifying a network type of over which the
                  media is flowing."
            SYNTAX INTEGER
                  {
                  ipv4(1),
                  ipv6(2),
                  asi(3)
                  }
-- {
-- ipv4(1),
-- ipv6(2),
-- asi(3)
-- } (ipv4..asi)
```

```
--
        TransportType ::= TEXTUAL-CONVENTION
              STATUS current
              DESCRIPTION
                    "An enumeration identifying a transport type of over which the
                    media is flowing.

                    Note that the values for this textual convention are NOT the
same
                    as the numbers used in the protocol field of IPv4 packets and
                    the Next Header Field of IPv6 packets.
                    See http://www.iana.org/assignments/protocol-numbers"
              SYNTAX INTEGER
                    {
                    notApplicable(0),
                    rtp(1)
                    }
-- {
-- notApplicable(0),
-- rtp(1)
-- }(notApplicable..rtp)
--
        AudioFECType ::= TEXTUAL-CONVENTION
              STATUS current
              DESCRIPTION
                    "An enumeration identifying the FEC type applied, if present."
              SYNTAX INTEGER
                    {
                    none(0),
                    smpte2021(1),
                    smpte2022(2),
                    rfc2733(3),
                    proprietary(4)
                    }
-- {
-- none(0),
-- smpte2021(1),
-- smpte2022(2),
-- rfc2733(3),
-- proprietary(4)
-- } (none..proprietary)
--
        VideoFECType ::= TEXTUAL-CONVENTION
              STATUS current
              DESCRIPTION
                    "An enumeration identifying the FEC type applied, if present."
              SYNTAX INTEGER
                    {
                    none(0),
                    smpte2022(2),
                    proprietary(4)
                    }
-- {
-- none(0),
-- smpte2022(2),
-- proprietary(4)
-- } (none..proprietary)
--
        BufferSize ::= TEXTUAL-CONVENTION
              STATUS current
              DESCRIPTION
                    "A type to indicate the current total size of the receive
buffer in
                    msecs."
              SYNTAX Unsigned32

        BufferOcpncyTime ::= TEXTUAL-CONVENTION
              STATUS current
              DESCRIPTION
                    "A type to report the amount of data, expressed in msecs,
```

```
                    occupying the receive buffer."
               SYNTAX Gauge32

          BufferOcpncyPercent ::= TEXTUAL-CONVENTION
               STATUS current
               DESCRIPTION
                    "A type to report the amount of data, expressed as a
                    percentage of the total receive buffer size, occupying the
                    receive buffer."
               SYNTAX INTEGER (1..100)

          TemperatureLocn ::= TEXTUAL-CONVENTION
               STATUS current
               DESCRIPTION
                    "A type to indicate the location where the temperature is
measured."
               SYNTAX OCTET STRING (SIZE (0..80))

          TemperatureTrend ::= TEXTUAL-CONVENTION
               STATUS current
               DESCRIPTION
                    "A type to report the current temperature at the measured
                    location.
                    The use of this type will allow changes (either up or down)
                    to be reported."
               SYNTAX Gauge32

          TemperatureStatus ::= TEXTUAL-CONVENTION
               STATUS current
               DESCRIPTION
                    "An enumeration identifying the temperature status levels.
                    Semantics are equipment specific."
               SYNTAX INTEGER
                    {
                    undetermined(0),
                    other(1),
                    unknown(2),
                    ok(3),
                    warning(4),
                    critical(5),
                    nonRecoverable(6)
                    }
-- {
-- undetermined(0),
-- other(1),
-- unknown(2),
-- ok(3),
-- warning(4),
-- critical(5),
-- nonRecoverable(6)
-- } (undetermined..nonRecoverable)
--
          BitRateType ::= TEXTUAL-CONVENTION
               STATUS current
               DESCRIPTION
                    "An enumeration identifying the video bit rate type applied

                    vbr = variable bit rate
                    cbr = constant bit rate"
               SYNTAX INTEGER
                    {
                    unspecified(0),
                    vbr(1),
                    cbr(2)
                    }
-- {
-- unspecified(0),
-- vbr(1),
-- cbr(2)
-- } (unspecified..cbr)
```

