



TECHNICAL REPORT

TR-135

Data Model for a TR-069 Enabled STB

Version: Issue 1
Version Date: December 2007

Notice

The Broadband Forum is a non-profit corporation organized to create guidelines for broadband network system development and deployment. This Technical Report has been approved by members of the Forum. This document is not binding on the Broadband Forum, any of its members, or any developer or service provider. This document is subject to change, but only with approval of members of the Forum.

This document is provided "as is," with all faults. Any person holding a copyright in this document, or any portion thereof, disclaims to the fullest extent permitted by law any representation or warranty, express or implied, including, but not limited to,

- (a) any warranty of merchantability, fitness for a particular purpose, non-infringement, or title;
- (b) any warranty that the contents of the document are suitable for any purpose, even if that purpose is known to the copyright holder;
- (c) any warranty that the implementation of the contents of the documentation will not infringe any third party patents, copyrights, trademarks or other rights.

This publication may incorporate intellectual property. The Broadband Forum encourages but does not require declaration of such intellectual property. For a list of declarations made by Broadband Forum member companies, please see www.broadband-forum.org.

Version History

Version Number	Version Date	Version Editors	Changes
Issue 1	December 2007	Marco Quacchia, Telecom Italia William Lupton, 2Wire Gilles Straub, Thomson	Original

Technical comments or questions about this document should be directed to:

Editors

Marco Quacchia	Telecom Italia	marco.quacchia@telecomitalia.it
William Lupton	2Wire	wlupton@2wire.com
Gilles Straub	Thomson	gilles.straub@thomson.net

**BroadbandHome™
Technical Working
Group
Chairs**

Greg Bathrick	PMC-Sierra	Greg_Bathrick@pmc-sierra.com
Heather Kirksey	Motive	hkirksey@motive.com

Table of Contents

1	PURPOSE	8
2	SCOPE.....	8
2.1	DEFINITIONS	9
2.2	CONVENTIONS	9
3	REFERENCES	9
4	USE CASES	9
4.1	CONFIGURATION	9
4.2	TROUBLE MANAGEMENT.....	9
4.3	PERFORMANCE MANAGEMENT	9
4.4	FAULT MANAGEMENT	9
5	ARCHITECTURE	9
5.1	CAPABILITIES	9
5.2	COMPONENTS.....	9
5.2.1	<i>FrontEnd</i>	9
5.2.2	<i>PVR</i>	9
5.2.3	<i>AudioDecoder</i>	9
5.2.4	<i>VideoDecoder</i>	9
5.2.5	<i>AudioOutput</i>	9
5.2.6	<i>VideoOutput</i>	9
5.2.7	<i>SCART</i>	9
5.2.8	<i>CA</i>	9
5.2.9	<i>DRM</i>	9
5.3	AV STREAMS.....	9
5.4	AV PLAYERS	9
5.5	SERVICE MONITORING	9
5.6	APPLICATIONS	9
5.6.1	<i>AudienceStats</i>	9
6	PARAMETER DEFINITIONS.....	9
6.1	NOTIFICATION REQUIREMENTS.....	9
7	PROFILE DEFINITIONS.....	9
7.1	NOTATION	9
7.2	BASELINE PROFILE	9
7.3	PVR PROFILE	9
7.4	DTT PROFILE	9
7.5	IPTVBASELINE PROFILE	9
7.6	RTCP PROFILE.....	9
7.7	RTPAVPF PROFILE	9
7.8	IPTVHOMENETWORK PROFILE.....	9
7.9	IGMP PROFILE.....	9

7.10	BASICPERFMON PROFILE	9
7.11	ECPERFMON PROFILE.....	9
7.12	VIDEOPERFMON PROFILE	9
7.13	AUDIOPERFMON PROFILE.....	9
7.14	AUDIENCESTATS PROFILE.....	9
7.15	ANALOGOUTPUT PROFILE.....	9
7.16	DIGITALOUTPUT PROFILE.....	9
7.17	CA PROFILE	9
7.18	DRM PROFILE.....	9
APPENDIX I – THEORY OF OPERATIONS		9
I.1	PROFILE USAGE	9
I.2	INSTANCE NUMBER USAGE.....	9
I.2.1	<i>Fixed objects with fixed purpose</i>	<i>9</i>
I.2.2	<i>Fixed objects with variable purpose</i>	<i>9</i>
I.2.3	<i>Objects created as needed.....</i>	<i>9</i>
I.3	SERVICE MONITORING	9
I.3.1	<i>Key properties</i>	<i>9</i>
I.3.2	<i>Operational overview.....</i>	<i>9</i>
I.3.3	<i>Service types and MainStream instances</i>	<i>9</i>
I.3.4	<i>Sample statistics overview.....</i>	<i>9</i>
I.3.5	<i>Sample statistics details</i>	<i>9</i>
I.3.6	<i>RTP statistics.....</i>	<i>9</i>
I.3.7	<i>MPEG2-TS statistics</i>	<i>9</i>
I.4	CONFIGURATION	9
I.4.1	<i>DVB-T Front End service list.....</i>	<i>9</i>
I.4.2	<i>AV Players.....</i>	<i>9</i>
I.5	TROUBLE MANAGEMENT.....	9
I.5.1	<i>Explicit connect to service.....</i>	<i>9</i>
I.6	PERFORMANCE MANAGEMENT	9
I.7	FAULT MANAGEMENT	9
I.8	AV STREAM AND AV PLAYER EXAMPLES.....	9
I.8.1	<i>Flow 3: DVB-T Front End to Analog AV Player</i>	<i>9</i>
I.8.2	<i>Flow 4: DVB-T Front End to PVR.....</i>	<i>9</i>
I.8.3	<i>Flow 5: PVR to Analog AV Player.....</i>	<i>9</i>
I.8.4	<i>Flow 6: IPTV to PVR</i>	<i>9</i>
I.8.5	<i>Flow 7: Home Network Media Server to Digital AV Player.....</i>	<i>9</i>
I.8.6	<i>Flow 8: PVR to Home Network Media Renderer.....</i>	<i>9</i>
I.8.7	<i>Flow 3 + Flow 7: Two AV Streams in Digital AV Player.....</i>	<i>9</i>

List of Figures

Figure 1/TR-135 – STB Context.....	9
Figure 2/TR-135 – STBService Object Structure	9
Figure 3/TR-135 – Logical AV Streaming Model	9
Figure 4/TR-135 – Logical AV Streaming Model with Example Streams.....	9
Figure 5/TR-135 – Logical AV Player Model	9
Figure 6/TR-135 – Sample statistics parameters.....	9
Figure 7/TR-135 – STB Context (same as Figure 1/TR-135).....	9

Summary

This document, *Data Model for a TR-069 Enabled STB*, defines the data model for remote management of Digital Television (IPTV or broadcast) functionality on Set Top Box (STB) devices via CWMP as defined in TR-069 and TR-106. It covers the data model for describing an STB device as well as rules regarding notifications on parameter value change. General use cases are also described, including standard data model profiles that would typically be seen while remotely managing a device of this nature.

Access to network and PVR content is managed by a (proprietary) IPTV Service Platform. The ACS may perform some initial configuration of a newly installed STB, but its main functions are configuration of STB parameters for trouble management and collection of statistics for QoS / QoE monitoring.

Broadband Forum TECHNICAL REPORT TR-135

Data Model for a TR-069 Enabled STB

1 Purpose

This document, *Data Model for a TR-069 Enabled STB*, defines the data model for remote management of Digital Television (IPTV or broadcast) functionality on STB devices via CWMP as defined in [1] and [2]. It covers the data model for describing an STB device as well as rules regarding notifications on parameter value change. General use cases are also described including standard data model profiles that would typically be seen while remotely managing a device of this nature.

This document defines STBService as the container associated with the remote management of objects for STB devices. CPE devices making use of an STBService object MUST adhere to all of the data-hierarchy requirements defined in [2]. In the context of [2], the STBService object is a service object. As such, individual CPE devices can contain one or more of these objects within their Services object alongside the generic data objects defined in [2]. The presence of more than one STBService object would be appropriate primarily where a CPE device serves as a management proxy for other non TR-069 capable STBService devices. For example, an Internet Gateway Device might serve as a management proxy for one or more non TR-069 capable STBs.

2 Scope

Figure 1 illustrates an STB and its relationships. This diagram defines the context for the remarks below and shows some of the data flows within and through the STB. The diagram also provides the examples of section I.8.

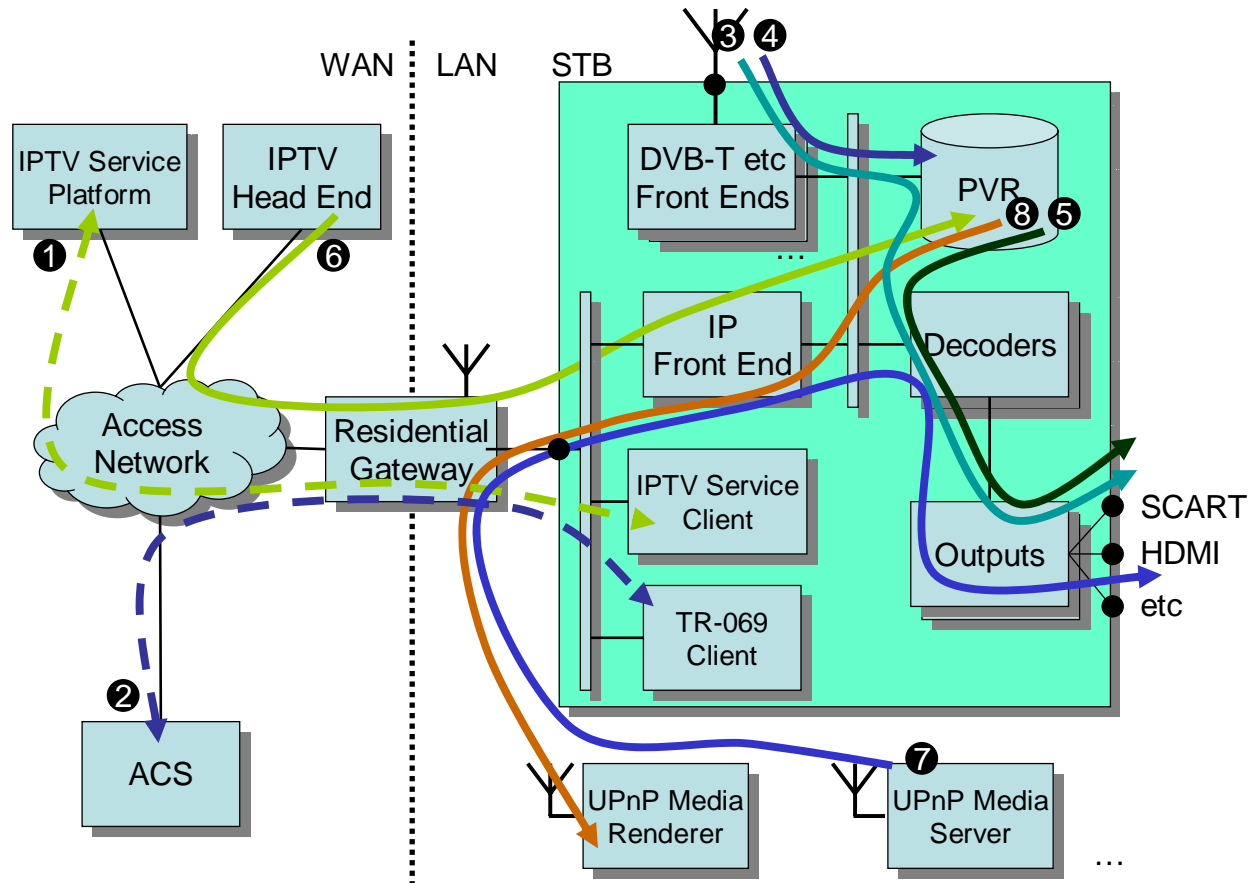


Figure 1/TR-135 – STB Context

The STB can be connected to a number of external networks, including broadband networks like IPTV and broadcast networks like DTT (Digital Terrestrial Television), CAB (Cable) and SAT (Satellite). The STB receives AV Streams from these external networks, if necessary via the Residential Gateway, and combines them in various ways for presentation on one or more display devices, each of which can present different content. The end user selects which content is displayed. The displays can be connected to the STB via an analog connector (e.g. a SCART [33] connector), a specific digital connector (e.g. an HDMI connector), or a digital network connector (e.g. Ethernet or WLAN interfaces). Consumption of AV content can be subject to rules imposed by means of CA (Conditional Access) or DRM (Digital Rights Management) systems. These may require the use of a Smart Card or may rely on secrets integrated into the decoding chip.

The PVR (Personal Video Recorder) function can send AV content, in compressed and possibly encrypted form, to a storage device. As well as recording the content and allowing it to be viewed after the recording is over, the PVR can include a time-delay, or time-shift, function. This allows the end user to suspend consumption of the content and subsequently to resume it from the point at which it was suspended. To do this the PVR acts as a buffer with length equal to the time for which viewing was suspended. Trick modes (Fast Forward, Rewind, and Pause) can be supported on live, recorded and time-delayed content. Other PVR functions can include the recording of VoD (Video on Demand) events, in which a piece of (protected) content is streamed

to the STB, or Push VoD, in which a piece of (protected) content is downloaded (pushed) to the PVR for possible subsequent viewing by the end user.

Access to network and PVR content is managed by a (proprietary) IPTV Service Platform. In addition to this, the ACS monitors the operation and performance of STBs, allows specific STB parameters to be checked, and carries out (limited) configuration of STBs.

Even though it is not easy to draw a line between the IPTV Service Platform and the ACS domains, it is reasonable to assume that the STB is controlled by the IPTV Service Platform for all tasks related to provisioning, media processing and Conditional Access / Digital Rights Management, and is controlled by the ACS for all tasks related to trouble management and monitoring of performance and operational status.

An STB should also be regarded as part of a Home Network, in which the STB can consume content as well as provide or relay content to other devices. Content may be stored locally within the STB, within another Home Network device such as a Media Center, or can come from one or more of the external networks to which the Home Network is connected. Local and remote access could be taking place at the same time.

Based on this scenario, the goals of this specification are as follows:

- Enable configuration by the ACS of those objects and parameters that are not the responsibility of the IPTV Service Platform.
- Enable operational status monitoring and checking of specific parameters of an STB from an ACS.
- Enable performance monitoring of an arbitrary set of STBs, from one to millions, through estimates of QoS (Quality of Service) and QoE (Quality of Experience), where QoS and QoE are defined in [3].
- Support various types of STB, including DTT and IP STBs, with or without PVR and other optional functionality.
- Accommodate STB devices that are embedded as part of an Internet Gateway Device, as defined in [2].
- Accommodate STB devices that are standalone, i.e. implemented in separate hardware devices, as defined in [2].

2.1 Definitions

The following definitions are used throughout this document.

ACS	Auto-Configuration Server. This is a component in the broadband network responsible for auto-configuration of the CPE for advanced services.
ATSC	Advanced Television Systems Committee. Digital television standards, primarily adopted in North America.
Audio Decoder	A logical component that receives an elementary stream, decodes it, and outputs an uncompressed (native format) audio stream.

Audio Output	A logical component that receives an uncompressed audio stream and adapts it as required by the presentation device (such as a TV set).
AV	Audio and Video.
AV Player	One or more <i>AV Streams</i> (one <i>AV Main Stream</i> , possibly several PIP video streams, one audio stream) connected to zero or more <i>AudioOutputs</i> and zero or more <i>VideoOutputs</i> .
AV Main Stream	This is used in two different contexts: <ul style="list-style-type: none"> • An <i>AV Stream</i> that is destined for the main screen rather than a <i>PIP</i> • An <i>AV Stream</i> associated with a <code>.ServiceMonitoring.MainStream.{i}</code> instance
AV Stream	One of the following chains, corresponding to a channel (or program) that the user is currently watching or recording (see also <i>AV Player</i>): <ul style="list-style-type: none"> • <i>FrontEnd</i> → {<i>AudioDecoder</i>, <i>VideoDecoder</i>} • <i>FrontEnd</i> → <i>PVR</i> → {<i>AudioDecoder</i>, <i>VideoDecoder</i>} • <i>FrontEnd</i> → <i>PVR</i> • <i>PVR</i> → {<i>AudioDecoder</i>, <i>VideoDecoder</i>} • <i>PVR</i> → <i>FrontEnd</i> • <i>FrontEnd</i> → <i>FrontEnd</i> • <i>FrontEnd</i> → <i>PVR</i> → <i>FrontEnd</i>
BTv	Broadcast Television.
CA	Conditional Access.
CAB	Cable.
CBR	Constant Bit Rate.
CPE	Customer Premises Equipment.
DCCP	Datagram Congestion Control Protocol.
DLNA	Digital Living Network Alliance.
DRM	Digital Rights Management.
DSM-CC	Digital Storage Media Command and Control.
DTT	Digital Terrestrial Television.
DVB	Digital Video Broadcasting. Digital television standards, widely used worldwide.
DVB-C	<i>DVB</i> Cable standard.
DVB-S	<i>DVB</i> Satellite standard.
DVB-T	<i>DVB</i> Terrestrial; <i>DVB DTT</i> standard.

DVD	Digital Versatile Disk, or Digital Video Disk.
EC	Error Correction.
ES	<i>MPEG</i> Elementary Stream. An audio, video (or text) stream.
FrontEnd	A logical component that converts an incoming multiplex from channel format, e.g. <i>OFDM</i> modulation for <i>DTT</i> or <i>RTP/UDP</i> encapsulation for IP, to a continuous (still multiplexed) bit stream.
FTPS	File Transfer Protocol over SSL (FTP/SSL), IETF RFC 2228.
HD	High Definition.
HDCP	High-bandwidth Digital Content Protection.
HDMI	High Definition Media Interface.
IGMP	Internet Group Management Protocol.
Interconnection Bus	Conceptual bus that interconnects <i>FrontEnds</i> , <i>AudioDecoders</i> , <i>VideoDecoders</i> and the <i>PVR</i> , multiplexing/demultiplexing and handling timing and synchronization, as necessary.
IPTV	Internet Protocol Television.
ISDB	Integrated Services Digital Broadcasting. Japanese digital television and audio broadcasting format.
Main Stream	<i>AV Main Stream</i> .
MPEG	Moving Picture Experts Group.
MPEG2-TS	MPEG-2 Transport Stream.
OFDM	Orthogonal Frequency Division Multiplexing.
OSS	Operations Support Systems.
Parameter	A name-value pair representing a manageable CPE parameter made accessible to an ACS for reading and/or writing.
PCR	<i>MPEG2-TS</i> Program Clock Reference.
PES	MPEG-2 Packetized Elementary Stream.
PIP	Picture In Picture.
Player	<i>AV Player</i> .
PPV	Pay Per View.
PS	MPEG-2 Program Stream.
PVR	Personal Video Recorder.
QoE	Quality of Experience.
QoS	Quality of Service.
RTCP	Real Time Control Protocol, IETF RFC 3550.

RTP	Real Time Protocol, IETF RFC 3550.
SAT	Satellite.
SCART	Société des Constructeurs d'Appareils Radiorécepteurs and Téléviseurs. A connector standard that supports various common analog signal types.
SCTP	Stream Control Transmission Protocol.
SD	Standard Definition.
SDO	Standards Development Organization.
SSL	Secure Sockets Layer.
STB	Set Top Box. This device contains Audio and Video decoders and is intended to be connected to Analog TV and / or Home Theaters.
Stream	<i>AV Stream.</i>
TFTP	Trivial File Transfer Protocol, IETF RFC 1350.
TS	<i>MPEG2-TS.</i>
UPnP	Universal Plug'n'Play.
USB	Universal Serial Bus.
VCR	Video Cassette Recorder.
Video Decoder	A logical component that receives an elementary stream, decodes it, and outputs an uncompressed (native format) video stream.
Video Output	A logical component that receives an uncompressed video stream and adapts it as required by the presentation device (such as a TV set).
VoD	Video on Demand.
WLAN	Wireless Local Area Network.

2.2 Conventions

In this document, several words are used to signify the requirements of the specification. These words are often capitalized.

MUST	This word, or the adjective “REQUIRED”, means that the definition is an absolute requirement of the specification.
MUST NOT	This phrase means that the definition is an absolute prohibition of the specification.
SHOULD	This word, or the adjective “RECOMMENDED”, means that there may exist valid reasons in particular circumstances to ignore this item, but the full implications must be understood and carefully weighted before choosing a different course.
MAY	This word, or the adjective “OPTIONAL”, means that this item is one of an allowed set of alternatives. An implementation that does not include this option MUST be prepared to inter-operate with another implementation that does include the option.

3 References

The following Broadband Forum Technical Reports and other references contain provisions, which, through reference in this text, constitute provisions of this Technical Report. At the time of publication, the editions indicated were valid. All Technical Reports and other references are subject to revision; users of this Technical Report are therefore encouraged to investigate the possibility of applying the most recent edition of the Technical Report and other references listed below. A list of the currently valid Broadband Forum Technical Reports is published at www.broadband-forum.org.

NOTE – The reference to a document within this Technical Report does not give it, as a stand-alone document, the status of a Technical Report.

The following documents are referenced by this specification.

- [1] TR-069 Amendment 1, *CPE WAN Management Protocol*, Broadband Forum Technical Report
- [2] TR-106 Amendment 1, *Home Network Data Model Template for TR-069-Enabled Devices*, Broadband Forum Technical Report
- [3] TR-126, *Triple-Play Services Quality of Experience (QoE) Requirements*, Broadband Forum Technical Report
- [4] TR-140, *TR-069 Data Model for Storage Service Devices*, Broadband Forum Technical Report
- [5] RFC 1350, *The TFTP Protocol (Revision 2)*, <http://www.ietf.org/rfc/rfc1350.txt>
- [6] RFC 2119, *Key words for use in RFCs to Indicate Requirement Levels*, <http://www.ietf.org/rfc/rfc2119.txt>
- [7] RFC 2228, *FTP Security Extensions*, <http://www.ietf.org/rfc/rfc2228.txt>
- [8] RFC 2326, *Real Time Streaming Protocol (RTSP)*, <http://www.ietf.org/rfc/rfc2326.txt>
- [9] RFC 2960, *Stream Control Transmission Protocol (SCTP)*, <http://www.ietf.org/rfc/rfc2960.txt>
- [10] RFC 3066, *Tags for the Identification of Languages*, <http://www.ietf.org/rfc/rfc3066.txt>
- [11] RFC 3376, *Internet Group Management Protocol (IGMP), Version 3*, <http://www.ietf.org/rfc/rfc3376.txt>
- [12] RFC 3986, *Uniform Resource Identifier (URI): Generic Syntax*, <http://www.ietf.org/rfc/rfc3986.txt>
- [13] RFC 3550, *RTP: A Transport Protocol for Real-Time Applications*, <http://www.ietf.org/rfc/rfc3550.txt>
- [14] RFC 4340, *Datagram Congestion Control Protocol (DCCP)*, <http://www.ietf.org/rfc/rfc4340.txt>
- [15] RFC 4585, *Extended RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback (RTP/AVPF)*, <http://www.ietf.org/rfc/rfc4585.txt>

- [16] RFC 4588, *RTP Retransmission Packet Format*, <http://www.ietf.org/rfc/rfc4588.txt>
- [17] ETSI EN 300 744, *Digital Video Broadcasting (DVB): Framing structure, channel coding and modulation for digital terrestrial television*
- [18] ETSI TS 102 034, *Digital Video Broadcasting (DVB); Transport of MPEG-2 Based DVB Services*
- [19] ISO/IEC 11172-1 (1993), *Information Technology - Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s - Part 1: System*
- [20] ISO/IEC 11172-2 (1993), *Information Technology - Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s - Part 2: Video*
- [21] ISO/IEC 11172-3 (1993), *Information Technology - Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s - Part 3: Audio*
- [22] ISO/IEC 13818-1 (2nd edition, 2000), *Information technology - Generic coding of moving picture and associated audio information: Systems*
- [23] ITU-T Rec. H.262 | ISO/IEC 13818-2 (2nd edition, 2000), *Information Technology - Generic Coding of moving pictures and associated audio information: Video*
- [24] ISO/IEC 13818-3 (2nd edition, 1998), *Information technology - Generic coding of moving picture and associated audio information: Audio*
- [25] ISO/IEC 13818-6 (2nd edition, 1998), *Information technology - Generic coding of moving picture and associated audio information: Extensions for DSM-CC*
- [26] ISO/IEC 14496-2:2001, *Information technology – Generic coding of audio-visual objects – Part 2: Visual*
- [27] ISO/IEC 14496-3:2001, *Information technology – Coding of Audio-Visual objects – Part 3: Audio*
- [28] ISO/IEC 23003-1:2007, *Information technology – MPEG audio technologies – Part 1: MPEG Surround*
- [29] ITU-T Rec. H.264 | ISO/IEC 14496-10:2004/AM 1, Part 10, *Advanced Video Coding AMENDMENT 1: AVC fidelity range extensions*
- [30] ITU-T Rec X.700, *Management Framework for Open Systems Interconnection (OSI) FOR CCITT Applications*
- [31] ISMA white paper, *Planning the Future of IPTV with ISMA*
- [32] DGTVi D-Book v1.0, Sep 2004, *Compatible receivers for the Italian market*
- [33] CENELEC technical standard EN 50049-1:1997
- [34] SMPTE 20022-1 Application Layer Forward Error Correction
- [35] ATIS IIF (IPTV Interoperability Forum), *Quality of Service Metrics for Linear Broadcast IPTV*, document number: ATIS-0800008, August 2007
- [36] ATSC, *Digital Audio Compression (AC-3) (E-AC-3) Standard, Rev. B*, document number A/52B, June 2005

[37] SMPTE, Television – *VC-1 Compressed Video Bitstream Format and Decoding Process*, document number 421M, 2006

[38] *SSH File Transfer Protocol*, <http://tools.ietf.org/wg/secsh/draft-ietf-secsh-filexfer>

4 Use cases

A number of remote management use cases can be considered for the scenario of section 2. Some of them are presented here. The STB data model supports at least the functionality that is implied by these use cases.

Classification of remote management activities can be done by making reference to the FCAPS model [30] for systems management, where FCAPS stands for:

- **F**ault
- **C**onfiguration
- **A**ccounting
- **P**erformance
- **S**ecurity

The STB data model does not need to account for all of the FCAPS functions. Usually, the FCAPS functions supported are Fault, Configuration and Performance. Accounting and Security functions usually take advantage of pre-existing infrastructure, so the relevant use cases are not considered here.

Configuration of the STB is done both by the IPTV Service Platform and by higher layer OSSs via the ACS. It can also be done by a trained technician, usually as a reaction to an end user complaint. This latter activity is here referred to as *Trouble Management*.

Performance Management can be performed by the ACS on a regular basis, to try and identify a malfunction as soon as it occurs, or by trouble management personnel on a limited set of STBs for special purposes.

Fault Management is generally driven by fault notifications from the STB. It is usually carried out automatically by higher layer OSS systems, based on signalling from the ACS.

These use cases are discussed briefly in the following sections, and are revisited in Appendix I.

4.1 Configuration

The ACS may perform some initial configuration of a newly installed STB. For example, it might initiate a channel scan in order to populate a DTT service list database, or it might set some user preferences such as audio and subtitling languages. During the initial configuration, the ACS can also update the STB firmware. Most of the initial configuration will be performed by the IPTV Service Platform.

4.2 Trouble Management

A trained technician may take control of the STB, generally in response to a customer complaint. The STB malfunction may be the result of improper customer settings, or may be due to network

or hardware problems. Access to the STB data model allows the technician to carry out a number of tasks, namely:

- Verify/Restore the STB configuration. The STB data model parameters under the ACS control can be re-configured to the correct values contained in the ACS.
- Verify/Update software version. Incorrect software version (e.g. the STB was switched off for a long time and was not included in the last software upgrade campaign) can cause improper operation. In this case the operator can force an upgrade of the STB software to the latest release.
- Perform diagnostics. The technician can run diagnostic tests to identify whether the trouble is in the network (and at which point) or the STB and try to classify the trouble.

Depending on the cases, the technician can carry out actions on specific subsets of STBs (identified e.g. by a range of serial numbers, by a specific software/hardware version, by the geographical area they are in) or on single devices.

4.3 Performance Management

The ACS carries out automatic monitoring of STB performance. Performance reports can include QoS parameters (e.g. network parameters such as average bit rate, jitter and packet loss ratio), QoE parameters (e.g. visual quality indicators or indicators of how fast on the average the channel change is), usage statistics (e.g. how many STBs were on at a certain time, or for how long each of them remained tuned to a certain channel).

Monitoring campaigns may be performed:

- Periodically on all STB devices to check that network and devices are working properly,
- On subsets of STB devices, for instance after identifying problems by means of periodic tests. Criteria to select subsets can be geographical or tied to specific characteristics of the STBs (manufacturer, hardware and/or software version),
- Periodically on specific STB devices. The problem here could be the management of an SLA (Service Level Agreement) with subscribers to premium services. Performance management could be used to identify problems on these lines as soon as they show up. Trouble management technicians could then act to (try to) solve them.

STB QoS and QoE reporting capabilities allow for “in service” “passive” measurements done at the service level. These are of fundamental importance to an operator in a number of cases, a number of which are listed hereinafter. Other cases are possible beyond those listed here:

- Understand and measure the QoE delivered to individual end users, via collection and aggregation of STB reports across the user base.
- Troubleshoot the service delivered: STB reporting allows near real time processing of collected reports and correlation of indicators that let the operator determine where the fault lies: in the head end, in the network, in the local loop, in the home network, or in the STB itself.
- Assess and measure the IPTV service as delivered in the mid to long term, and define and control whether performance objectives are being met.

- Pro-actively catch some hidden behavior which is increasing, and is reducing service performance, but has not yet been noticed by the end user.
- Pro-actively manage certain end users who are receiving a poor level of service but who have not yet called customer care.
- Configure and define operations management service quality thresholds on aggregated reports that can be tuned in order to take action before problems are noticed or reported by the end users.
- Understand loop and end-to-end behavior in order to design and assess error correction strategies for the IPTV service.
- Manage service maintenance and understand the impact on the IPTV service of any changes in the network, device upgrades or new device insertion.

4.4 Fault Management

The ACS automatically collects events from the STB for various reasons, including detection of faults. A way to detect faults taking advantage of TR-069 notification features could be the following: the data model contains parameters describing the operational status of specific functional blocks in the STB, and Active Notification is enabled for these parameters. In case of an STB error these parameters change their values and the Active Notification mechanism delivers information about the STB fault to the ACS. The ACS recognizes the fault and consequently notifies the OSS in charge of the End-to-End Fault Management operations.

5 Architecture

This data model describes only functions that are strictly specific to an STB. Other functions that could be present in an STB as well as in other devices, like the Hard Disk of the PVR, or a Smart Card, are modelled separately.

Figure 2 depicts the STBService object structure as seen in the Parameter Definitions section. This figure provides a high-level overview of the different objects that exist in this data model and how they are nested.

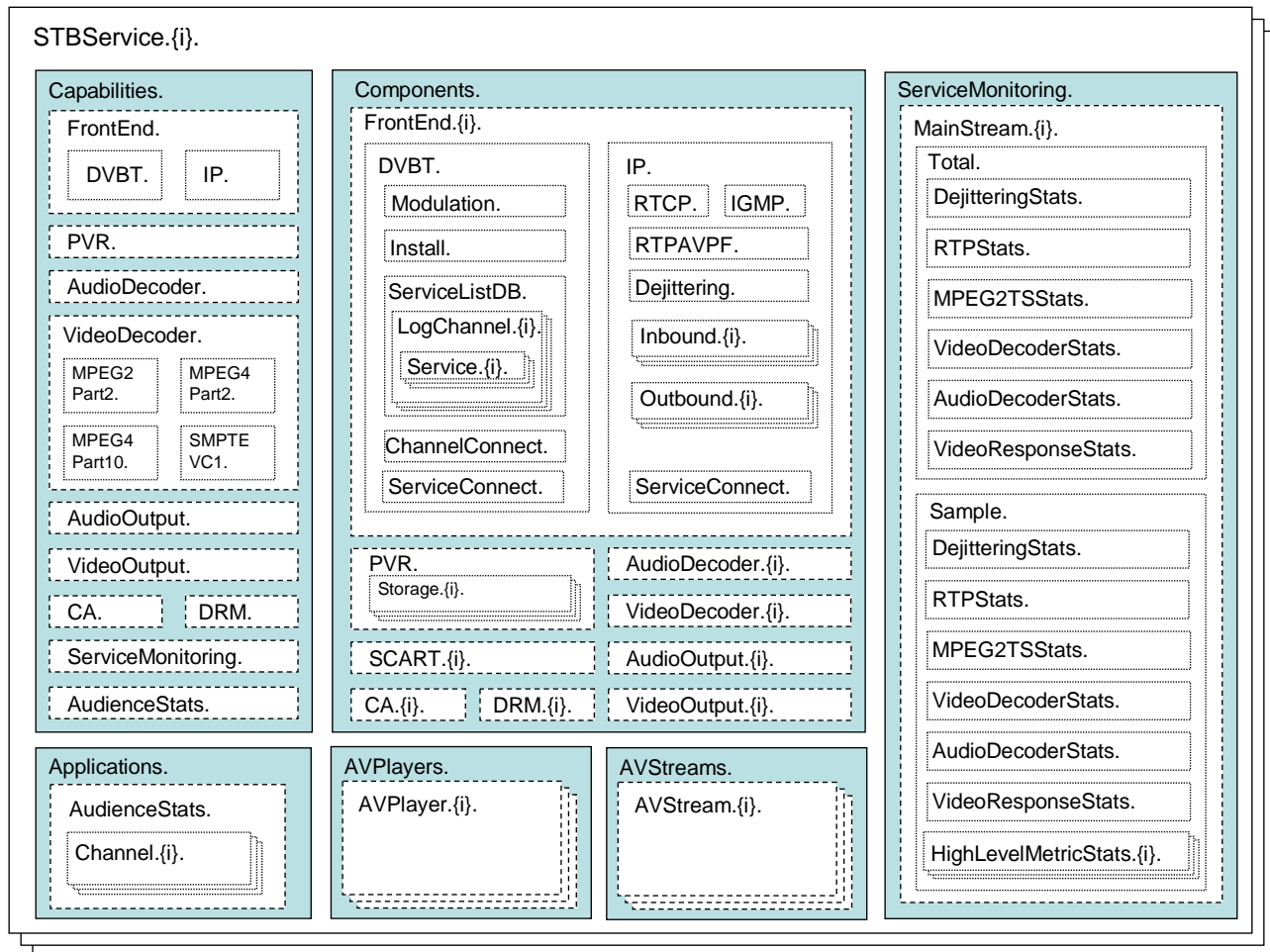


Figure 2/TR-135 – STBService Object Structure

The following sections give a high-level overview of the STB data model. Figure 2 illustrates the object hierarchy but does not indicate the logical relationships between objects. For an explanation of these logical relationships, please refer to the descriptions of the AVStreams (Section 5.3) and AVPlayers (Section 5.4) objects.

5.1 Capabilities

The STB data model contains a Capabilities object that describes what every component (functional block) of the STB can do. Component CCC's capabilities are modelled in `.Capabilities.CCC.`; each CCC instance is modelled in `.Components.CCC.{i}`.

Capabilities describe, among other details, the supported audio and video standards. Audiovisual standards often indicate which audio standards are allowed with which video standards. In addition to that, certain STBs may not support all combinations of audio and video standards listed. For these reasons the Capabilities object lists, for every video standard, the audio standards supported.

Capabilities consist only of read-only objects and parameters, meaning that only a firmware update will cause the values to be altered.

5.2 Components

The STB data model contains a Components object that describes the device's functional blocks. It contains the following components: FrontEnd, PVR, AudioDecoder, VideoDecoder, AudioOutput, VideoOutput, SCART, CA and DRM, each of which (except for PVR) can be multi-instance.

Component Objects are created statically and persist throughout STB operation, i.e. the STB creates the Component Object instances that it supports at boot time and does not add or delete any during operation. Not all the object instances are necessarily operational at the same time, so they can be enabled / disabled via an "Enable" parameter, and their status (Enabled / Disabled / Error) is made available to the ACS via a "Status" parameter. Furthermore, object instances need to be identified, so an STB-chosen name is made available to the ACS via a "Name" parameter.

5.2.1 FrontEnd

Front End objects model network side interfaces.

A Front End acts as an interface between the network and the inner functional blocks of the STB. The network can be the home network or an external network (e.g. DTT, IPTV). Depending on the network type, connections modelled by Front End objects can be unidirectional (e.g. DTT), or bidirectional (IP). An IP Front End can be bi-directional because the STB can be both a content destination and a content source.

The Front End instances in a given STB will normally correspond closely with the STB's hardware capabilities. For example, a hybrid STB with two DTT tuners and an IPTV interface would be expected to have two DTT Front Ends and a single IP Front End, all of which could in theory be operational at the same time.

This version of the data model does not define CAB (Cable) and SAT (Satellite) Front End objects, so vendor extensions such as `.Components.FrontEnd.{i}.X_ABCDEF_DVBS` will be needed if such Front Ends are to be modelled.

Note that the IP Front End models the STB's LAN connection and is a logical rather than physical concept, which means that an STB never has more than one IP Front End, even if it has more than one LAN IP or physical interface (which are modelled by the generic objects defined in [2]). The IP Front End is typically capable of handling multiple input and output streams at the same time.

It is assumed that input audio, video and data that are part of the same program are synchronized so that they can be displayed consistently. Synchronization can be achieved via a multiplexed stream, for which the most common format is MPEG2-TS [22]. It can also be achieved at the IP layer by sending elementary (single media) streams directly over IP. Different Front End objects can support different synchronization formats, with or without multiplexing.

The simplest case for multiplexing is the broadcast interface, for instance DTT in which the (Multi Program) MPEG2-TS packets are mapped directly onto the physical layer. Multiplexing, timing and synchronization are all carried out by the MPEG2-TS layer.

A more complex case is that of the IP interface. The IP protocol stack, like the DTT physical layer, allows carriage of the MPEG2 Transport Stream packets. In addition to this, the IP framework also offers the option of implementing multiplexing, timing and synchronization on

its own, meaning that the MPEG2-TS layer is not strictly necessary. For example, timing and synchronization can be implemented in the IP framework by means of the RTP protocol, the whole protocol stack being RTP/UDP/IP. Another possible protocol stack is HTTP/TCP/IP. The IP framework also allows describing the audio /video /data multiplex. This IP-only option is far less common at the moment than MPEG2-TS over IP but could gain momentum in the future as it is backed by a certain number of SDOs, for example by ISMA [31].

Owing to the discontinuous nature of IP transmission, a de-jittering buffer is modelled at the Inbound side of the IP FrontEnd. The buffer size can be modified for trouble management purposes, though this is disruptive of the normal operation, to minimize underflows and overflows while keeping the buffering delay to a minimum.

Monitoring of the de-jittering buffer status is carried out at the MPEG2-TS level by counting the ingress and egress MPEG2-TS packet rate.

5.2.2 PVR

The PVR stores programs coming from any Front End and sends stored programs to Audio and/or Video Decoders or to the (Output) IP Front End. All of the embedded storage accessible to the PVR is modelled via TR-140 [4] StorageService instances within the STB. As noted in section 2, the PVR also performs standard (VCR-like) recording functions as well as advanced ones like time-delay or trick modes. It is assumed that PVR functions are managed by the IPTV Service Platform.

The STB data model's support for the PVR is limited to describing its capabilities and to referencing the TR-140 StorageService objects. In addition, AV Stream (Section 5.3) objects have a PVRState parameter which indicates whether they are using the PVR and, if so, its playback state.

5.2.3 AudioDecoder

AudioDecoder objects describe the functional blocks in charge of audio decoding.

An Audio Decoder receives an elementary audio stream, decodes the audio, and outputs an uncompressed native audio stream to an Audio Output object.

5.2.4 VideoDecoder

VideoDecoder objects describe the functional blocks in charge of video decoding.

A Video Decoder receives an elementary video stream, decodes the video, and outputs an uncompressed native video stream to a Video Output object.

5.2.5 AudioOutput

AudioOutput objects describe the functional blocks in charge of audio rendering.

An Audio Output receives uncompressed audio streams from one or more Audio Decoders and performs format adaptations as required by the relevant presentation standard (e.g., analog mono or stereo audio as needed for speakers and/or headphones, S/P DIF or HDMI/HDCP as needed by specific digital devices). Adaptation of the audio to the specified output format can include digital-to-analog conversion or other analog or digital processing, including encryption.

Each Audio Output is mapped to one or more physical output connectors. Where an Audio Output is mapped directly to a SCART connector, this is indicated in the data model via a reference from the Audio Output to the corresponding SCART instance (other types of physical connector are not modelled).