```
--

--
-- Node definitions
--

        -- 1.0.62379.7
        measurement OBJECT IDENTIFIER ::= { iec62379 7 }


   END
```

Annex C
(informative)

Worked examples

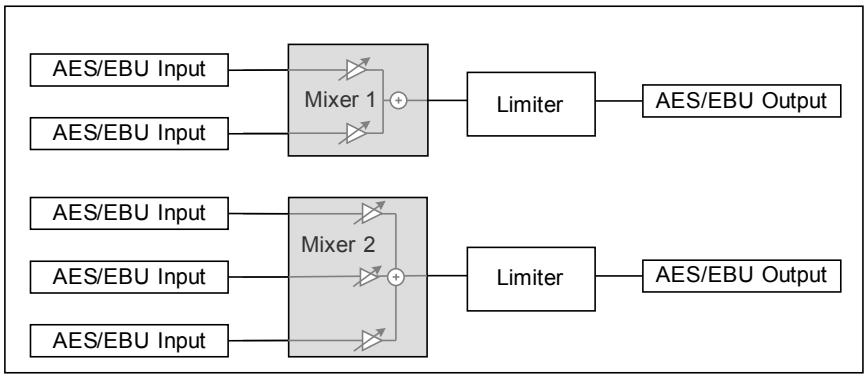
This Annex provides a worked example to show how parts 1, 2, 3 and 7 of this standard may be used in a vendor's MIB to standardise the measurement methods required by the EBU ECN-IPM Group. If there is any inconsistency between this Annex and the main clauses of parts 1, 2, 3 and 7, those clauses take precedence.

NOTE The examples shown here are intended to provide only a high level view of the process involved. Document IEC 62379-2 Audio Annex E & F provide detailed examples of how the block structure may be used to model audio devices. Similar processes may be applied to model video devices. See IEC 62379-3 Ed.1.

C.1 Example 1

To show the relationships between the various tables defined in Parts 1, 2, 3 & 7 of IEC 62379 the example shown here is a modified version of example 1 in Annex E of IEC 62379-2 Audio, to model an audio device which is then extended to describe a device that has multiple encoders/decoders of different types present in it and where there are multiple measurement blocks present.

The modified audio example device has two mixers, one with two inputs, the other with three, providing in addition, a limiter for each, as represented by the following diagram:



NOTE This unit example could be built using a single mixer block with five inputs and two outputs, but this arrangement is more appropriate to illustrate the use of measurement blocks.

Details of the audio that might be carried are not relevant for the description of this example, so are not included.

C.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.

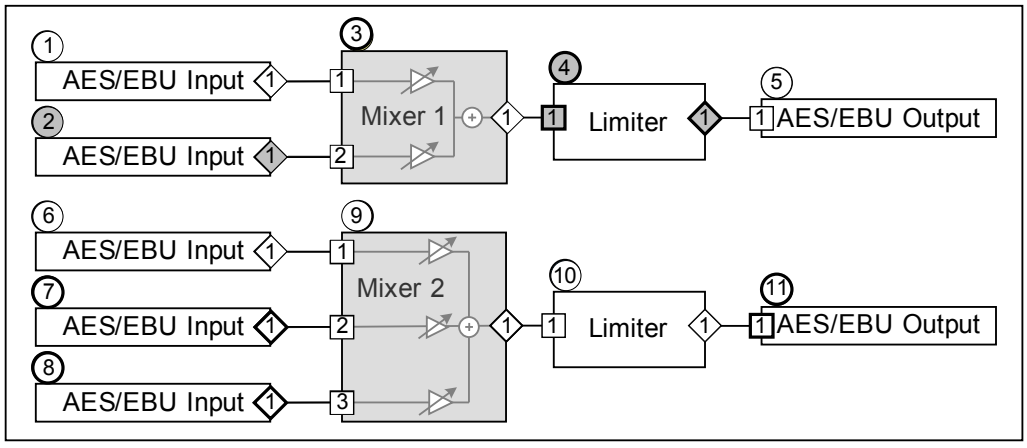
The table is located at iso.standard.iec67329.general.generalMIB.block.blockTable (1.0.62379.1.1.2.1).

As there are eleven blocks in the example, this table will have eleven 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)
⑥	1.0.62379.2.1.1 (Audio port)
⑦	1.0.62379.2.1.1 (Audio port)
⑧	1.0.62379.2.1.1 (Audio port)
⑨	1.0.62379.2.1.2 (Mixer block)
⑩	1.0.62379.2.1.5 (Limiter block)
⑪	1.0.62379.2.1.1 (Audio port)

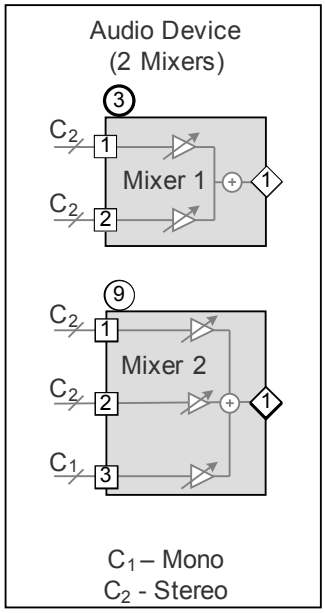
This identifies each of the block types that exist within a unit, along with the unique identifier `[blockId (1)]`. Although the blockId's are shown as incrementing numbers, any positive integer number can be used and in any order, as long as each is unique.

In example 1 of Annex E IEC 62379-2 Audio, the structure of the unit is shown with a connector table to express all the links that exist between the blocks defined in the block table. The connector table contains each block's blockId (in circles), input numbers (squares) and output numbers (diamonds) and the diagram of the unit now looks like this, annotated with these shaped numbers:



NOTE The connector table contents are not necessary for describing the use of the measurement blocks. For the same reason, nor are the contents or description of either the mode table or the audio port tables as shown in example 1 of Annex E IEC 62379-2 Audio, although all would be required for a real implementation of the extended example shown here.

For clarity, only the mixer section of the above diagram will be now used, so that the structure of an audio device having these two mixer blocks will now look like the diagram below, with the related block table:



.blockTable .blockEntry (.1.1)		
blockId (1)	*blockType (2)*	
①	1.0.62379.2.1.1 (Audio port)	**Not shown for clarity**
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)	**Not shown for clarity**
⑤	1.0.62379.2.1.1 (Audio port)	
⑥	1.0.62379.2.1.1 (Audio port)	
⑦	1.0.62379.2.1.1 (Audio port)	
⑧	1.0.62379.2.1.1 (Audio port)	
⑨	1.0.62379.2.1.2 (Mixer block)	
⑩	1.0.62379.2.1.5 (Limiter block)	**Not shown for clarity**
⑪	1.0.62379.2.1.1 (Audio port)	

The two mixer blocks, limiter blocks and audio port blocks are standard blocks defined in IEC 62379-2. The mixer inputs (and outputs) can be stereo, mono or other channel arrangement as defined in the audio signal formats.

C.1.2 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.

.aMixerBlockTable .aMixerBlockEntry (.1.1)		
aMixerBlockId (1)	*aMixerFadeDuration (2)*	*aMixerFadeNow (3)*
③		
⑨		

.aMixerInputTable .aMixerInputEntry (.2.1)				
aMixerInput BlockId (1)	*aMixerInput Number (2)*	*aMixerInput Level (3)*	*aMixerInput FadeToLevel (4)*	*aMixerInput Delay (5)*
③	1			
③	2			
⑨	1			
⑨	2			
⑨	3			

As the example has two mixer blocks, one with two inputs, the other with three, `aMixerBlockTable` has two entries (of type `AMixerBlockEntry`). This defines the overall fader information for each mixer block.

The larger of the two tables, `aMixerInputTable` has five entries (of type `AMixerInputEntry`), two for mixer block ID 3 and three for mixer block ID 9. The number of times these exist being dependent upon the number of inputs in each mixer block. This allows each input of each mixer block to be uniquely identified.

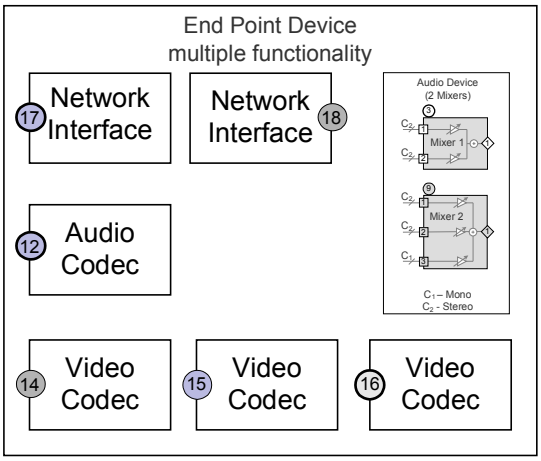
The remaining entries in this table would contain information relevant to each input.

Note Other information pertaining to these mixer blocks is not included, such as the connections to other parts of the initial example. This information would be held in other tables, such as connection and/or mode tables, similar to the example shown in the IEC62379-2 document.

C.1.3 Multiple Functionality Device

Having considered how the process of assigning blockIds and input identifiers to describe a unit, a similar principle can be applied in this example, where we consider a single device with multiple functionality within it, consisting of what we have already used (2 mixer blocks) and now adding 2 network interfaces, 1 audio codec and 3 video codecs, each with a differing number of associated audio components.

Each type of standard measurement block has an entry in the main block table, appearing as many times as there are codecs or network interfaces, each with a unique identifier (blockId). Measurement Blocks of the same type all have the same OID. i.e. all audio measurement blocks have the same OID, but different blockId's.



Audio codec – single audio	12	
14 Video codec – 1 video 2 audio	19	
15 Video codec – 1 video 3 audio	13	
16 Video codec – 1 video 2 audio	X	