5.2.6 VideoOutput

VideoOutput objects describe the functional blocks in charge of video rendering.

A Video Output receives uncompressed video streams from one or more Video Decoders, and performs format adaptations as required by the relevant presentation standard (e.g. analog or digital displays, possibly with an encrypted link between STB and display). Adaptation of the video to the specified output format can include various actions like digital-to-analog conversion, resizing, aspect ratio conversion, addition of analog or digital protection.

Each Video Output is mapped to one or more physical output connectors. Where a Video Output is mapped directly to a SCART connector, this is indicated in the data model via a reference from the Video Output to the corresponding SCART object instance (other types of physical connector are not modelled).

5.2.7 SCART

The SCART (Société des Constructeurs d'Appareils Radiorecepteurs et Televiseurs) connector [33] is specified by CENELEC, a European SDO, and is extremely popular in European AV devices. The video format being sent to the monitor (CVBS, S-Video, RGB etc) and the aspect ratio ("presence" control signal) are made available in the data model.

Use of SCART connectors in STBs allows for easy integration with existing analog devices. Usually, up to two SCART sockets are provided in an STB, one for Standard Definition Television, the other for the VCR.

5.2.8 CA

A CA (Conditional Access) component contains details of one of the CA mechanisms that may be supported by the STB. In principle, there may be any number of CA platforms, including none (for instance in a free-to-air STB).

5.2.9 DRM

A DRM (Digital Rights Management) component contains details of one of the DRM mechanisms that may be supported by the STB. In principle, there can be any number of DRM platforms, including none (for instance in a free-to-air STB).

5.3 AV Streams

An AV Stream is modelled by a chain of components that work together in one of the following ways:

- FrontEnd → {AudioDecoder, VideoDecoder} (Normal viewing)
- FrontEnd → PVR → {AudioDecoder, VideoDecoder} (Normal viewing with time delay / trick modes / recording)

- FrontEnd → PVR (Recording)
- PVR → {AudioDecoder, VideoDecoder} (Viewing of pre-recorded content)
- PVR → FrontEnd (STB acting as a Media Centre: viewing of pre-recorded content on a home network device)
- FrontEnd → FrontEnd (Streaming of network content, e.g. DTT or IPTV, to a home network device)
- FrontEnd → PVR → FrontEnd (Streaming of network content, e.g. DTT or IPTV, to a home network device with time delay / trick modes / recording)

The streaming model supported by the STB data model is, therefore, that an Audio + Video / Audio-only / Video-only stream flows from a Front End (e.g. DTT, IP) to the Audio and Video Decoders, the PVR or the IP Outbound Front End; alternatively, a stream from the PVR flows to Audio and Video Decoders or to the IP Outbound Front End.

AV Streams are modelled in `.AVStreams.AVStream.{i}`. Each AV Stream object indicates whether it involves the PVR, and is also associated with the relevant Front End, Audio Decoder and Video Decoder instances.

Figure 3 illustrates the component relationships. In the Figure, most of the components are connected by an Interconnection Bus. The Interconnection Bus performs demultiplexing/multiplexing functions when required, and also takes timing and synchronization constraints into account. The Interconnection Bus also carries out adaptation between Front Ends, Decoders and PVR data format that in principle could be different. For example, no assumption is made about the PVR data format, although MPEG2-TS [22] is likely. The Interconnection Bus, though essential from the conceptual point of view, needs no modelling and thus is not present in the STB data model.

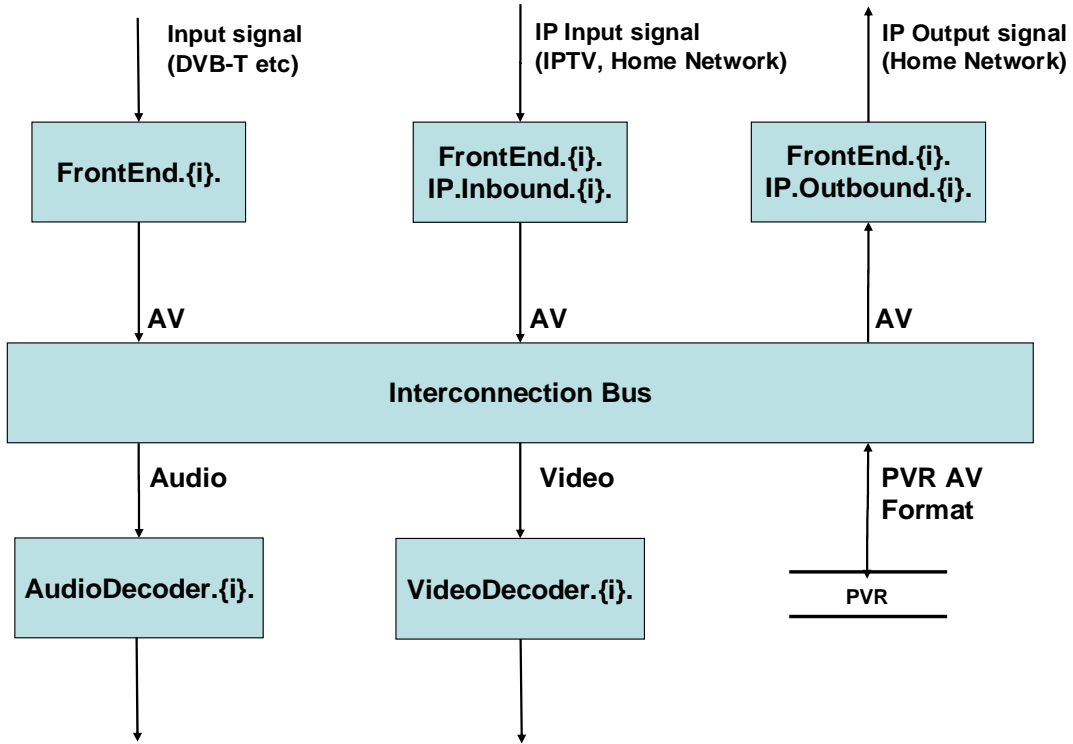


Figure 3/TR-135 – Logical AV Streaming Model

Figure 4 is a version of Figure 3 that includes shaded regions that illustrate some possible AV streams.

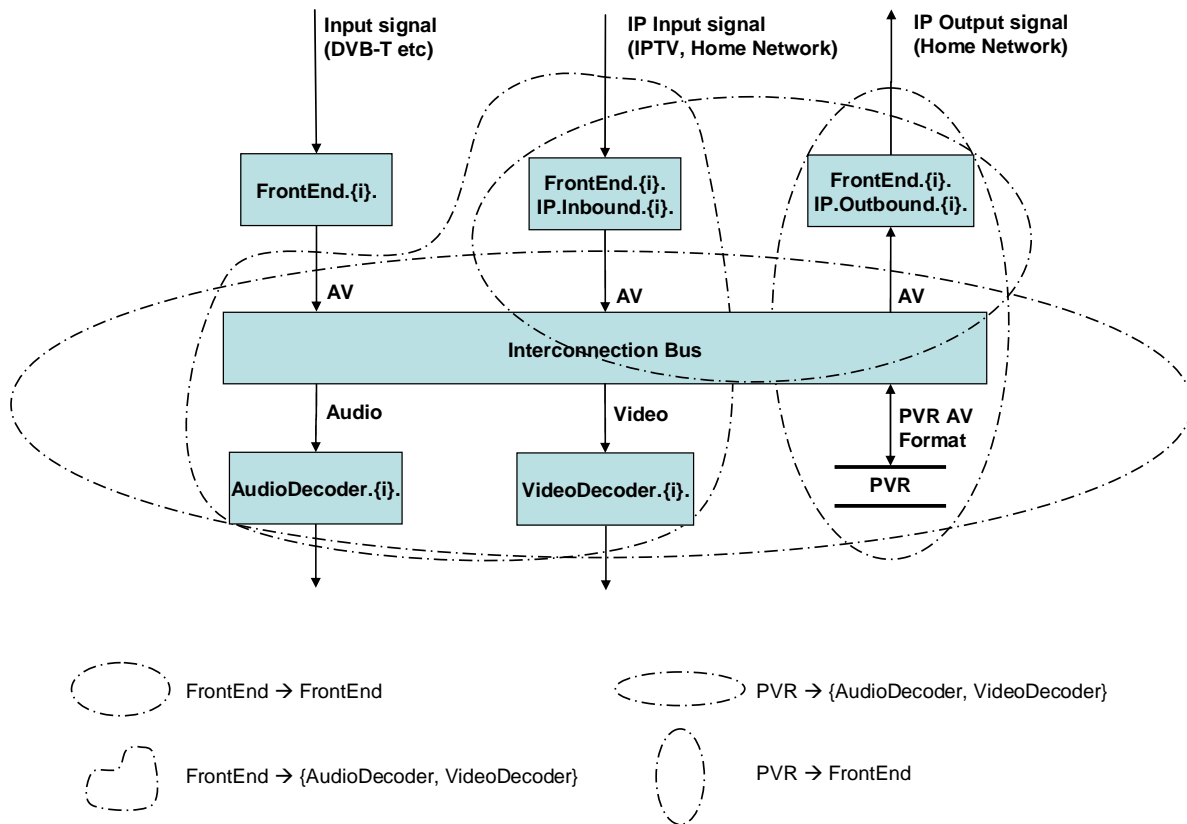


Figure 4/TR-135 – Logical AV Streaming Model with Example Streams

5.4 AV Players

An AV Player takes one or more AV streams and associates them with Audio Output and Video Output objects.

In many cases, the end user will be watching a single program and there will only be a single AVStream. In some more complex cases, e.g. involving simultaneous program watching (Picture in Picture / Mosaic), multiple AVStreams, possibly coming from different Front Ends, need to be combined and delivered together.

AV Players are modelled in `.AVPlayers.AVPlayer.{i}`. Each AV Player object is associated with one Main AV Stream, zero or more PIP AV Streams, zero or more Audio Output objects, and zero or more Video Output objects.

The end user may wish to send several AVStreams to different output devices simultaneously, e.g. one stream to a TV set and another stream to a VCR or DVD recorder, so AVPlayer needs to be a multi-instance object.

For example, an AV Stream could be sent both to a VCR via an analog RGB output (typically via SCART) and to the HDMI output. This could be modelled by means of a single AV Player referencing the AV Stream object and multiple Audio Output and Video Output objects. A case

involving multiple AV Player instances is the viewing of a program on a TV set and the simultaneous recording of another program on a VCR.

Figure 5 illustrates the logical AV Player model, showing the case where one Main AV stream is displayed along with two PIP (Picture in Picture) streams. The Audio stream is usually associated to the Main Video stream but this does not have to be the case. The data model accounts for the case of a totally independent Audio stream rendered along with the set of Main and PIP Video streams.

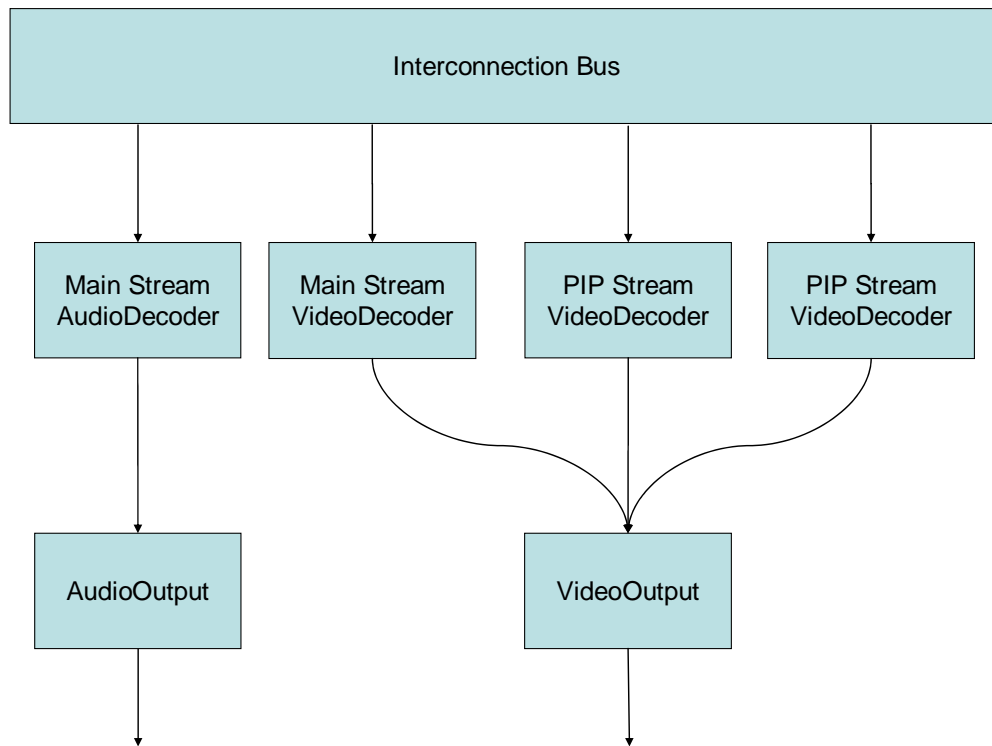


Figure 5/TR-135 – Logical AV Player Model

5.5 Service Monitoring

Service Monitoring statistics are collected by service type, e.g. IPTV, VoD, DTT. The main reason for defining service types is that they correspond to different protocol stacks and configurations, and statistics collected across multiple service types would be meaningless.

Statistics are broken down into seven categories: De-jittering, RTP, MPEG2-TS, Video decoding, Audio decoding, Video response and High-level metrics. Many parameters in the RTP and MPEG2-TS categories are taken from [35].

For each category (exception: High-level metrics have no Total statistics) there are two types of statistics:

- Total statistics, which typically simply count the number of occurrences of something, e.g. the number of received packets, since the STB last booted or since statistics were last reset.

- Sample statistics, which are measured over a sample interval, and are made available to the ACS as a list of the most recent n samples.

5.6 Applications

The STB data model contains an Applications object that contains information relating to high-level applications.

5.6.1 AudienceStats

This object contains audience viewing statistics, organised by channel.

6 Parameter Definitions

Table 1 lists the objects associated with an STB CPE device and their associated parameters. This table defines version 1.0 of the STBService data model.

The notation used to indicate the data type of each parameter, and the notation associating with multi-instance objects, follows the notation defined in [2].

Note that, as specified in section 3.3 of [2], a vendor may extend the standardized parameter list with vendor-specific parameters, objects and enumeration values.

Table 1/TR-135 – Parameter list for an STB CPE device

Name ¹	Type	Write ²	Description	Object Default ³
.STBService.{i}.	object	-	The top-level object for an STB CPE.	-
Enable	boolean	W	Enables or disables this STBService instance.	-
.STBService.{i}.Capabilities.	object	-	The overall capabilities of the STB CPE. This is a constant read-only object, meaning that only a firmware update will cause these values to be altered.	-
MaxActiveAVStreams	int[-1:]	-	Maximum number of simultaneously active AV streams supported by the STB. A value of -1 indicates no specific limit on the number of active streams. A value of 0, while not strictly illegal, would indicate an STB that doesn't support AV streams, which is extremely unlikely.	-
MaxActiveAVPlayers	int[-1:]	-	Maximum number of simultaneously active AV players supported by the STB. An AV player is associated with a presentation device (such as a TV set). Note that this is the strict maximum but, depending on the AV streams to be decoded, a given STB may not always have the resources to run all of these AV players, e.g. it might be able to decode one main picture and one PIP in standard definition, but just a main picture and no PIP in high definition. A value of -1 indicates no specific limit on the number of active players. A value of 0, while not strictly illegal, would indicate an STB that doesn't support AV players, which is extremely unlikely.	-
.STBService.{i}.Capabilities.-FrontEnd.	object	-	Front-end capabilities.	-
.STBService.{i}.Capabilities.-FrontEnd.DVBT.	object	-	Capabilities of the DVB-T receiver.	-

¹ The full name of a Parameter is the concatenation of the root object name as defined in [2], the object name shown in the yellow header, and the individual Parameter name.

² “W” indicates the parameter MAY be writable (if “W” is not present, the parameter is defined as read-only). For an object, “W” indicates object instances can be Added or Deleted.

³ The default value of the parameter on creation of an object instance via TR-069. If the default value is an empty string, this is represented by the symbol <Empty>.

Name ¹	Type	Write ²	Description	Object Default ³
MaxActiveDVBTStreams	int[-1:]	-	Maximum number of simultaneous active AV streams supported by the DVB-T FrontEnd. A value of -1 indicates no specific limit on the number of active streams.	-
MaxLogicalChannels	int[-1:]	-	Maximum number of logical channels that can be contained in a Service List Database. A value of -1 indicates no specific limit on the number of logical channels.	-
.STBService.{i}.Capabilities.-FrontEnd.IP.	object	-	IP front-end capabilities.	-
MaxActiveIPStreams	int[-1:]	-	Maximum number of simultaneous active AV streams supported by the IP FrontEnd. A value of -1 indicates no specific limit on the number of active streams.	-
MaxActiveInboundIPStreams	int[-1:]	-	Maximum number of simultaneous active Inbound AV streams supported by the IP FrontEnd. A value of -1 indicates no specific limit on the number of active streams.	-
MaxActiveOutboundIPStreams	int[-1:]	-	Maximum number of simultaneous active Outbound AV streams supported by the IP FrontEnd. A value of -1 indicates no specific limit on the number of active streams.	-
StreamingControlProtocols	string	-	Comma-separated list of streaming control protocols supported. Each value is taken from the list: "RTSP" supports RTSP [8] "IGMP" supports IGMPv1 [11] "IGMPv2" supports IGMPv2 and lower versions [11] "IGMPv3" supports IGMPv3 and lower versions [11] "DSM-CC" supports DSM-CC [25]	-
StreamingTransportProtocols	string	-	Comma-separated list of network protocols supported for AV streaming. Each item is taken from the list: "UDP" "TCP" "SCTP" [9] "DCCP" [14] "RTP" [13] "HTTP"	-
StreamingTransportControlProtocols	string	-	Comma-separated list of transport control protocols supported for AV streaming in combination to RTP. Each item is taken from the list: "RTCP" supports sending RTCP receiver reports as defined in [13] "ARQ" as defined in [15] and [16] "AL-FEC" as defined in [34]	-

Name ¹	Type	Write ²	Description	Object Default ³
DownloadTransportProtocols	string	-	Comma-separated list of network protocols supported for AV download. Each item is taken from the list: "HTTP" "HTTPS" "FTP" "FTPS" FTP/SSL [7] "SFTP" SSH file transfer protocol [38] "FTTP" [5]	-
MultiplexTypes	string	-	Comma-separated list of AV multiplex types supported by the front end. Each item is taken from the list: "None" "MPEG1-SYS" [19] "MPEG2-PS" [22] "VOB" [22] "MPEG2-TS" [22]	-
MaxDejitteringBufferSize	int[-1:]	-	Describes the maximum de-jittering buffer size, in bytes, supported by the STB. A value of -1 indicates no specific limit on the buffer size.	
.STBService.{i}.Capabilities.PVR.	object	-	PVR capabilities.	-
MaxIOStreams	int[-1:]	-	Maximum number of AV streams that the PVR can simultaneously record. A value of -1 indicates no specific limit on the number of streams.	-
MaxRecordingStreams	int[-1:]	-	Maximum number of AV streams that the PVR can simultaneously record. A value of -1 indicates no specific limit on the number of streams.	-
MaxPlaybackStreams	int[-1:]	-	Maximum number of AV streams that the PVR can simultaneously play back. A value of -1 indicates no specific limit on the number of streams.	-
MaxTimeDelay	int[-1]	-	Maximum time delay, in seconds, that the PVR time-shift mode can support. A value of -1 indicates no specific limit on the time delay.	-
.STBService.{i}.Capabilities.-AudioDecoder.	object	-	Audio decoder capabilities.	-