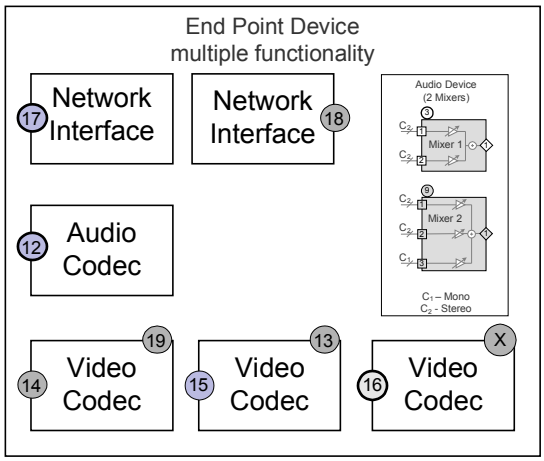
These blockId's are for the different types of measurement blocks only. In practice there would be additional blockId entries for the codecs themselves and whatever facilities that these provided within the equipment, similar to the audio mixer blocks example shown earlier.

.blockTable.blockEntry (.1.1)		
blockId (1)	*blockType (2)*	
1	1.0.62379.2.1.1 (Audio port)	Not shown for clarity
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)	
6	1.0.62379.2.1.1 (Audio port)	Not shown for clarity
7	1.0.62379.2.1.1 (Audio port)	
8	1.0.62379.2.1.1 (Audio port)	
9	1.0.62379.2.1.2 (Mixer block)	
10	1.0.62379.2.1.5 (Limiter block)	Not shown for clarity
11	1.0.62379.2.1.1 (Audio port)	
12	1.0.62379.7.1.1 (Audio Measurement)	
13	1.0.62379.7.1.1 (Audio Measurement)	
14	1.0.62379.7.1.2 (Video Measurement)	
15	1.0.62379.7.1.2 (Video Measurement)	
16	1.0.62379.7.1.2 (Video Measurement)	
17	1.0.62379.7.1.3 (Network Measurement)	
18	1.0.62379.7.1.3 (Network Measurement)	
19	1.0.62379.7.1.1 (Audio Measurement)	
X	1.0.62379.7.1.1 (Audio Measurement)	

For example, an audio codec might be constructed from one or more standard audio blocks specified in IEC 62379-2 (or elsewhere), such as a converter block, so this would also have an entry in the main block table.

C.1.3.1 Audio Components

Having now defined each measurement block ID, we now need to deal with the various audio components associated with each measurement block. The audio measurement table is used commonly for both audio only codecs and all the audio components associated with the video codecs.



Video Measurement Table			
	vMtAudio BlockId (2)	*vMtNetwork BlockId (3)*	*Other Video Parameters*
(14)	(13)		
(15)	(19)		
(16)	(X)		

Network Measurement Table	
nMtBlockId (1)	*Other Network Parameters*
(17)	
(18)	

Audio Measurement Table			
aMtBlockId (1)	*aMtAudioComponent Number (2)*	*aMtNetwork BlockId (3)*	*Other Audio Parameters*
(12)	*1*		
(13)	*1*		
(13)	*2*		
(13)	*3*		
(19)	*1*		
(19)	*2*		
(X)	*1*		
(X)	*2*		

Audio codec (ID 12) has only one audio associated with it, so there is only one entry line in the audio measurement table. Video codec (ID 14) has 2 audio components. It has one entry in the video measurement table and an associated audio measurement block ID of 19. This is used to identify the row(s) in the audio measurement table that are the associated audio components with this video. The audio measurement table has therefore 2 entries for the audio components with ID 19 and the audio component number is used to uniquely identify each audio component.

Although not completed in this example, whichever network the codecs might be connected to internally, the appropriate network ID would need to be entered into the audio and video measurement tables matching up with the correct audio or video codec block ID.

A similar process is used for all the other video components and their associated audio components. Each network interface has ID's of their own in the network measurement table.

This whole process allows for devices with multiple functionality, where the functionality may be different or repeated many times.

C.1.4 Summary of Tables