Name ¹	Type	Write ²	Description	Object Default ³
AudioStandards	string	-	Comma-separated list of audio standards supported by this STB. Possible items are: "MPEG1-Part3-Layer2" [21] "MPEG1-Part3-Layer3" [21] "MPEG2-Part3 -Layer2" [24] "MPEG2-Part3-Layer3" [24] "MPEG2.5" "MP3-PRO" "MP3-Surround" "MPEG2-AAC-LC" Low Complexity [24] "MPEG4-AAC-LC" Low Complexity [30] "MPEG4-AAC-MP" Main Profile [30] "MPEG4-BSAC" Scalable AAC [30] "MPEG4-AAC-HE-v2" High Efficiency Profile Level 2 [30] "MPEG-Surround" [28] "DOLBY-AC3" Dolby Digital (AC-3) [36] "DOLBY-DD+" Dolby Digital Plus [36] "DTS" Digital Theatre System "DTS-HD"	-
.STBService.{i}.Capabilities.-VideoDecoder.	object	-	Video decoder capabilities.	-
VideoStandards	string	-	Comma-separated list of video standards supported by this STB. Possible items are: "MPEG2-Part2" [22] "MPEG4-Part2" [25] "MPEG4-Part10" Same as MPEG4 AVC and H.264 [29] "SMPTE-VC-1" [37]	-
.STBService.{i}.Capabilities.-VideoDecoder.MPEG2Part2.	object	-	Object describing the set of supported MPEG2 profiles and levels for this STB. It also describes the set of audio standards supported when MPEG2 Part 2 is used as the video standard.	-
AudioStandards	string	-	Comma-separated list of supported Audio Standards supported by the Player when associated with MPEG2 Part 2 Video. Each item is taken from the list defined by .Capabilities.AudioDecoder.AudioStandards.	-
ProfileLevelNumberOfEntries	unsignedInt	-	Number of instances of ProfileLevel.	-

Name ¹	Type	Write ²	Description	Object Default ³
.STBService.{i}.Capabilities.-VideoDecoder.MPEG2Part2.-ProfileLevel.{i}.	object	-	Table to describe the set of profiles and levels combinations supported by the STB when MPEG2 Part 2 is used as video standard. Each entry in this table refers to a distinct combination of profile and level. The table MUST include a distinct entry for each supported combination of these parameters.	-
Profile	string	-	Identifier of the MPEG2 Part 2 profile, taken from the list: "SP" Simple Profile "MP" Main Profile "SNR" SNR Scalable "Spt" Spatially Scalable "HP" High Profile "4:2:2"	-
Level	string	-	Identifier of the MPEG2 part 2 level, taken from the list: "LL" Low Level "ML" Main Level "H-14" High-1440 "HL" High Level	-
MaximumDecodingCapability	unsignedInt	-	The maximum decoding capability for this video standard in Kilobits per second. This strictly depends on the profile and level selected.	-
.STBService.{i}.Capabilities.-VideoDecoder.MPEG4Part2.	object	-	Object describing the set of supported profiles and levels for this STB. It also describes the set of audio standards supported when MPEG4 Part 2 is used as the video standard.	-
AudioStandards	string	-	Comma-separated list of supported Audio Standards supported by the Player when associated with MPEG4 Part 2 video. Each item is taken from the list defined by .Capabilities.AudioDecoder.AudioStandards.	-
ProfileLevelNumberOfEntries	unsignedInt	-	Number of instances of ProfileLevel.	-

Name ¹	Type	Write ²	Description	Object Default ³
.STBService.{i}.Capabilities.-VideoDecoder.MPEG4Part2.-ProfileLevel.{i}.	object	-	Table to describe the set of profiles and levels combinations supported by the STB when MPEG4 Part 2 is used as video standard. Each entry in this table refers to a distinct combination of profile and level. The table MUST include a distinct entry for each supported combination of these parameters.	-
Profile	string	-	Identifier of the MPEG4 Part 2 profile, taken from the list: "SIMPLE" "SIMPLE SCALABLE" "CORE" "CORE SCALABLE" "ADVANCED CORE" "MAIN" "N-BIT" "ADVANCED REAL TIME SIMPLE" "ADVANCED CODING EFFICIENCY" "SIMPLE STUDIO" "CORE STUDIO" "ADVANCED SIMPLE" "FINE GRANULARITY SCALABLE" "ADVANCED SCALABLE TEXTURE" "ANIMATED 2D MESH" "BASIC ANIMATED TEXTURE" "STILL SCALABLE TEXTURE" "SIMPLE FACE" "SIMPLE FBA"	-
Level	string	-	Identifier of the MPEG4 Part 2 level, taken from the list: "L5" "L4" "L4a" "L3b" "L3" "L2" "L1" "L0"	-
MaximumDecodingCapability	unsignedInt	-	The maximum decoding capability for this video standard in Kilobits per second. This strictly depends on the profile and level selected.	-
.STBService.{i}.Capabilities.-VideoDecoder.MPEG4Part10.	object	-	Object describing the set of supported profiles and levels for this STB. It also describes the set of audio standards supported when MPEG4 Part 10 is used as the video standard.	-
AudioStandards	string	-	Comma-separated list of supported Audio Standards supported by the Player when associated with MPEG4 Part 10 video. Each item is taken from the list defined by .Capabilities.AudioDecoder.AudioStandards.	-
ProfileLevelNumberOfEntries	unsignedInt	-	Number of instances of ProfileLevel.	-

Name ¹	Type	Write ²	Description	Object Default ³
.STBService.{i}.Capabilities.- VideoDecoder.MPEG4Part10.- ProfileLevel.{i}.	object	-	Table to describe the set of profiles and levels combinations supported by the STB when MPEG4 Part 10 is used as video standard. Each entry in this table refers to a distinct combination of profile and level. The table MUST include a distinct entry for each supported combination of these parameters.	-
Profile	string	-	Comma-separated list of supported MPEG4 Part 10 profiles. Each item is an enumeration of: "BASELINE" "MAIN" "EXTENDED" "HIGH" "HIGH 10" "HIGH 4:2:2" "HIGH 4:4:4"	-
Level	string	-	Comma-separated list of supported MPEG4 Part 10 Levels. Each item is an enumeration of: "1" "1b" "1.1" "1.2" "1.3" "2" "2.1" "2.2" "3" "3.1" "3.2" "4" "4.1" "4.2" "5" "5.1"	-
MaximumDecodingCapability	unsignedInt	-	The maximum decoding capability for this video standard in Kilobits per second. This strictly depends on the profile and level selected.	-
.STBService.{i}.Capabilities.- VideoDecoder.SMPTEVC1.	object	-	Object describing the set of supported - SMPTE-VC-1 profiles and levels for this STB. It also describes the set of audio standards supported when SMPTE-VC-1 is used as the video standard.	-
AudioStandards	string	-	Comma-separated list of supported Audio Standards supported by the Player when associated with SMPTE-VC-1 video. Each item is taken from the list defined by .Capabilities.AudioDecoder.AudioStandards.	-
ProfileLevelNumberOfEntries	unsignedInt	-	Number of instances of ProfileLevel.	-

Name ¹	Type	Write ²	Description	Object Default ³
.STBService.{i}.Capabilities.- VideoDecoder.SMPTEVC1.- ProfileLevel.{i}.	object	-	Table to describe the set of profiles and levels combinations supported by the STB when SMPTE-VC-1 is used as video standard. Each entry in this table refers to a distinct combination of profile and level. The table MUST include a distinct entry for each supported combination of these parameters.	-
Profile	string	-	Identifier of the SMPTE-VC-1 profile, taken from the list: "SP" Simple Profile "MP" Main Profile "AP" Advanced Profile	-
Level	string	-	Identifier of the SMPTE-VC-1 level, taken from the list: "LL" Low Level "ML" Medium Level "HL" High Level "L0" Advanced Profile only "L1" Advanced Profile only "L2" Advanced Profile only "L3" Advanced Profile only "L4" Advanced Profile only	-
MaximumDecodingCapability	unsignedInt	-	The maximum decoding capability for this video standard in Kilobits per second. This strictly depends on the profile and level selected.	-
.STBService.{i}.Capabilities.- AudioOutput.	object	-	Audio output capabilities. The audio output stage may include selecting output formats, selecting output connectors, and copy protection.	-
AudioFormats	string	-	Comma-separated list of supported audio output formats. Each item is taken from the list: "ANALOG-MONO" "ANALOG-2-CHANNELS" "ANALOG-5.1-CHANNELS" "DIGITAL-OPTICAL-SP/DIF" "DIGITAL-COAXIAL-SP/DIF" "HDMI" "RF" (modulated analog signals on coax) Note that these refer to supported output formats and not to physical connectors.	-
.STBService.{i}.Capabilities.- VideoOutput.	object	-	Video output capabilities. The video output stage may include selecting output formats, selecting output connectors, and copy protection.	-

Name ¹	Type	Write ²	Description	Object Default ³
CompositeVideoStandards	string	-	Comma-separated list of analog composite video standards supported for video output. This applies only to CVBS video output. Each item is taken from the list: "NTSC-J" "NTSC-M" "PAL-B" "PAL-D" "PAL-G" "PAL-H" "PAL-I" "PAL-N" "PAL-M" "SECAM-B" "SECAM-G" "SECAM-H" "SECAM-D" "SECAM-K" "SECAM-K1" "SECAM-L"	-
VideoFormats	string	-	Comma-separated list of supported video output formats. Each item is taken from the list: "CVBS" "S-Video" (same as Y/C) "YPrPb" "RGsB" (RGB with Sync on Green) "RGB" (RGB with external Sync) "HDMI" "DVI" "RF" (modulated analog signals on coax) Note that these refer to supported output formats and not to physical connectors.	-
Macrovision	string(32)	-	Supported version of Macrovision digital content protection standard.	-
HDCP	string(32)	-	Supported version of HDCP digital content protection standard.	-
.STBService.{i}.Capabilities.CA.	object	-	This object describes the characteristics of the Conditional Access, if any.	-
CASystems	string(1024)	-	Comma-separated list of unique identifiers (e.g. commercial names) of supported CA systems.	-
.STBService.{i}.Capabilities.DRM.	object	-	This object describes the characteristics of the Digital Rights Management, if any.	-
DRMSystems	string(1024)	-	Comma-separated list of unique identifiers (e.g. commercial names) of supported DRM systems.	-
.STBService.{i}.Capabilities.-ServiceMonitoring.	object	-	This object describes the capabilities of the ServiceMonitoring object.	-

Name ¹	Type	Write ²	Description	Object Default ³
ServiceTypes	string	-	Comma-separated list of supported service types, each of which relates to an AV stream source. Each item is taken from the list: "IPTV" WAN-sourced IPTV stream "VoD" WAN-sourced VoD stream "IP" WAN-sourced IP stream (includes IPTV and VoD) "CAB" Cable, e.g. DVB-C, Front End stream "DTT" DTT, e.g. DVB-T, Front End stream "SAT" Satellite, e.g. DVB-S, Front End stream "PVR" PVR play-out stream Note that none of the standard service types cover LAN-sourced IP streams.	-
MaxActiveMainStreams	int[-1:]	-	Maximum number of AV Main streams for which the STB can simultaneously collect statistics. A value of -1 indicates no specific limit on the number of active streams.	-
MinSampleInterval	int[-1:]	-	Minimum sample interval in seconds that the STB MUST be able to support. A value of -1 indicates no specific minimum sample interval.	-
MaxReportSamples	int[-1:]	-	Maximum number of samples of each statistic that the STB is able to store and report. A value of -1 indicates no specific maximum number of samples.	-
HighLevelMetricNames	string(1024)	-	Comma-separated list of the names of supported high-level metrics.	-
.STBService.{i}.Capabilities- .AudienceStats.	object	-	This object describes the capabilities of the AudienceStats object.	-
MaxAudienceStatsChannels	int[-1]	-	Maximum number of channels that the STB MUST support in the AudienceStats object A value of -1 indicates no specific limit on the number of channels.	-
.STBService.{i}.Components.	object	-	Details of STB logical or physical internal components. Unlike the Capabilities object, which is read-only and which doesn't describe individual instances, the Components object contains some writeable parameters.	-
FrontEndNumberOfEntries	unsignedInt	-	Number of FrontEnd instances.	-
AudioDecoderNumberOfEntries	unsignedInt	-	Number of AudioDecoder instances.	-
VideoDecoderNumberOfEntries	unsignedInt	-	Number of VideoDecoder instances.	-
AudioOutputNumberOfEntries	unsignedInt	-	Number of AudioOutput instances.	-
VideoOutputNumberOfEntries	unsignedInt	-	Number of VideoOutput instances.	-
SCARTNumberOfEntries	unsignedInt	-	Number of SCART instances.	-
CANumberOfEntries	unsignedInt	-	Number of CA instances.	-
DRMNumberOfEntries	unsignedInt	-	Number of DRM instances.	-

Name ¹	Type	Write ²	Description	Object Default ³
.STBService.{i}.Components.-FrontEnd.{j}.	object	-	FrontEnd instance table. Each front-end instance, including vendor-specific front-end types, will normally have a sub-object that corresponds to the front-end type (e.g. DVB-T or IP).	-
Enable	boolean	W	Enables or disables this front-end.	-
Status	string	-	The status of this front-end. Enumeration of: "Disabled" "Enabled" "Error" (OPTIONAL) The "Error" value MAY be used by the CPE to indicate a locally defined error condition.	-
Name	string(256)	-	Human-readable name associated with this front-end.	-
.STBService.{i}.Components.-FrontEnd.{j}.DVBT.	object	-	DVB-T front-end details. This object will normally be present if and only if this front-end is a DVB-T front-end. Parameters are taken from the DGTVi D-Book [32].	-
.STBService.{i}.Components.-FrontEnd.{j}.DVBT.Modulation.	object	-	DVB-T modulation details.	-
Frequency	unsignedInt	W	Indicates the current frequency in kHz. The value of this parameter is undefined while installing the service list (see .Components.-FrontEnd.{j}.DVBT.Install).	-
ChannelBandwidth	string	W	Indicates the channel bandwidth in use. Enumeration of: "8MHz" "7MHz" "6MHz"	-
Constellation	string	W	Indicates the current constellation in use. Enumeration of: "QPSK" "16QAM" "64QAM"	-
HierarchicalInformation	string	W	Hierarchical parameter alpha. The higher the value of alpha, the more noise-immune the High Priority Modulation is. Enumeration of: "1" (no hierarchy) "2" "4"	-
CodeRateHP	string	W	Indicates the current code rate in use. Code rate applies to HP in case of hierarchical coding. Enumeration of: "1/2" "2/3" "3/4" "5/6" "7/8"	-

Name ¹	Type	Write ²	Description	Object Default ³
CodeRateLP	string	W	Indicates the current code rate in use. Code rate applies to LP in case of hierarchical coding. Enumeration of: "0" Coding is not hierarchical "1/2" "2/3" "3/4" "5/6" "7/8"	-
GuardInterval	string	W	Indicates the current guard interval in use. Enumeration of: "1/32" "1/16" "1/8" "1/4"	-
TransmissionMode	string	W	Indicates the number of carriers in an OFDM frame. Enumeration of: "2k" "8k"	-
SNR	unsignedInt	-	Signal/Noise ratio in the carrier band, measured in dB.	-
BER	unsignedInt	-	Bit Error Ratio before correction, expressed in multiples of 1e-6.	-
CBER	unsignedInt	-	Bit Error Ratio after correction, expressed in multiples of 1e-9.	-
Locked	boolean	-	Indicates whether or not the tuner is locked on a frequency.	-
.STBService.{i}.Components.-FrontEnd.{j}.DVBT.Install.	object	-	DVB-T service list database installation. Supports maintenance of the service list database by scanning frequencies in order to detect services. The service list database is in .Components.-FrontEnd.{i}.DVBT.ServiceListDatabase.	-
Start	boolean	W	Starts (when set to true) or stops (when set to false) scanning for services. The value is not saved in device state and is always false when read. Discovered services are merged with those already present in the service list database. This supports cases where the frequency spectrum is not continuous. In order to force a re-install, the service list database needs to be reset (see .Components.FrontEnd.{i}.DVBT.ServiceListDatabase.Reset).	-
Status	string	-	Indicates the status of the scanning process. Enumeration of: "Disabled" scanning is not in progress "Enabled" scanning is in progress "Error" (OPTIONAL) The "Error" value MAY be used by the CPE to indicate a locally defined error condition.	-

Name ¹	Type	Write ²	Description	Object Default ³
Progress	unsignedInt [0:100]	-	Progression in % of the current scan. The value is undefined when a scan is not in progress.	-
StartFrequency	unsignedInt	W	Frequency (in kHz) at which to start the scanning process.	-
StopFrequency	unsignedInt	W	Frequency (in kHz) at which to stop the scanning process.	-
.STBService.{i}.Components.-FrontEnd.{i}.DVBT.ServiceList-Database.	object	-	DVB-T service list database.	-
Reset	boolean	W	When set to true, resets the service list database. Setting it to false has no effect. The value is not saved in device state and is always false when read.	-
TotalServices	unsignedInt	-	Total number of services defined in the service list database, i.e. the number of different Dvblds in the database.	-
LogicalChannelNumberOfEntries	unsignedInt	-	Number of logical channels in the service list database.	-
.STBService.{i}.Components.-FrontEnd.{i}.DVBT.ServiceList-Database.LogicalChannel.{i}.	object	-	DVB-T logical channel information.	-
LogicalChannelNumber	unsignedInt	-	Logical channel number.	-
ServiceNumberOfEntries	unsignedInt	-	Number of services associated with this logical channel.	-
.STBService.{i}.Components.-FrontEnd.{i}.DVBT.ServiceList-Database.LogicalChannel.{i}.-Service.{i}.	object	-	DVB-T service information.	-
Dvbld	string(12)	-	DVB Triplet [18] that uniquely identifies the service in the service list database. The DVB Triplet is a 12-byte string that is a concatenation of 4 hexadecimal characters for each of the following: Original Network Id, Transport Stream Id, Service Id. For example: OriginalNetworkId = 8442 = 0x20FA TransportStreamId = 4 = 0x0004 ServiceId = 257 = 0x0101 -> Dvbld = "20FA00040101"	-
Frequency	unsignedInt	-	Transmitter frequency (in kHz) for this service.	-
BER	unsignedInt	-	Bit Error Ratio before correction, expressed in multiples of 1e-6 (indicates the quality of this service).	-
Preferred	boolean	W	When set to true, identifies this as the preferred service for this logical channel and automatically sets "Preferred" to false for other services associated with this logical channel. When set to false, cancels this service's preferred status and causes the service with the lowest BER (best received quality) to become the preferred service for this logical channel. Whenever frequencies are re-scanned, all "Preferred" parameters are implicitly reset to false, i.e. a re-scan will always use the BER to determine preferred services, and will discard any past explicit overrides.	-

Name ¹	Type	Write ²	Description	Object Default ³
.STBService.{i}.Components.-FrontEnd.{j}.DVBT.LogicalChannelConnect.	object	-	Connect to a DVB-T logical channel.	-
LogicalChannelNumber	unsignedInt	W	Logical channel number to connect to (the logical channel's preferred service will be connected to). If a ServiceMonitoring MainStream instance with ServiceType = "DTT" is currently enabled, the STB MUST ensure that performance data for this service is collected in that instance. The value is not saved in device state. The value when read is not specified and need not be the value that was most recently written.	-
.STBService.{i}.Components.-FrontEnd.{j}.DVBT.ServiceConnect.	object	-	Connect to a DVB-T service.	-
Dvbld	string(12)	W	Dvbld of the service to connect to. If a ServiceMonitoring MainStream instance with ServiceType = "DTT" is currently enabled, the STB MUST ensure that performance data for this service is collected in that instance. The value is not saved in device state. The value when read is not specified and need not be the value that was most recently written.	-
.STBService.{i}.Components.-FrontEnd.{j}.IP.	object	-	IP front-end details. This object will normally be present if and only if the front-end is an IP front-end.	-
InboundNumberOfEntries	unsignedInt	-	Number of Inbound instances.	-
OutboundNumberOfEntries	unsignedInt	-	Number of Outbound instances.	-
ActiveInboundIPStreams	unsignedInt	-	Number of Inbound AV streams currently active (i.e. with Status != "Disabled", which means audio or AV data are being processed).	-
ActiveOutboundIPStreams	unsignedInt	-	Number of Outbound AV streams currently active (i.e. with Status != "Disabled", which means audio or AV data are being processed).	-
.STBService.{i}.Components.-FrontEnd.{j}.IP.RTCP.	object	-	Parameters related to RTCP receiver report generation as defined in [13].	-
Enable	boolean	W	Enables or disables RTCP receiver report generation.	-
Status	string	-	The status of RTCP receiver report generation. Enumeration of: "Disabled" "Enabled" "Error" (OPTIONAL) The "Error" value MAY be used by the CPE to indicate a locally defined error condition.	-
TxRepeatInterval	unsignedInt[1:]	W	Transmission repeat interval, in milliseconds.	-
.STBService.{i}.Components.-FrontEnd.{j}.IP.RTPAVPF.	object	-	Parameters related to RTP AVPF configuration as defined in [15], [16].	-
Enable	boolean	W	Enables or disables RTP/AVPF receiver report generation.	-

Name ¹	Type	Write ²	Description	Object Default ³
Status	string	-	The status of RTP/AVPF receiver report generation. Enumeration of: "Disabled" "Enabled" "Error" (OPTIONAL) The "Error" value MAY be used by the CPE to indicate a locally defined error condition.	-
RetransmitTimeout	unsignedInt	W	Time in milliseconds the receiver shall wait for a retransmitted packet before considering it to be lost.	-
MinLossPackets	unsignedInt	W	Minimum number of lost packets the receiver shall wait for before generating a retransmission request.	-
MaxReportTime	unsignedInt	W	Maximum time in milliseconds the receiver shall wait between a lost packet and the generation of a retransmission request.	-
MaxLossBurst	unsignedInt	W	Maximum number of consecutive lost packets for which the receiver shall attempt recovery.	-
.STBService.{i}.Components.-FrontEnd.{j}.IP.IGMP.	object	-	Parameters that are specific to IGMP clients when changing channels.	-
Enable	boolean	W	Enables or disables all IGMP support on the receiver side of the CPE.	-
Status	string	-	Status of IGMP support on the receiver side of the CPE. Enumeration of: "Disabled" "Enabled" "Error" (OPTIONAL) The "Error" value MAY be used by the CPE to indicate a locally defined error condition.	-
MaximumNumberOfConcurrent-Groups	unsignedInt	-	Maximum number of group memberships allowed simultaneously. For example, this would need to be at least 2 in order to support a Multicast-delivered main picture and a Multicast-delivered PIP.	-
MaximumNumberOfTracked-Groups	unsignedInt	-	Maximum number of groups for which statistics will be tracked. This relates only to the maintenance of statistics, and does not affect the operation of the STB.	-
LoggingEnable	boolean	W	Controls logging of IGMP traffic. The CPE MAY choose to throttle log entries, e.g. if joins and leaves are occurring at an excessive rate.	-
DSCPMark	unsignedInt [0:63]	W	Diffserv code point to be used for outgoing IGMP packets.	-
VLANIDMark	int[-1:]	W	VLAN ID (as defined in 802.1Q) to be used for outgoing IGMP packets. A value of -1 indicates the default value is to be used. If either the VLANIDMark or Ethernet PriorityMark is greater than zero, then the outgoing frames MUST be tagged. Otherwise, the outgoing frames MAY be tagged or untagged.	-

Name ¹	Type	Write ²	Description	Object Default ³
EthernetPriorityMark	int[-1:]	W	Ethernet priority code (as defined in 802.1D) to be used for outgoing IGMP packets. A value of -1 indicates the default value is to be used. If either the VLANIDMark or EthernetPriorityMark are greater than zero, then the outgoing frames MUST be tagged. Otherwise, the outgoing frames MAY be tagged or untagged.	-
ClientVersion	string	-	The highest IGMP version supported by the IGMP Client. Enumeration of: "v1" "v2" "v3"	-
ClientRobustness	unsignedInt	W	Robustness is a way of indicating how vulnerable the network is to lost packets. IGMP can recover from robustness minus 1 lost IGMP packets. RFC 3376 specifies a default of 2.	-
ClientUnsolicitedReportInterval	unsignedInt [1:25]	W	The number of seconds between repetitions of a host's initial report of membership in a group. RFC 3376 specifies a default of 1s.	-
ClientGroupNumberOfEntries	unsignedInt	-	Number of entries in the ClientGroup table.	-
ClientGroupStatsNumberOfEntries	unsignedInt	-	Number of entries in the ClientGroupStats table.	-
.STBService.{i}.Components.-FrontEnd.{j}.IP.IGMP.ClientGroup.{i}.	object	-	IGMP Group table. This is the database that is maintained by the IGMP client to reflect the current group membership. The CPE MUST support at least MaximumNumberOfConcurrentGroups table entries.	-
GroupAddress	string	-	The IP multicast group address for which this entry contains information.	-
UpTime	unsignedInt	-	Time elapsed in seconds since the multicast group has been a member on this interface.	-
.STBService.{i}.Components.-FrontEnd.{j}.IP.IGMP.ClientGroupStats.{i}.	object	-	IGMP Group statistics table. This stores per-group historical statistical data. The CPE MUST support at least MaximumNumberOfTrackedGroups table entries. The CPE MAY reclaim entries based on the oldest membership reports.	-
GroupAddress	string	-	The IP multicast group address for which this entry contains information.	-
TotalStart	unsignedInt	-	Number of seconds since the beginning of the period used for collection of Total statistics. Statistics SHOULD continue to be accumulated across CPE reboots, though this may not always be possible.	-
CurrentDayStart	unsignedInt	-	Number of seconds since the beginning of the period used for collection of CurrentDay statistics. The CPE MAY align the beginning of each CurrentDay interval with days in the UTC time zone, but is not required to do so. Statistics SHOULD continue to be accumulated across CPE reboots, though this may not always be possible.	-

Name ¹	Type	Write ²	Description	Object Default ³
QuarterHourStart	unsignedInt	-	Number of seconds since the beginning of the period used for collection of QuarterHour statistics. The CPE MAY align the beginning of each QuarterHour interval with real-time quarter-hour intervals, but is not required to do so. Statistics SHOULD continue to be accumulated across CPE reboots, though this may not always be possible.	-
.STBService.{i}.Components.-FrontEnd.{i}.IP.IGMP.ClientGroup-Stats.{i}.Total.	object	-	Total statistics for this multicast group.	-
NumberOfJoins	unsignedInt	-	Number of IGMP joins during this period.	-
NumberOfLeaves	unsignedInt	-	Number of IGMP leaves during this period.	-
MaxJoinDelay	unsignedInt	-	Maximum latency delay, in milliseconds, from the sending of a Join message for this group to the arrival of the first IP packet for this group during this period.	-
.STBService.{i}.Components.-FrontEnd.{i}.IP.IGMP.ClientGroup-Stats.{i}.CurrentDay.	object	-	Current day statistics for this multicast group.	-
NumberOfJoins	unsignedInt	-	Number of IGMP joins during this period.	-
NumberOfLeaves	unsignedInt	-	Number of IGMP leaves during this period.	-
MaxJoinDelay	unsignedInt	-	Maximum latency delay, in milliseconds, from the sending of a Join message for this group to the arrival of the first IP packet for this group during this period.	-
.STBService.{i}.Components.-FrontEnd.{i}.IP.IGMP.ClientGroup-Stats.{i}.QuarterHour.	object	-	Current quarter hour statistics for this multicast group.	-
NumberOfJoins	unsignedInt	-	Number of IGMP joins during this period.	-
NumberOfLeaves	unsignedInt	-	Number of IGMP leaves during this period.	-
MaxJoinDelay	unsignedInt	-	Maximum latency delay, in milliseconds, from the sending of a Join message for this group to the arrival of the first IP packet for this group during this period.	-
.STBService.{i}.Components.-FrontEnd.{i}.IP.Dejittering.	object	-	Parameters related to the de-jittering buffer configuration for all inbound IP streams.	-
BufferSize	unsignedInt	W	De-jittering buffer size, measured in milliseconds, on the receiver side of the CPE. The STB is responsible for converting this into the actual buffer size in bytes.	-
BufferInitialLevel	unsignedInt	W	Describes the number of milliseconds worth of data that should be in the receiver side dejittering buffer before play-out can start. Typically this would be half of the buffer size.	-
.STBService.{i}.Components.-FrontEnd.{i}.IP.Inbound.{i}.	object	-	Inbound IP streams currently entering the STB via this front-end.	-
Status	string	-	The status of this Inbound stream. Enumeration of: "Disabled" "Enabled" "Error" (OPTIONAL) The "Error" value MAY be used by the CPE to indicate a locally defined error condition.	-

Name ¹	Type	Write ²	Description	Object Default ³
StreamingControlProtocol	string	-	Network protocol currently used for controlling streaming of the source content, or an empty string if the content is not being streamed or is being streamed but is not being controlled. If non-empty, the string MUST be one of the .Capabilities.FrontEnd.IP.StreamingControl-Protocols values.	-
StreamingTransportProtocol	string	-	Network protocol currently used for streaming the source content, or an empty string if the content is not being streamed. If non-empty, the string MUST be one of the .Capabilities.FrontEnd.IP.StreamingTransport-Protocols values.	-
StreamingTransportControlProtocol	string	-	Transport control protocol currently used for streaming the source content, or an empty string if the content is not being streamed. If non-empty, the string MUST be one of the .Capabilities.FrontEnd.IP.StreamingTransportControlProtocols values.	-
MultiplexType	string	-	AV multiplex type currently used for the source content, or an empty string if the content is not being demultiplexed. If non-empty, the string MUST be one of the .Capabilities.FrontEnd.IP.MultiplexTypes values.	-
DownloadTransportProtocol	string	-	Network protocol currently used for downloading the source content, or an empty string if the content is not being downloaded. If non-empty, the string MUST be one of the .Capabilities.FrontEnd.IP..DownloadTransportProtocols values.	-
SourceAddress	string	-	IP address of the source of the current stream content.	-
SourcePort	unsignedInt [0:65535]	-	TCP or UDP port number of the source of the current stream content, or 0 if the content is not being delivered via IP or if not applicable.	-
DestinationAddress	string	-	IP address of the destination of the current stream content, or an empty string if the content is not being delivered via IP.	-
DestinationPort	unsignedInt [0:65535]	-	TCP or UDP port number of the destination of the current stream content, or 0 if the content is not being delivered via IP or if not applicable.	-
URI	string(256)	-	RFC 3986 URI that indicates the current source (possibly including Multicast group and port, if relevant) of the stream content, or an empty string if the source is not known or cannot be represented as a URI. For example, for a WAN-delivered stream this might be a content provider's URL, or for a LAN-delivered stream it might be an UPnP AV URN. If specified, the URI MUST be consistent with the other source-related parameters.	-

Name ¹	Type	Write ²	Description	Object Default ³
.STBService.{i}.Components.-FrontEnd.{j}.IP.Outbound.{i}.	object	-	Parameters related to an outbound IP stream.	-
Status	string	-	The status of this Outbound stream. Enumeration of: "Disabled" "Enabled" "Error" (OPTIONAL) The "Error" value MAY be used by the CPE to indicate a locally defined error condition.	-
MultiplexType	string	-	AV multiplex type currently used for the destination content, or an empty string if the content is not being demultiplexed. If non-empty, the string MUST be one of the .Capabilities.FrontEnd.IP.MultiplexTypes values.	-
URI	string(256)	-	RFC 3986 URI that indicates the current destination (possibly including Multicast group and port, if relevant) of the stream content, or an empty string if the destination is not known or cannot be represented as a URI. For example, this might be an UPnP AV URN.	-
.STBService.{i}.Components.-FrontEnd.{j}.IP.ServiceConnect.	object	-	Parameters used to force connection to a dedicated IP stream for test purposes.	-

Name ¹	Type	Write ²	Description	Object Default ³
URI	string(256)	W	<p>RFC 3986 URI that describes the service to connect to in the form:</p> <pre><trsp>://<addr>:<port>?StrCtrl=<strctrl>-?StrTrspCtrl=<trspctrl>?MuxType=<muxtype></pre> <p><addr> is the IP address of the service source (for Unicast) or IGMP group (for Multicast).</p> <p><port> is the associated TCP/UDP port. If not specified, the STB is assumed to know and use an appropriate default port.</p> <p><trsp> is the streaming transport protocol and MUST be one of the values in .Capabilities.-FrontEnd{i}.IP.StreamingTransportProtocols, which SHOULD be converted to lower-case.</p> <p><strctrl> is the streaming control protocol and MUST be one of the values in .Capabilities.-FrontEnd{i}.IP.StreamingControlProtocols. If not specified, no streaming control protocol is to be used.</p> <p><trspctrl> is the streaming transport control protocol and MUST be one of the values in .Capabilities.FrontEnd{i}.IP.StreamingTransportControlProtocols. If not specified, no streaming transport control protocol is to be used.</p> <p><muxtype> is the multiplex type and MUST be one of the values in .Capabilities.FrontEnd.IP.-MultiplexTypes. This tells the STB which multiplex type to expect. If not specified, the STB is assumed to know, or to be able to determine, the multiplex type.</p> <p>Note that download protocols are not supported by the above syntax. However, the specified syntax is just a baseline, and STBs MAY support an extended syntax.</p> <p>Example of Multicast MPEG2-TS over RTP with no transport control:</p> <pre>rtp://224.112.15.18:3257?StrCtrl=IGMP-?MuxType=MPEG2-TS</pre> <p>Example of Multicast MPEG2-TS over RTP with AL-FEC transport control:</p> <pre>rtp://224.112.15.18:3257?StrCtrl=IGMP-?StrTrspCtrl=AL-FEC?MuxType=MPEG2-TS</pre> <p>Example of Unicast MPEG2-TS over UDP with RTSP transport control:</p> <pre>udp://174.146.35.12:3257?StrCtrl=RTSP-?MuxType=MPEG2-TS</pre> <p>Example of Unicast raw AV over RTP with RTCP control and RTSP transport control:</p> <pre>rtp://174.146.35.12:3257?StrCtrl=RTSP-?StrTrspCtrl=RTCP?MuxType=None</pre> <p>If a ServiceMonitoring MainStream instance with ServiceType = "IP" or "IPTV" (or "VoD" if the URI accesses a VoD service) is currently enabled, the STB MUST ensure that performance data for this service is collected in that instance.</p> <p>The value is not saved in device state. The value when read is not specified and need not be the value that was most recently written</p>	-

Name ¹	Type	Write ²	Description	Object Default ³
.STBService.{i}.Components.PVR.	object	-	PVR details. This is a read-only object that exists only to reference PVR storage objects.	-
StorageNumberOfEntries	unsignedInt	-	Number of Storage instances.	-
.STBService.{i}.Components.-PVR.Storage.{i}.	object	-	PVR storage table. There MUST be an instance for each storage object that can be accessed by the PVR. Entries MAY be dynamically created and deleted as storage devices become available and unavailable.	-
Reference	string(256)	-	References the corresponding StorageService [4] instance, or an object contained within such an instance, e.g. a PhysicalMedium, LogicalVolume or Folder instance. The value is the full hierarchical name of the corresponding object. Example: "Device.-Services.StorageService.1".	-
.STBService.{i}.Components.-AudioDecoder.{i}.	object	-	Audio decoder instance table. It contains data representing the current status of the Audio decoder.	-
Enable	boolean	W	Enables or disables this audio decoder.	-
Status	string	-	The status of this audio decoder. Enumeration of: "Disabled" "Enabled" "Error" (OPTIONAL) The "Error" value MAY be used by the CPE to indicate a locally defined error condition.	-
Name	string(256)	-	Human-readable name associated with this audio decoder.	-
AudioStandard	string	-	Audio standard currently being processed by this audio decoder, or an empty string if no audio standard is currently being processed. If non-empty, the value MUST one of the .Capabilities.AudioDecoder.AudioStandards values.	-
.STBService.{i}.Components.-VideoDecoder.{i}.	object	-	Video decoder instance table. It contains data representing the current status of the video decoder.	-
Enable	boolean	W	Enables or disables this video decoder.	-
Status	string	-	The status of this video decoder. Enumeration of: "Disabled" "Enabled" "Error" (OPTIONAL) The "Error" value MAY be used by the CPE to indicate a locally defined error condition.	-
Name	string(256)	-	Human-readable name associated with this video decoder.	-
MPEG2Part2	string(256)	-	Path name of the MPEG2 Part 2 profile and level object instance. The value MUST be an empty string (if MPEG2 Part 2 is not used), be of the form ".Capabilities.VideoDecoder.MPEG2Part2.-ProfileLevel.{i}", or else be a full path name. For example: .Capabilities.VideoDecoder.MPEG2Part2.-ProfileLevel.2	-

Name ¹	Type	Write ²	Description	Object Default ³
MPEG4Part2	string(256)	-	Path name of the MPEG4 Part 2 profile and level object instance. The value MUST be an empty string (if MPEG4 Part 2 is not used), be of the form ".Capabilities.VideoDecoder.MPEG4Part2.-ProfileLevel.{i}", or else be a full path name. For example: .Capabilities.VideoDecoder.MPEG4Part2.-ProfileLevel.2	-
MPEG4Part10	string(256)	-	Path name of the MPEG4 Part 10 profile and level object instance. The value MUST be an empty string (if MPEG4 Part 10 is not used), be of the form ".Capabilities.VideoDecoder.MPEG4Part10.-ProfileLevel.{i}", or else be a full path name. For example: .Capabilities.VideoDecoder.MPEG4Part10.-ProfileLevel.2	-
SMPTEVC1	string(256)	-	Path name of the SMPTE-VC-1 profile and level object instance. The value MUST be an empty string (if SMPTE-VC-1 is not used), be of the form ".Capabilities.VideoDecoder.SMPTEVC1.-ProfileLevel.{i}", or else be a full path name. For example: .Capabilities.VideoDecoder.SMPTEVC1.-ProfileLevel.2	-
ContentAspectRatio	string	-	Indicates the native aspect ratio of the content available at this decoder. Enumeration of: "4:3" "16:9"	-
.STBService.{i}.Components.-AudioOutput.{i}	object	-	Audio output instance table.	-
Enable	boolean	W	Enables or disables this audio output.	-
Status	string	-	The status of this audio output. Enumeration of: "Disabled" "Enabled" "Muted" (i.e. Enabled and Muted) "Error" (OPTIONAL) The "Error" value MAY be used by the CPE to indicate a locally defined error condition.	-
Name	string(256)	-	Human-readable name associated with this audio output.	-
AudioFormat	string	-	Currently active audio output format, or an empty string if no audio output format is currently active. If non-empty, the string MUST be one of the .Capabilities.AudioOutput.AudioFormats values. Note that this parameter refers to output formats and not to physical connectors.	-

Name ¹	Type	Write ²	Description	Object Default ³
AudioLevel	unsignedInt [0:100]	W	The audio level on this audio output as a percentage (0 is quietest and 100 is loudest). This setting is independent of the mute status. For some audio formats, the audio level is not meaningful, in which case this parameter SHOULD have a fixed value of 0.	-
CancelMute	boolean	W	When set to true, cancels this audio output's mute status. Setting it to false has no effect. The value is not saved in device state and is always false when read.	-
SCARTs	string(1024)	-	Path names of the SCART object instances associated with this audio output. The value MUST be an empty string (if not associated with any SCART object instances) or else be a comma-separated list of path names, each of which is either of the form ".Components.SCART.{i}", or else is a full path name. For example: .Components.SCART.2 .Components.SCART.2,.Components.SCART.3	-
.STBService.{i}.Components.VideoOutput.{i}	object	-	Video output instance table.	-
Enable	boolean	W	Enables or disables this video output.	-
Status	string	-	The status of this video output. Enumeration of: "Disabled" "Enabled" "Error" (OPTIONAL) The "Error" value MAY be used by the CPE to indicate a locally defined error condition.	-
Name	string(256)	-	Human-readable name associated with this Video output.	-
CompositeVideoStandard	string	-	The type of analog video standard currently in use or an empty string if no analog video standard is currently in use. If non-empty, the string MUST be one of the .Capabilities.VideoOutput.CompositeVideoStandards values. Note that this parameter applies only to the CVBS output format.	-
VideoFormat	string	W	Currently active video output format, or an empty string if no video output format is currently active. If non-empty, the string MUST be one of the .Capabilities.VideoOutput.VideoFormats values. Note that this parameter refers to output formats and not to physical connectors.	-

Name ¹	Type	Write ²	Description	Object Default ³
AspectRatioBehaviour	string	-	Indicates the aspect ratio behaviour for display on monitors with an aspect ratio different from that of the content, which is specified in "ContentAspectRatio" (e.g. reproduction of HD content on SD monitors and vice versa). Enumeration of: "None" "Letterbox" "CenterCutOut" "Combined" (both of the above)	-
Macrovision	boolean	-	Indicates whether Macrovision analog protection is being used on this video output. The Macrovision version supported is indicated in .Capabilities.VideoOutput.-Macrovision.	-
HDCP	boolean	-	Indicates whether HDCP is being used on this video output. The HDCP version supported is indicated in Capabilities.VideoOutput.HDCP.	-
SCARTs	string(1024)	-	Path names of the SCART object instances associated with this video output. The value MUST be an empty string (if not associated with any SCART object instances) or else be a comma-separated list of path names, each of which is either of the form ".Components.SCART.{i}", or else is a full path name. For example: .Components.SCART.2 .Components.SCART.2,.Components.SCART.3	-
.STBService.{i}.Components.-SCART.{i}.	object	-	SCART configuration. This object allows configuration of the STB's AV switching.	
Enable	boolean	W	Enables or disables this SCART connector.	-
Status	string	-	The status of this SCART connector. Enumeration of: "Disabled" "Enabled" "Error" (OPTIONAL) The "Error" value MAY be used by the CPE to indicate a locally defined error condition.	-
Name	string(256)	-	Human-readable name associated with this connector.	-
Presence	boolean	W	Enables or disables the Presence control signal (pin 8 of the SCART connector), an output from the STB to control the display device connected to the SCART. When enabled (set to true) the STB MUST drive pin 8 to the appropriate level to indicate the output video format (SD or HD) to the connected device. When disabled (set to false) the STB MUST NOT drive pin 8.	-
.STBService.{i}.Components.-CA.{i}.	object	-	This object describes the characteristics of the Conditional Access, if any.	-

Name ¹	Type	Write ²	Description	Object Default ³
Enable	boolean	W	Enables or disables this CA system.	-
Status	string	-	The status of this CA system. Enumeration of: "Disabled" "Enabled" "Error" (OPTIONAL) The "Error" value MAY be used by the CPE to indicate a locally defined error condition.	-
Name	string(256)	-	Indicates a unique identifier (e.g. the commercial name: "La 7 più", "Mediaset Premium") for this CA system. This name MUST appear in the .Capabilities.CA.CA-Systems list.	-
SmartCardReader	string(256)	-	Path name of the smart card reader object instance containing the smart card (if any) associated with this CA system. The value MUST be an empty string (if not associated with a smart card) or else be the full path name of the corresponding smart card reader object. For example: Device.SmartCardReader.1	-
.STBService.{i}.Components.-DRM.{i}.	object	-	This object describes the characteristics of the Digital Rights Management, if any.	-
Enable	boolean	W	Enables or disables this DRM system.	-
Status	string	-	The status of this DRM system. Enumeration of: "Disabled" "Enabled" "Error" (OPTIONAL) The "Error" value MAY be used by the CPE to indicate a locally defined error condition.	-
Name	string(256)	-	Indicates a unique identifier (e.g. the commercial name: "La 7 più", "Mediaset Premium") for this DRM system. This name MUST appear in the .Capabilities.DRM.DRM-Systems list.	-
SmartCardReader	string(256)	-	Path name of the smart card reader object instance containing the smart card (if any) associated with this DRM system. The value MUST be an empty string (if not associated with a smart card) or else be the full path name of the corresponding smart card reader object. For example: Device.SmartCardReader.1	-
.STBService.{i}.AVStreams.	object	-	AV Streams object. If more than one AV stream can be active at a time, it may contain several AVStream instances.	-
ActiveAVStreams	unsignedInt	-	Number of AV streams currently active (i.e. with Status != "Disabled", which means audio or AV data are being processed).	-
AVStreamNumberOfEntries	unsignedInt	-	Number of AVStream instances.	-

Name ¹	Type	Write ²	Description	Object Default ³
.STBService.{i}.AVStreams.-AVStream.{i}.	object	-	Details of each AVStream. AV streams are created statically. Each AV stream corresponds to a valid {FrontEnd, Audio-Decoder, VideoDecoder} instance combination (although if the PVR is active for this stream, only {FrontEnd} or {Audio-Decoder, VideoDecoder} will be in use).	-
Status	string	-	The status of this AV stream. Enumeration of: "Disabled" "Enabled" "Error_PVRWriteFailure" "Error_PVRReadFailure" "Error" Unspecified error (OPTIONAL) An AV stream is disabled if any of the referenced objects are disabled. If an AV stream is disabled then the values of other AV stream parameters are not significant. The "Error" value MAY be used by the CPE to indicate a locally defined error condition.	-
Name	string(256)	-	Human-readable name associated with this stream, e.g. read from the DVB service information metadata.	-
PVRState	string	-	Indicates whether this stream is using the PVR and, if so, the PVR state. Enumeration of: "Disabled" PVR not used by this stream "Stopped" PVR playback not active "Play" PVR playback active (and playing) "Paused" PVR playback active (paused) "FastForward" PVR playback active (fast forward) "Rewind" PVR playback active (rewind) PVR record state can be inferred from whether the stream has an input FrontEnd: if so and PVRState is not Disabled, the PVR is recording.	-
FrontEnd	string(256)	-	Path name of the input FrontEnd object instance associated with this AV stream. The value MUST be an empty string (if not associated with a FrontEnd object), be of the form ".Components.FrontEnd.{i}", or else be a full path name. For example: .Components.FrontEnd.2	-
Inbound	string(256)	-	Path name of the inbound IP stream object instance associated with the FrontEnd for this AV stream. The value MUST be an empty string (if the inbound stream is not delivered over IP), be of the form ".Components.FrontEnd.{i}.IP.Inbound.{i}", or else be a full path name. For example: .Components.FrontEnd.2.Inbound.3	-

Name ¹	Type	Write ²	Description	Object Default ³
Outbound	string(256)	-	Path name of the outbound IP stream object instance associated with the output FrontEnd for this AV stream (which is always the IP FrontEnd). The value MUST be an empty string (if the outbound stream is not delivered over IP), be of the form ".Components.FrontEnd.{i}.-IP.Outbound.{i}", or else be a full path name. For example: .Components.FrontEnd.2.Outbound.3	-
AudioDecoder	string(256)	-	Path name of the Audio Decoder object instance associated with this AV stream. The value MUST be an empty string (if not associated with an Audio Decoder object), be of the form ".Components.AudioDecoder.{i}", or else be a full path name. For example: .Components.AudioDecoder.2	-
VideoDecoder	string(256)	-	Path name of the Video Decoder object instance associated with this AV stream. The value MUST be an empty string (if not associated with a Video Decoder object), be of the form ".Components.VideoDecoder.{i}", or else be a full path name. For example: .Components.VideoDecoder.2	-
CA	string(256)	-	Path name of the CA system object instance associated with this AV stream. The value MUST be an empty string (if not associated with a CA object), be of the form ".Components.CA.{i}", or else be a full path name. For example: .Components.CA.2	-
DRM	string(256)	-	Path name of the DRM system object instance associated with this AV stream. The value MUST be an empty string (if not associated with a DRM object), be of the form ".Components.DRM.{i}", or else be a full path name. For example: .Components.DRM.2	-
.STBService.{i}.AVPlayers.	object	-	AV Players object. If more than one AV player can be active at a time, it may contain several AVPlayer instances. AV players are created statically. Each AV player corresponds to a combination of several AV streams (one main screen video, several PIP, one audio) being connected to an AV presentation device (such as a TV set). There is no guarantee that the STB has the resources to enable all of its AV players at the same time.	-
ActiveAVPlayers	unsignedInt	-	Number of AV players currently active (i.e. with Status != "Disabled", which means the AVPlayer is processing an audio or AV stream).	-
AVPlayerNumberOfEntries	unsignedInt	-	Number of AVPlayer instances.	-

Name ¹	Type	Write ²	Description	Object Default ³
PreferredAudioLanguage	string(64)	W	Human-readable string, as specified in [10], describing the audio language that SHOULD be used by the STB when receiving a multi-language audio multiplex. An empty string indicates that there is no preferred audio language.	-
PreferredSubtitlingLanguage	string(64)	W	Human-readable string, as specified in [10], describing the subtitling language that SHOULD be used by the STB when receiving multi-language subtitling. A value of "None" indicates that no subtitling is required. An empty string indicates that there is no preferred subtitling language.	-
.STBService.{i}.AVPlayers.-AVPlayer.{i}.	object	-	List of AV Player objects. Each item is one particular AV player being processed by the STB.	-
Enable	boolean	W	Enables or disables this AV player.	-
Status	string	-	The status of this AV player. Enumeration of: "Disabled" "Enabled" "Error" (OPTIONAL) If an AV player is disabled then the values of other AV player parameters are not significant. The "Error" value MAY be used by the CPE to indicate a locally defined error condition.	-
Name	string(256)	-	Human-readable name associated with this AV player.	-
AudioLanguage	string(64)	-	Human-readable string, as specified in [10], describing the audio language being currently processed by the AV player.	-
SubtitlingStatus	string	-	The status of subtitling for this AV player. Enumeration of: "Disabled" "Enabled" "Error" (OPTIONAL) Subtitles are assumed to be shown only on the main screen. The "Error" value MAY be used by the CPE to indicate a locally defined error condition.	-
SubtitlingLanguage	string(64)	-	Human-readable string, as specified in [10], describing the subtitling language being currently processed by the AV player. Subtitles are assumed to be shown only on the main screen.	-
AudioOutputs	string(1024)	-	Path names of the audio output object instances associated with this AV player. The value MUST be an empty string (if not associated with any audio output object instances) or else be a comma-separated list of path names, each of which is either of the form ".Components.AudioOutput.{i}", or else is a full path name. For example: .Components.AudioOutput.2	-

Name ¹	Type	Write ²	Description	Object Default ³
VideoOutputs	string(1024)	-	<p>Path names of the video output object instances associated with this AV player.</p> <p>The value MUST be an empty string (if not associated with any video output object instances) or else be a comma-separated list of the path names, each of which is either of the form ".Components.VideoOutput.{i}", or else is a full path name. For example: .Components.VideoOutput.2</p>	-
MainStream	string(256)	-	<p>Path name of the AV stream object instance associated with this AV player that is being sent to the main screen.</p> <p>The value MUST be an empty string (if not associated with an AV stream object), be of the form ".AVStreams.AVStream.{i}", or else be a full path name. For example: .AVStreams.AVStream.2</p>	-
PIPStreams	string(1024)	-	<p>Path names of the AV stream object instances associated with this AV player that are being displayed as Picture In Picture (PIP).</p> <p>The value MUST be an empty string (if not associated with any PIP AV stream objects) or else be a comma-separated list of the path names, each is which is either of the form ".AVStreams.AVStream.{i}", or else is a full path name. For example: .AVStreams.AVStream.2 .AVStreams.AVStream.2,.AVStreams.AV-Stream.3</p>	-

Name ¹	Type	Write ²	Description	Object Default ³
.STBService.{i}.ServiceMonitoring.	object	-	<p>Contains statistics relating to the QoS / QoE of Main AV streams. These statistics are broken down into categories, e.g. de-jittering and RTP, and for each category there are two types of statistics:</p> <ul style="list-style-type: none"> Total statistics, which count the number of occurrences of something. Sample statistics, which are measured over a sample interval, and are made available to the ACS as a list of the most recent n samples. <p>Total statistics are always collected (collection cannot be disabled, but the values can be reset).</p> <p>For Sample statistics, an STB uses a single sample interval and number of samples (n) for all MainStream instances. The sample interval can be aligned with absolute time.</p> <p>When Sample statistics collection is enabled, statistics collection MUST proceed regardless of whether there are any active Main AV streams. For example, when the STB is in Standby mode, there will be no active Main AV streams, but sample intervals will continue to be timed as normal.</p> <p>Refer to section I.3 for non-normative explanation of the operation of the ServiceMonitoring object.</p>	-
SampleEnable	boolean	W	<p>Enables or disables collection of Sample statistics.</p> <p>When collection of Sample statistics is enabled, any stored samples are discarded, and the first sample interval begins immediately.</p>	-
SampleState	string	-	<p>Indicates availability of Sample statistics. Enumeration of:</p> <ul style="list-style-type: none"> "Disabled" Collection is disabled "Enabled" Collection is enabled "Trigger" Collection is enabled and the ACS should now fetch the collected data <p>The Trigger value is only used for triggering the ACS to fetch the collected data and can only be used when FetchSamples is in the range [1:ReportSamples].</p> <p>The transition from Enabled -> Trigger -> Enabled MUST be instantaneous and so will result in only a single value change for notification purposes.</p>	-

Name ¹	Type	Write ²	Description	Object Default ³
SampleInterval	unsignedInt[1:]	W	<p>The sample interval in seconds. Each statistic is measured over this sample interval.</p> <p>The STB MAY reject a request to set SampleInterval to less than .Capabilities.-ServiceMonitoring.MinSampleInterval.</p> <p>Sample intervals MUST begin every SampleInterval seconds, with no delay between samples.</p> <p>If SampleInterval is changed while collection of Sample statistics is enabled, any stored samples are discarded, and the first sample interval begins immediately.</p> <p>For example, if ReportSamples is 24 and SampleInterval is 3600 (an hour), the STB can store up to a day's worth of samples for each statistic.</p>	-
ReportSamples	unsignedInt[1:]	W	<p>The number of samples that the STB can store and report for each statistic.</p> <p>The STB MUST permit ReportSamples to be set to at least .Capabilities.ServiceMonitoring.-MaxReportSamples.</p> <p>If ReportSamples is changed while collection of Sample statistics is enabled, the STB will truncate or extend its statistics buffers as appropriate, but statistics collection MUST NOT otherwise be affected.</p> <p>For example, if ReportSamples is 24 and SampleInterval is 3600 (an hour), the STB can store up to a day's worth of samples for each statistic.</p>	-
FetchSamples	unsignedInt	W	<p>If collection of Sample statistics is enabled and FetchSamples is in the range [1:ReportSamples] then SampleState MUST transition from Enabled -> Trigger -> Enabled on completion of every FetchSamples sample intervals. Otherwise, this transition MUST NOT occur.</p> <p>For example, if ReportSamples is 25 and FetchSamples is 24, then the STB will store 25 values for each statistic and the above SampleState transition will occur every 24 sample intervals, which means that the ACS could delay for up to two sample intervals before reading the stored values and would still not miss any samples (see also ForceSample).</p>	-

Name ¹	Type	Write ²	Description	Object Default ³
TimeReference	dateTime	W	<p>An absolute time reference in UTC to determine when sample intervals will complete. Each sample interval MUST complete at this reference time plus or minus an integer multiple of SampleInterval.</p> <p>This time reference also determines when the SampleState Enabled -> Trigger -> Enabled transitions that are controlled by Fetch-Samples will occur. If collection of Sample statistics is enabled and FetchSamples is in the range [1:ReportSamples] then each such SampleState transition MUST occur at this reference time plus or minus an integer multiple of FetchSamples * SampleInterval (the fetch interval).</p> <p>TimeReference is used only to set the "phase" of the sample and fetch intervals. The actual value of TimeReference can be arbitrarily far into the past or future.</p> <p>If TimeReference is changed while collection of Sample statistics is enabled, any stored samples are discarded, and the first sample interval begins immediately.</p> <p>The Unknown Time value defined in [2] indicates that no particular time reference is specified. That is, the STB MAY locally choose the time reference, and is required only to adhere to the specified sample and fetch intervals.</p> <p>If absolute time is not available to the STB, its sample and fetch interval behavior MUST be the same as if the TimeReference parameter was set to the Unknown Time value.</p> <p>For example, if SampleInterval is 3600 (an hour) and if TimeReference is set to UTC midnight on some day (in the past, present, or future) then sample intervals will complete on each UTC hour (00:00, 01:00, 02:00 etc).</p> <p>If, in addition, FetchSamples is 24, then the fetch interval is 86400 (a day) and Sample-State Enabled -> Trigger -> Enabled transitions will occur every day at UTC midnight.</p> <p>Note that, if TimeReference is set to a time other than the Unknown Time, the first sample interval (which has to begin immediately) will almost certainly be shorter than SampleInterval). This is why TimeReference is defined in terms of when sample intervals complete rather than start.</p>	-

Name ¹	Type	Write ²	Description	Object Default ³
ForceSample	boolean	W	<p>When set to true, forces statistics for the current sample to be calculated and updated in the data model. Setting it to false has no effect. The value is not saved in device state and is always false when read.</p> <p>If this is the first time that ForceSample has been set to true during the current sample interval, this MUST cause a new value to be added to each of the Sample statistics comma-separated list parameters, and the ReportEndTime and all SampleSeconds parameters MUST be updated accordingly.</p> <p>If this is not the first time that ForceSample has been set to true during the current sample interval, then the new values that were added as described in the previous paragraph, and the ReportEndTime and all SampleSeconds parameters, MUST be updated accordingly.</p> <p>Note that ForceSample just provides a “sneak preview” of the current sample. It does not create a new sample and it does not interfere with the sample interval schedule.</p> <p>At the end of each sample interval, if ForceSample was set to true during the sample interval then the new values that were added as described above, and the ReportEndTime and all SampleSeconds parameters, will be updated accordingly. In other words, the partial sample data that was created when ForceSample was set to true will be updated one last time at the end of the sample interval.</p>	-
ReportStartTime	dateTime	-	The absolute time at which the sample interval for the first stored sample (for each statistic) started.	-
ReportEndTime	dateTime	-	<p>The absolute time at which the sample interval for the last stored sample (for each statistic) ended.</p> <p>If ForceSample has been used to force statistics for the current sample to be calculated and updated in the data model, then ReportEndTime MUST be updated to reflect the actual time over which stored data was collected.</p>	-
MainStreamNumberOfEntries	unsignedInt	-	Number of MainStream instances.	-

Name ¹	Type	Write ²	Description	Object Default ³
.STBService.{i}.ServiceMonitoring.MainStream.{i}	object	-	<p>List of Main AV stream objects. Each instance is associated with a specified service type and will collect statistics only for the main stream that matches that service type.</p> <p>Note that this means that streams that do not match an instance's service type MUST NOT contribute to the instance's Total and Sample statistics.</p> <p>It is up to the STB to determine which AV stream should be regarded as the Main AV stream for a given service type at any given time. For example, if two active streams are associated with the same service type, the STB might prefer streams that are associated with an AV player, or if more than one stream is associated with an AV player, the STB might prefer a player that is outputting a DVI stream to a player that is outputting an RF stream.</p> <p>Note that some parameters, e.g. .Components.FrontEnd.{i}.IP.ServiceConnect.URI, include requirements that restrict the STB's freedom to determine which AV stream should be regarded as the Main AV stream for a given service.</p>	-
Enable	boolean	W	Enables or disables collection of Total and Sample statistics for this object instance.	
Status	string	-	<p>Total and Sample statistics collection status for this object instance. Enumeration of:</p> <ul style="list-style-type: none"> "Disabled" "Enabled" "Error" (OPTIONAL) <p>The "Error" value MAY be used by the CPE to indicate a locally defined error condition.</p>	
ServiceType	string	W	<p>Service type associated with this main stream instance, or an empty string if this instance is disabled.</p> <p>If non-empty, the string MUST be one of the .Capabilities.ServiceMonitoring.ServiceTypes values.</p> <p>The STB MUST reject a request that would result in a given AV stream contributing to more than one ServiceMonitoring MainStream instance, e.g. it would reject a request to enable simultaneous collection of both IPTV and IP statistics, or to collect IPTV statistics on more than one instance.</p>	

Name ¹	Type	Write ²	Description	Object Default ³
AVStream	string(256)	-	Path name of the Main AV stream object instance currently associated with this ServiceMonitoring main stream instance. The value MUST be a path name, either of the form ".AVStreams.AVStream.{i}" or else a full path name. For example: .AVStreams.AVStream.2 Note that, during the period of time over which Sample statistics are stored, this ServiceMonitoring main stream instance might be associated with many different AVStream instances. This parameter just provides information about which AVStream is currently being monitored.	-
Gmin	unsignedInt	W	Minimum number of consecutive received packets after the end of an RTP Loss Event. A Loss Event is defined as a sequence of lost packets, possibly including <i>islands</i> of received packets. Each island consists of up to (Gmin - 1) received packets (a sequence of Gmin received packets terminates the Loss Event, and so is not an island).	-
SevereLossMinDistance	unsignedInt	W	The minimum distance required between error events before an RTP Loss Event is considered severe. Used in the calculation of SevereLossIndexCount and SevereLossIndexCountBeforeEC.	-
SevereLossMinLength	unsignedInt	W	The minimum value for the count of lost packets per RTP Loss Event before the count of lost packets is considered severe. Used in the calculation of SevereLossLengthCount and SevereLossLengthCount BeforeEC	-
.STBService.{i}.ServiceMonitoring.MainStream.{i}.Total.	object	-	Total statistics since this ServiceMonitoring main stream instance was last enabled or Total statistics were last reset.	-
Reset	boolean	W	When set to true, resets Total statistics for this ServiceMonitoring main stream instance. Setting it to false has no effect. The value is not saved in device state and is always false when read.	-
ResetTime	unsignedInt	-	Number of seconds since the Total statistics were last enabled or reset for this ServiceMonitoring main stream instance.	-
.STBService.{i}.ServiceMonitoring.MainStream.{i}.Total.-DejitteringStats.	object	-	Total de-jittering statistics for this ServiceMonitoring main stream instance.	-
Overruns	unsignedInt	-	Total number of times the receive jitter buffer has overrun for this AV stream.	-
Underruns	unsignedInt	-	Total number of times the receive jitter buffer has underrun for this AV stream.	-

Name ¹	Type	Write ²	Description	Object Default ³
.STBService.{i}.ServiceMonitoring.MainStream.{j}.Total-RTPStats.	object	-	Total RTP statistics for this ServiceMonitoring main stream instance. If no EC is applied the "BeforeEC" statistics provide the same values as the corresponding statistics, e.g. when there is no EC, PacketsReceivedBeforeEC = PacketsReceived.	-
PacketsExpected	unsignedInt	-	Total number of RTP packets expected for this AV stream as described in [13] A.3.	-
PacketsDiscarded	unsignedInt	-	Total number of packets discarded at the RTP layer for this AV stream because they were too late for playout, regardless of their being out of sequence or not. These statistics are collected when no EC is applied or after any EC if available.	-
PacketsOutOfSequence	unsignedInt	-	Total number of packets out of sequence at the RTP level for this AV stream, regardless of their being too late for playout or not. These statistics are collected when no EC is applied or after any EC if available.	-
PacketsReceived	unsignedInt	-	Total number of RTP packets received for this AV stream. These statistics are collected when no EC is applied or after any EC if available.	-
PacketsReceivedBeforeEC	unsignedInt	-	Total number of RTP packets received for this AV stream. These statistics are collected before any EC, if available, is applied.	-
PacketsLost	unsignedInt	-	Total number of RTP packets lost for this stream. These statistics are collected when no EC is applied or after any EC if available.	-
PacketsLostBeforeEC	unsignedInt	-	Total number of RTP packets lost for this stream. These statistics are collected before any EC, if available, is applied.	-
LossEvents	unsignedInt	-	Total number of Loss Events for this stream. These statistics are collected when no EC is applied or after any EC if available.	-
LossEventsBeforeEC	unsignedInt	-	Total number of Loss Events for this stream. These statistics are collected before any EC, if available, is applied.	-
SevereLossIndexCount	unsignedInt	-	Total number of Loss Events closer than SevereLossMinDistance. These statistics are collected when no EC is applied or after any EC if available.	-
SevereLossIndexCountBeforeEC	unsignedInt	-	Total number of Loss Events closer than SevereLossMinDistance. These statistics are collected before any EC, if available, is applied.	-
SevereLossLengthCount	unsignedInt	-	Total number of Loss Events longer than SevereLossMinLength. These statistics are collected when no EC is applied or after any EC if available.	-

Name ¹	Type	Write ²	Description	Object Default ³
SevereLossLengthCountBeforeEC	unsignedInt	-	Total number of Loss Events longer than SevereLossMinLength. These statistics are collected before any EC, if available, is applied.	-
RetransmitTimeouts	unsignedInt	-	Total number of RTP/AVPF retransmission timeouts ([15], [16]) If RTP/AVPF is not used this parameter does not increment	-
.STBService.{i}.ServiceMonitoring.MainStream.{i}.Total.-MPEG2TSStats.	object	-	Total MPEG2-TS statistics for this ServiceMonitoring main stream instance.	-
TSPacketsReceived	unsignedInt	-	Total number of MPEG2-TS packets received for this AV stream.	-
TSSyncByteErrorCount	unsignedInt	-	Total number of MPEG2-TS packets with an invalid transport sync byte received for this AV stream. Not meaningful when MPEG2-TS is transported over IP.	-
TSSyncLossCount	unsignedInt	-	Total number of MPEG2-TS Loss of Synchronization Events for this AV stream. Not meaningful when MPEG2-TS is transported over IP.	-
PacketDiscontinuityCounter	unsignedInt	-	Total number of MPEG2-TS Discontinuity errors that have been captured for this AV stream. This parameter accumulates all of the discontinuities observed for all currently monitored PIDs. These statistics are collected when no CA decryption scheme is in use or after any CA decryption if in use.	-
PacketDiscontinuityCounter-BeforeCA	unsignedInt	-	Total number of MPEG2-TS Discontinuity errors that have been captured for this AV stream. This parameter accumulates all of the discontinuities observed for all currently monitored PIDs. These statistics are collected before any CA decryption scheme, if used, is applied.	-
.STBService.{i}.ServiceMonitoring.MainStream.{i}.Total.-VideoDecoderStats.	object	-	Total video decoder application layer statistics for this ServiceMonitoring main stream instance.	-
FrameRate	unsignedInt	-	The frame rate at which the STB played out the video content for this AV stream.	-
DecodedFrames	unsignedInt	-	The number of video frames that were decoded completely (errorless frames) or partially (corrupted frames) for this AV stream.	-
LostFrames	unsignedInt	-	The number of video frames that could not be reproduced by the STB for this AV stream.	-
ConcealedFrames	unsignedInt	-	The number of video frames that were reproduced by the STB for which some information was missing and loss concealment was applied for this AV stream.	-
lDecodedFrames	unsignedInt	-	The number of I frames that were decoded completely (errorless frames) or partially (corrupted frames) for this AV stream.	-
lLostFrames	unsignedInt	-	The number of I frames that could not be reproduced by the STB for this AV stream.	-

Name ¹	Type	Write ²	Description	Object Default ³
IConcealedFrames	unsignedInt	-	The number of I frames that were reproduced by the STB for which some information was missing and loss concealment applied for this AV stream.	-
PDecodedFrames	unsignedInt	-	The number of P frames that were decoded completely (errorless frames) or partially (corrupted frames) for this AV stream.	-
PLostFrames	unsignedInt	-	The number of P frames that could not be reproduced by the STB for this AV stream.	-
PConcealedFrames	unsignedInt	-	The number of P frames that were reproduced by the STB for which some information was missing and loss concealment applied for this AV stream.	-
BDecodedFrames	unsignedInt	-	The number of B frames that were decoded completely (errorless frames) or partially (corrupted frames) for this AV stream.	-
BLostFrames	unsignedInt	-	The number of B frames that could not be reproduced by the STB for this AV stream.	-
BConcealedFrames	unsignedInt	-	The number of B frames that were reproduced by the STB for which some information was missing and loss concealment applied for this AV stream.	-
AVResynchCounter	unsignedInt	-	The number of AV resynchronizations that were performed by the STB for this AV stream.	-
.STBService.{i}.ServiceMonitoring.MainStream.{i}.Total.-AudioDecoderStats.	object	-	Total audio decoder application layer statistics for this ServiceMonitoring main stream instance.	-
DecodedFrames	unsignedInt	-	The number of audio frames that were decoded completely (errorless frames) or partially (corrupted frames) for this AV stream.	-
DecodingErrors	unsignedInt	-	The number of audio decoding errors detected for this AV stream.	-
.STBService.{i}.ServiceMonitoring.MainStream.{i}.Total.-VideoResponseStats.	object	-	Total video response statistics for this ServiceMonitoring main stream instance.	-
MinimumVideoSystemResponse	unsignedInt	-	Minimum time period (in milliseconds) between the instant of transmitting an IGMP JOIN command or an RTSP SETUP command, and the display of the first I frame for this AV stream.	-
MaximumVideoSystemResponse	unsignedInt	-	Maximum time period (in milliseconds) between the instant of transmitting an IGMP JOIN command or an RTSP SETUP command, and the display of the first I frame for this AV stream.	-

Name ¹	Type	Write ²	Description	Object Default ³
.STBService.{i}.ServiceMonitoring.MainStream.{j}.Sample.	object	-	<p>Sample statistics collected since collection was last enabled or reset.</p> <p>Most of the parameters in this object are comma-separated lists of individual statistics values, each of which corresponds to a sample interval.</p> <p>The statistics values in these comma-separated lists MUST be in time order, with the oldest one first and the most recent one last.</p> <p>Each comma-separated list can contain up to ReportSamples statistics values. When the list is full, each new value causes the oldest one to be discarded.</p> <p>If no data was collected for a statistic during a sample interval then the value for that sample interval MUST be zero unless otherwise specified in the description of the individual statistics parameter.</p>	-
SampleSeconds	string	-	<p>Comma-separated list; each entry is the number of seconds during which data was collected for this AV stream during the sample interval.</p> <p>Each statistics category (Dejittering, RTP etc) has its own SampleSeconds parameter, which indicates the number of seconds during which data was collected for that statistics category during the sample interval.</p> <p>Individual SampleSeconds values can be less than the SampleInterval, for several reasons, including:</p> <ul style="list-style-type: none"> The STB was in Standby mode for all or part of the sample interval. TimeReference has been set to a time other than the Unknown Time and the current sample interval started part of the way through a scheduled sample interval. ForceSample has been used to force statistics for the current sample to be calculated and updated in the data model. 	-
SignificantChanges	string	-	Comma-separated list; each entry is the total number of times that a channel change (or other potentially disruptive event) occurred for this AV stream during the sample interval.	-
PVRTimeShift	string	-	Comma-separated list; each entry is a Boolean that indicates whether PVR, with a non-zero time-shift, was used at any time for this AV stream during the sample interval.	-
HighLevelMetricStatsNumberOfEntries	unsignedInt	-	Number of HighLevelMetricStats instances.	-
.STBService.{i}.ServiceMonitoring.MainStream.{j}.Sample.-DejitteringStats.	object	-	De-jittering Sample statistics for this ServiceMonitoring main stream instance.	-

Name ¹	Type	Write ²	Description	Object Default ³
SampleSeconds	string	-	Comma-separated list; each entry is the number of seconds during which de-jittering data was collected for this AV stream during the sample interval. Individual SampleSeconds values can be less than the sample interval, for several reasons, including: Any of the reasons for which the .ServiceMonitoring.MainStream.{i}.Sample.SampleSeconds values might be less than the sample interval. De-jittering data collection is not appropriate or possible for this AV stream.	-
Overruns	string	-	Comma-separated list; each entry is the total number of times the receive jitter buffer has overrun for this AV stream during the sample interval.	-
Underruns	string	-	Comma-separated list; each entry is the total number of times the receive jitter buffer has underrun for this AV stream during the sample interval.	-
EmptyBufferTime	string	-	Comma-separated list; each entry is the time period, in milliseconds, for which the playout buffer was empty for this AV stream during the sample interval.	-
.STBService.{i}.ServiceMonitoring.MainStream.{i}.Sample.RTPStats.	object	-	RTP Sample statistics for this Service-Monitoring main stream instance. If no EC is applied the "BeforeEC" statistics provide the same values as the corresponding statistics, e.g. when there is no EC, PacketsReceivedBeforeEC = PacketsReceived.	-
SampleSeconds	string	-	Comma-separated list; each entry is the number of seconds during which RTP data was collected for this AV stream during the sample interval. Individual SampleSeconds values can be less than the sample interval, for several reasons, including: Any of the reasons for which the .ServiceMonitoring.MainStream.{i}.Sample.SampleSeconds values might be less than the sample interval. RTP data collection is not appropriate or possible for this AV stream.	-
PacketsExpected	string	-	Comma-separated list; each entry is the total number of RTP packets expected for this AV stream during the sample interval as described in [13] A.3.	-
PacketsDiscarded	string	-	Comma separated list; each entry is the number of packets that were discarded at the RTP layer for this AV stream during the sample interval because they were too late for playout, regardless of their being out of sequence or not. These statistics are collected when no EC is applied or after any EC if available.	-

Name ¹	Type	Write ²	Description	Object Default ³
PacketsOutOfSequence	string	-	Comma-separated list; each entry is the number of packets out of sequence at the RTP level for this AV stream during the sample interval, regardless of their being too late for playout or not. These statistics are collected when no EC is applied or after any EC if available.	-
PacketsReceived	string	-	Comma-separated list; each entry is the total number of RTP packets received for this AV stream during the sample interval. These statistics are collected when no EC is applied or after any EC if available.	-
PacketsReceivedBeforeEC	string	-	Comma-separated list; each entry is the total number of RTP packets received for this AV stream during the sample interval. These statistics are collected before any EC, if available, is applied.	-
RetransmitTimeouts	string	-	Comma-separated list; each entry is the number of RTP/AVPF retransmission timeouts ([15], [16]) for this AV stream during the sample interval.	-
PacketsLost	string	-	Comma-separated list; each entry is the total number of RTP packets lost for this AV stream during the sample interval. Calculated by looking for missing RTP sequence headers. These statistics are collected when no EC is applied or after any EC if available.	-
PacketsLostBeforeEC	string	-	Comma-separated list; each entry is the total number of RTP packets lost for this AV stream during the sample interval. Calculated by looking for missing RTP sequence headers. These statistics are collected before any EC, if available, is applied.	-
LossEvents	string	-	Comma-separated list; each entry is the total number of Loss Events for this AV stream during the sample interval. These statistics are collected when no EC is applied or after any EC if available.	-
LossEventsBeforeEC	string	-	Comma-separated list; each entry is the total number of Loss Events for this AV stream during the sample interval. These statistics are collected before any EC, if available, is applied.	-
SevereLossIndexCount	string	-	Comma-separated list; each entry is the total number of Loss Events that occurred at a distance shorter than SevereLossMinDistance from the previous Loss Event for this AV stream during the sample interval. These statistics are collected when no EC is applied or after any EC if available.	-

Name ¹	Type	Write ²	Description	Object Default ³
SevereLossIndexCountBeforeEC	string	-	Comma-separated list; each entry is the total number of Loss Events that occurred at a distance shorter than SevereLossMinDistance from the previous Loss Event for this AV stream during the sample interval. These statistics are collected before any EC, if available, is applied.	-
AverageLossDistance	string	-	Comma-separated list; each entry is the average distance (in RTP packets) between consecutive Loss Events for this AV stream during the sample interval. These statistics are collected when no EC is applied or after any EC if available.	-
AverageLossDistanceBeforeEC	string	-	Comma-separated list; each entry is the average distance in RTP packets between consecutive loss events for this AV stream during the sample interval. These statistics are collected before any EC, if available, is applied.	-
MinimumLossDistance	string	-	Comma-separated list; each entry is the length in RTP packets of the shortest distance between consecutive Loss Events for this AV stream during the sample interval. These statistics are collected when no EC is applied or after any EC if available.	-
MinimumLossDistanceBeforeEC	string	-	Comma-separated list; each entry is the length in RTP packets of the shortest distance between consecutive Loss Events for this AV stream during the sample interval. These statistics are collected before any EC, if available, is applied.	-
SevereLossLengthCount	string	-	Comma-separated list; each entry is the total number of Loss Events with length in RTP packets greater than SevereLossMinLength for this AV stream during the sample interval. These statistics are collected when no EC is applied or after any EC if available.	-
SevereLossLengthCountBeforeEC	string	-	Comma-separated list; each entry is the total number of Loss Events with length in RTP packets greater than SevereLossMinLength for this AV stream during the sample interval. These statistics are collected before any EC, if available, is applied.	-
MaximumLossPeriod	string	-	Comma-separated list; each entry is the length in RTP packets of the longest Loss Event for this AV stream during the sample interval. These statistics are collected when no EC is applied or after any EC if available.	-
MaximumLossPeriodBeforeEC	string	-	Comma-separated list; each entry is the length in RTP packets of the longest Loss Event for this AV stream during the sample interval. These statistics are collected before any EC, if available, is applied.	-

Name ¹	Type	Write ²	Description	Object Default ³
AverageLossPeriod	string	-	Comma-separated list; each entry is the average length in RTP packets of Loss Events for this AV stream during the sample interval. These statistics are collected when no EC is applied or after any EC if available.	-
AverageLossPeriodBeforeEC	string	-	Comma-separated list; each entry is the average length in RTP packets of Loss Events for this AV stream during the sample interval. These statistics are collected before any EC, if available, is applied.	-
MinimumLossPeriod	string	-	Comma-separated list; each entry is the length in RTP packets of the shortest RTP Loss Event for this AV stream during the sample interval. These statistics are collected when no EC is applied or after any EC if available.	-
MinimumLossPeriodBeforeEC	string	-	Comma-separated list; each entry is the length in RTP packets of the shortest RTP Loss Event for this AV stream during the sample interval. These statistics are collected before any EC, if available, is applied.	-
.STBService.{i}.ServiceMonitoring.MainStream.{i}.Sample.-MPEG2TSStats.	object	-	MPEG2-TS Sample statistics for this Service-Monitoring main stream instance.	-
SampleSeconds	string	-	Comma-separated list; each entry is the number of seconds during which MPEG2-TS data was collected for this AV stream during the sample interval. Individual SampleSeconds values can be less than the sample interval, for several reasons, including: Any of the reasons for which the .ServiceMonitoring.MainStream.{i}.Sample.SampleSeconds values might be less than the sample interval. MPEG2-TS data collection is not appropriate or possible for this AV stream.	-
TSPacketsReceived	string	-	Comma-separated list; each entry is the total number of MPEG2-TS packets received for this AV stream during the sample interval.	-
TSPacketsDrained	string	-	Comma-separated list; each entry is the total number of good MPEG2-TS packets removed from the buffer for this AV stream during the sample interval. Packets received minus packets drained provides an indication how close the buffer came to over or under running during the sample interval.	-
TSSyncByteErrorCount	string	-	Comma-separated list; each entry is the total number of MPEG2-TS packets with an invalid transport sync byte for this AV stream during the sample interval. Not meaningful when MPEG2-TS is transported over IP.	-

Name ¹	Type	Write ²	Description	Object Default ³
TSSyncLossCount	string	-	Comma-separated list; each entry is the total number of MPEG2-TS Loss of Synchronization Events for this AV stream during the sample interval. Not meaningful when MPEG2-TS is transported over IP.	-
PacketDiscontinuityCounter	string	-	Comma-separated list; each entry is the total number of MPEG2-TS Discontinuity errors that were captured for this AV stream during the sample interval. These statistics are collected when no CA decryption scheme is in use or after any CA decryption if in use.	-
PacketDiscontinuityCounter-BeforeCA	string	-	Comma-separated list; each entry is the total number of MPEG2-TS Discontinuity errors that were captured for this AV stream during the sample interval. These statistics are collected before any CA decryption scheme, if used, is applied.	-
.STBService.{i}.ServiceMonitoring.MainStream.{i}.Sample.VideoDecoderStats.	object	-	Video decoder Sample statistics for this ServiceMonitoring main stream instance.	-
SampleSeconds	string	-	Comma-separated list; each entry is the number of seconds during which video decoder data was collected for this AV stream during the sample interval. Individual SampleSeconds values can be less than the sample interval, for several reasons, including: Any of the reasons for which the .ServiceMonitoring.MainStream.{i}.Sample.SampleSeconds values might be less than the sample interval. Video Decoder data collection is not appropriate or possible for this AV stream.	-
FrameRate	string	-	Comma-separated list; each entry is the frame rate at which the STB played out the video content for this AV stream during the sample interval.	-
DecodedFrames	string	-	Comma-separated list; each entry is the number of video frames that were decoded completely (errorless frames) or partially (corrupted frames) by the STB for this AV stream during the sample interval.	-
LostFrames	string	-	Comma-separated list; each entry is the number of video frames that could not be reproduced by the STB for this AV stream during the sample interval.	-
ConcealedFrames	string	-	Comma-separated list; each entry is the number of video frames that were reproduced by the STB for which some information was missing for this AV stream during the sample interval and loss concealment was applied.	-
IDecodedFrames	string	-	Comma-separated list; each entry is the number of I frames that were decoded completely (errorless frames) or partially (corrupted frames) by the STB for this AV stream during the sample interval.	-

Name ¹	Type	Write ²	Description	Object Default ³
ILostFrames	string	-	Comma-separated list; each entry is the number of I frames that could not be reproduced by the STB for this AV stream during the sample interval.	-
IConcealedFrames	string	-	Comma-separated list; each entry is the number of I frames that were reproduced by the STB for which some information was missing for this AV stream during the sample interval and loss concealment applied.	-
PDecodedFrames	string	-	Comma-separated list; each entry is the number of P frames that were decoded completely (errorless frames) or partially (corrupted frames) by the STB for this AV stream during the sample interval.	-
PLostFrames	string	-	Comma-separated list; each entry is the number of P frames that could not be reproduced by the STB for this AV stream during the sample interval.	-
PConcealedFrames	string	-	Comma-separated list; each entry is the number of P frames that were reproduced by the STB for which some information was missing for this AV stream during the sample interval and loss concealment applied.	-
BDecodedFrames	string	-	Comma-separated list; each entry is the number of B frames that were decoded completely (errorless frames) or partially (corrupted frames) by the STB for this AV stream during the sample interval.	-
BLostFrames	string	-	Comma-separated list; each entry is the number of B frames that could not be reproduced by the STB for this AV stream during the sample interval.	-
BConcealedFrames	string	-	Comma-separated list; each entry is the number of B frames that were reproduced by the STB for which some information was missing for this AV stream during the sample interval and loss concealment applied.	-
AVResynchCounter	string	-	Comma-separated list; each entry is the number of AV resynchronizations that were performed by the STB for this AV stream during the sample interval.	-
.STBService.{i}.ServiceMonitoring.MainStream.{i}.Sample.AudioDecoderStats.	object	-	Audio decoder Sample statistics for this ServiceMonitoring main stream instance.	-
SampleSeconds	string	-	Comma-separated list; each entry is the number of seconds during which audio decoder data was collected for this AV stream during the sample interval. Individual SampleSeconds values can be less than the sample interval, for several reasons, including: Any of the reasons for which the .ServiceMonitoring.MainStream.{i}.Sample.SampleSeconds values might be less than the sample interval. Audio decoder data collection is not appropriate or possible for this AV stream.	-

Name ¹	Type	Write ²	Description	Object Default ³
DecodedFrames	string	-	Comma-separated list; each entry is the number of audio frames that were decoded completely (errorless frames) or partially (corrupted frames) by the STB for this AV stream during the sample interval.	-
DecodingErrors	string	-	Comma-separated list; each entry is the number of audio decoding errors detected by the STB for this AV stream during the sample interval.	-
.STBService.{i}.ServiceMonitoring.MainStream.{i}.Sample.-VideoResponseStats.	object	-	Video response Sample statistics for this ServiceMonitoring main stream instance.	-
SampleSeconds	string	-	Comma-separated list; each entry is the number of seconds during which video response data was collected for this AV stream during the sample interval. Individual SampleSeconds values can be less than the sample interval, for several reasons, including: Any of the reasons for which the .ServiceMonitoring.MainStream.{i}.Sample.SampleSeconds values might be less than the sample interval. Video response data collection is not appropriate or possible for this AV stream.	-
AverageVideoSystemResponse	string	-	Comma-separated list; each entry is the time period (in milliseconds) between the instant of transmitting an IGMP JOIN or an RTSP SETUP command and the display of the first I frame, computed as the average of the VideoSystemResponse events that occurred for this AV stream during the sample interval.	-
MinimumVideoSystemResponse	unsignedInt	-	Comma-separated list; each entry is the minimum time period (in milliseconds) between the instant of transmitting an IGMP JOIN or an RTSP SETUP command and the display of the first I frame for this AV stream during the sample interval.	-
MaximumVideoSystemResponse	unsignedInt	-	Comma-separated list; each entry is the maximum time period (in milliseconds) between the instant of transmitting an IGMP JOIN or an RTSP SETUP command and the display of the first I frame for this AV stream during the sample interval.	-
.STBService.{i}.ServiceMonitoring.MainStream.{i}.Sample.-HighLevelMetricStats.{i}.	object	-	High-level metric Sample statistics for this ServiceMonitoring main stream instance. One or more different high-level metrics can be supported.	-
MetricName	string(256)	-	Indicates a unique identifier for this high-level metric. This name MUST appear in the .Capabilities.ServiceMonitoring.HighLevel-MetricNames list.	-
Metric1Threshold	unsignedInt [0:65535]	W	The threshold value that controls the calculation of the Metric1Failures parameter.	-
Metric2Threshold	unsignedInt [0:65535]	W	The threshold value that controls the calculation of the Metric2Failures parameter.	-

Name ¹	Type	Write ²	Description	Object Default ³
SampleSeconds	string	-	Comma-separated list; each entry is the number of seconds during which this high-level metric's data was collected for this AV stream during the sample interval. Individual SampleSeconds values can be less than the sample interval, for several reasons, including: Any of the reasons for which the .ServiceMonitoring.MainStream.{i}.Sample.SampleSeconds values might be less than the sample interval. This high-level metric's data collection is not appropriate or possible for this AV stream.	-
Metric1	string	-	Comma-separated list; each entry is a high-level metric calculated for this AV stream during the sample interval. The interpretation of the metric values is algorithm-specific; the only constraints are that values must be numeric, in the range [0:65535] and that lower values indicate worse QoE than do higher values.	-
Metric2	string	-	Comma-separated list; each entry is a high-level metric calculated for this AV stream during the sample interval. The interpretation of the metric values is algorithm-specific; the only constraints are that values must be numeric, in the range [0:65535] and that lower values indicate worse QoE than do higher values.	-
Metric1Failures	unsignedInt	-	Counts the number of times (since the last time this MainStream and Sample statistics collection were both enabled) that a newly-calculated value of high-level metric Metric1 was less than or equal to Metric1Threshold. This parameter can be incremented at any time during a sample interval, and might be incremented more than once during a single sample interval.	-
Metric2Failures	unsignedInt	-	Counts the number of times (since the last time this MainStream and the Sample statistics collection were both enabled) that a newly-calculated value of high-level metric Metric2 was less than or equal to Metric2-Threshold. This parameter can be incremented at any time during a sample interval, and might be incremented more than once during a single sample interval.	-
.STBService.{i}.Applications.	object	-	Applications.	-
.STBService.{i}.Applications.-AudienceStats.	object	-	Audience viewing statistics.	-
Enable	boolean	W	Enables or disables collection of viewing statistics. Enabling collection of viewing statistics also resets the viewing statistics table.	

Name ¹	Type	Write ²	Description	Object Default ³
Reset	boolean	W	When set to true resets the viewing statistics table. Setting it to false has no effect. The value is not saved in device state and is always false when read. Resetting statistics MUST reset the Duration parameter for all entries in the Channel table.	-
ResetTime	unsignedInt	-	Number of seconds since the Audience statistics were last enabled or reset.	-
ChannelNumberOfEntries	unsignedInt	-	Number of instances of Channel.	-
.STBService.{i}.Applications.- AudienceStats.Channel.{i}.	object	-	Table to record what the STB has been receiving. Time durations are recorded only for an AVPlayer's main AVStream (i.e. for TV channels being displayed in the main screen). Each entry corresponds to a given TV channel and is indexed by channel name (the instance number is chosen by the STB and is not related to the channel number).	-
Name	string(256)	-	String describing the TV channel being displayed in the main screen. MUST be the value of the corresponding .AVStreams.-AVStream.{i}.Name parameter.	-
Duration	unsignedInt	-	The cumulative duration of this channel in seconds.	-

6.1 Notification Requirements

CPE MUST support Active Notification (see [1]) for all parameters defined in the STBService data model with the exception of those parameters listed in Table 2. For only those parameters listed in Table 2, the CPE MAY reject a request by an ACS to enable Active Notification via the SetParameterAttributes RPC by responding with fault code 9009 as defined in [1] (Notification request rejected).

Table 2 includes all parameters that can continually change. This includes Total statistics, e.g. PacketsReceived changes every time a packet is received, but excludes Sample statistics, which don't change more than once per sample interval.

CPE MUST support Passive Notification (see [1]) for all parameters defined in the STBService data model, with no exceptions.

Table 2/TR-135 – Parameters for which Active Notification MAY be denied by the CPE

Parameter ⁴
.STBService.{i}.Components.FrontEnd.{i}.DVBT.Modulation.
SNR
BER
CBER
.STBService.{i}.Components.FrontEnd.{i}.DVBT.Install.

⁴ The name of a Parameter referenced in this table is the concatenation of the root object name as defined in [2], the object name shown in the yellow header, and the individual Parameter name.

Parameter ⁴
Progress
.STBService.{i}.Components.FrontEnd.{i}.DVBT.ServiceListDatabase.LogicalChannel.{i}.Service.{i}.
BER
.STBService.{i}.Components.FrontEnd.{i}.IP.IGMP.ClientGroup.{i}.
UpTime
.STBService.{i}.Components.FrontEnd.{i}.IP.IGMP.ClientGroupStats.{i}.
TotalStart
CurrentDayStart
QuarterHourStart
.STBService.{i}.Components.FrontEnd.{i}.IP.IGMP.ClientGroupStats.{i}.Total.
NumberOfJoins
NumberOfLeaves
.STBService.{i}.Components.FrontEnd.{i}.IP.IGMP.ClientGroupStats.{i}.CurrentDay.
NumberOfJoins
NumberOfLeaves
.STBService.{i}.Components.FrontEnd.{i}.IP.IGMP.ClientGroupStats.{i}.QuarterHour.
NumberOfJoins
NumberOfLeaves
.STBService.{i}.ServiceMonitoring.MainStream.{i}.Total.
ResetTime
.STBService.{i}.ServiceMonitoring.MainStream.{i}.Total.DejitteringStats.
Overruns
Underruns
.STBService.{i}.ServiceMonitoring.MainStream.{i}.Total.RTPStats.
PacketsExpected
PacketsReceived
PacketsReceivedBeforeEC
.STBService.{i}.ServiceMonitoring.MainStream.{i}.Total.MPEG2TSStats.
TSPacketsReceived
.STBService.{i}.ServiceMonitoring.MainStream.{i}.Total.VideoDecoderStats.
DecodedFrames
IDecodedFrames
PDecodedFrames
BDecodedFrames
.STBService.{i}.ServiceMonitoring.MainStream.{i}.Total.AudioDecoderStats.
DecodedFrames
.STBService.{i}.Applications.AudienceStats.
ResetTime
.STBService.{i}.Applications.AudienceStats.Channel.{i}.
Duration

7 Profile Definitions

This section specifies the profiles defined for version 1.0 of the STBService data model. The use of profiles for this data model follows the definition and usage conventions described in [2].

7.1 Notation

The following abbreviations are used to specify profile requirements:

Abbreviation	Description
R	Read support is REQUIRED.
W	Both Read and Write support is REQUIRED.
P	The object is REQUIRED to be present.
C	Creation and deletion of the object via AddObject and DeleteObject is REQUIRED.

7.2 Baseline Profile

Table 3 defines the Baseline:1 profile for the STBService:1 object. The minimum required version for this profile is STBService:1.0.

Table 3/TR-135 – Baseline:1 profile definition for STBService:1

Name	Requirement
.STBService.{i}.Capabilities.	P
MaxActiveAVStreams	R
MaxActiveAVPlayers	R
.STBService.{i}.Capabilities.PVR.	P
MaxIOStreams	R
MaxRecordingStreams	R
MaxPlaybackStreams	R
MaxTimeDelay	R
.STBService.{i}.Capabilities.AudioDecoder.	P
AudioStandards	R
.STBService.{i}.Capabilities.VideoDecoder.	P
VideoStandards	R
.STBService.{i}.Capabilities.VideoDecoder.MPEG2Part2.	P
AudioStandards	R
ProfileLevelNumberOfEntries	R
.STBService.{i}.Capabilities.VideoDecoder.MPEG2Part2.ProfileLevel.{i}.	P
Profile	R
Level	R
MaximumDecodingCapability	R
.STBService.{i}.Capabilities.VideoDecoder.MPEG4Part2.	P
AudioStandards	R
ProfileLevelNumberOfEntries	R
.STBService.{i}.Capabilities.VideoDecoder.MPEG4Part2.ProfileLevel.{i}.	P
Profile	R
Level	R
MaximumDecodingCapability	R

Name	Requirement
.STBService.{i}.Capabilities.VideoDecoder.MPEG4Part10.	P
AudioStandards	R
ProfileLevelNumberOfEntries	R
.STBService.{i}.Capabilities.VideoDecoder.MPEG4Part10.ProfileLevel.{i}.	P
Profile	R
Level	R
MaximumDecodingCapability	R
.STBService.{i}.Capabilities.VideoDecoder.SMPTEVC1.	P
AudioStandards	R
ProfileLevelNumberOfEntries	R
.STBService.{i}.Capabilities.VideoDecoder.SMPTEVC1.ProfileLevel.{i}.	P
Profile	R
Level	R
MaximumDecodingCapability	R
.STBService.{i}.Capabilities.AudioOutput.	P
AudioFormats	R
.STBService.{i}.Capabilities.VideoOutput.	P
CompositeVideoStandards	R
VideoFormats	R
Macrovision	R
HDCP	R
.STBService.{i}.Capabilities.CA.	P
CASystems	R
.STBService.{i}.Capabilities.DRM.	P
DRMSystems	R
.STBService.{i}.Capabilities.ServiceMonitoring.	P
MaxActiveMainStreams	R
MinSampleInterval	R
MaxReportSamples	R
HighLevelMetricNames	R
.STBService.{i}.Components.	P
FrontEndNumberOfEntries	R
AudioDecoderNumberOfEntries	R
VideoDecoderNumberOfEntries	R
AudioOutputNumberOfEntries	R
VideoOutputNumberOfEntries	R
.STBService.{i}.Components.FrontEnd.{i}.	P
Status	R
Name	R
.STBService.{i}.Components.AudioDecoder.{i}.	P
Status	R
Name	R
AudioStandard	R
.STBService.{i}.Components.VideoDecoder.{i}.	P
Status	R
Name	R

Name	Requirement
MPEG2Part2	R
MPEG4Part2	R
MPEG4Part10	R
SMPTEVC1	R
ContentAspectRatio	R
.STBService.{i}.Components.AudioOutput.{i}.	P
Status	R
Name	R
AudioFormat	R
AudioLevel	R
CancelMute	W
.STBService.{i}.Components.VideoOutput.{i}.	P
Status	R
Name	R
CompositeVideoStandard	R
VideoFormat	R
AspectRatioBehaviour	R
.STBService.{i}.AVStreams.	P
ActiveAVStreams	R
AVStreamNumberOfEntries	R
.STBService.{i}.AVStreams.AVStream.{i}.	P
Status	R
Name	R
FrontEnd	R
AudioDecoder	R
VideoDecoder	R
.STBService.{i}.AVPlayers.	P
ActiveAVPlayers	R
AVPlayerNumberOfEntries	R
PreferredAudioLanguage	W
PreferredSubtitlingLanguage	W
.STBService.{i}.AVPlayers.AVPlayer.{i}.	P
Status	R
Name	R
AudioLanguage	R
SubtitlingStatus	R
SubtitlingLanguage	R
AudioOutputs	R
VideoOutputs	R
MainStream	R
PIPStreams	R

7.3 PVR Profile

Table 4 defines the PVR:1 profile for the STBService:1 object. The minimum required version for this profile is STBService:1.0.

Table 4/TR-135 – PVR:1 profile definition for STBService:1

Name	Requirement
.STBService.{i}.Components.PVR.	P
StorageNumberOfEntries	R
.STBService.{i}.Components.PVR.Storage.{i}.	P
Reference	R
.STBService.{i}.AVStreams.AVStream.{i}.	P
PVRState	R

7.4 DTT Profile

Table 5 defines the DTT:1 profile for the STBService:1 object. The minimum required version for this profile is STBService:1.0.

Table 5/TR-135 – DTT:1 profile definition for STBService:1

Name	Requirement
.STBService.{i}.Capabilities.FrontEnd.DVBT.	P
MaxActiveDVBTStreams	R

Name	Requirement
.STBService.{i}.Components.FrontEnd.{i}.DVBT.Modulation.	P
Frequency	W
ChannelBandwidth	W
Constellation	W
HierarchicalInformation	W
CodeRateHP	W
CodeRateLP	W
GuardInterval	W
TransmissionMode	W
SNR	R
BER	R
CBER	R
Locked	R
.STBService.{i}.Components.FrontEnd.{i}.DVBT.Install.	P
Start	W
Status	R
Progress	R
StartFrequency	W
StopFrequency	W
.STBService.{i}.Components.FrontEnd.{i}.DVBT.ServiceListDatabase.	P
Reset	W
TotalServices	R
LogicalChannelNumberOfEntries	R
.STBService.{i}.Components.FrontEnd.{i}.DVBT.ServiceListDatabase.LogicalChannel.{i}.	P
LogicalChannelNumber	R
ServiceNumberOfEntries	R
.STBService.{i}.Components.FrontEnd.{i}.DVBT.ServiceListDatabase.LogicalChannel.{i}.-Service.{i}.	P
Dvbld	R
Frequency	R
BER	R
Preferred	R
.STBService.{i}.Components.FrontEnd.{i}.DVBT.LogicalChannelConnect.	P
LogicalChannelNumber	W
.STBService.{i}.Components.FrontEnd.{i}.DVBT.ServiceConnect.	P
Dvbld	W

7.5 IPTVBaseline Profile

Table 6 defines the IPTVBaseline:1 profile for the STBService:1 object. The minimum required version for this profile is STBService:1.0.

Table 6/TR-135 – IPTVBaseline:1 profile definition for STBService:1

Name	Requirement
.STBService.{i}.Capabilities.FrontEnd.IP.	P
StreamingControlProtocols	R
StreamingTransportProtocols	R

Name	Requirement
StreamingTransportControlProtocols	R
DownloadTransportProtocols	R
MultiplexTypes	R
MaxDejitteringBufferSize	R
.STBService.{i}.Components.FrontEnd.{i}.IP.	P
InboundNumberOfEntries	R
.STBService.{i}.Components.FrontEnd.{i}.IP.IGMP.	P
Status	R
MaximumNumberOfConcurrentGroups	R
ClientVersion	R
ClientUnsolicitedReportInterval	W
.STBService.{i}.Components.FrontEnd.{i}.IP.Dejittering.	P
BufferSize	W
BufferInitialLevel	W
.STBService.{i}.Components.FrontEnd.{i}.IP.Inbound.{i}.	P
StreamingControlProtocol	R
StreamingTransportProtocol	R
StreamingTransportControlProtocol	R
MultiplexType	R
DownloadTransportProtocol	R
SourceAddress	R
SourcePort	R
DestinationAddress	R
DestinationPort	R
URI	R
.STBService.{i}.Components.FrontEnd.{i}.IP.ServiceConnect.	P
URI	W
.STBService.{i}.AVStreams.AVStream.{i}.	P
Inbound	R

7.6 RTCP Profile

Table 7 defines the RTCP:1 profile for the STBService:1 object. The minimum required version for this profile is STBService:1.0.

Table 7/TR-135 – RTCP:1 profile definition for STBService:1

Name	Requirement
.STBService.{i}.Components.FrontEnd.{i}.IP.RTCP.	P
Enable	W
Status	R
TxRepeatInterval	W

7.7 RTPAVPF Profile

Table 8 defines the RTPAVPF:1 profile for the STBService:1 object. The minimum required version for this profile is STBService:1.0.

Table 8/TR-135 – RTPAVPF:1 profile definition for STBService:1

Name	Requirement
.STBService.{i}.Components.FrontEnd.{i}.IP.RTPAVPF.	P
Enable	W
Status	R
RetransmitTimeout	W
MinLossPackets	W
MaxReportTime	W
MaxLossBurst	W

7.8 IPTVHomeNetwork Profile

Table 9 defines the IPTVHomeNetwork:1 profile for the STBService:1 object. The minimum required version for this profile is STBService:1.0.

Table 9/TR-135 – IPTVHomeNetwork:1 profile definition for STBService:1

Name	Requirement
.STBService.{i}.Components.FrontEnd.{i}.IP.	P
OutboundNumberOfEntries	R
.STBService.{i}.Components.FrontEnd.{i}.IP.Outbound.{i}.	P
MultiplexType	R
URI	R
.STBService.{i}.AVStreams.AVStream.{i}.	P
Outbound	R

7.9 IGMP Profile

Table 10 defines the IGMP:1 profile for the STBService:1 object. The minimum required version for this profile is STBService:1.0.

Table 10/TR-135 – IGMP:1 profile definition for STBService:1

Name	Requirement
.STBService.{i}.Components.FrontEnd.{i}.IP.IGMP.	P
Enable	R
Status	R
MaximumNumberOfTrackedGroups	R
LoggingEnable	W
DSCPMark	W
VLANIDMark	W
EthernetPriorityMark	W
ClientVersion	R
ClientRobustness	W
ClientUnsolicitedReportInterval	W
ClientGroupNumberOfEntries	R
ClientGroupStatsNumberOfEntries	R
.STBService.{i}.Components.FrontEnd.{i}.IP.IGMP.ClientGroup.{i}.	P
GroupAddress	R
UpTime	R
.STBService.{i}.Components.FrontEnd.{i}.IP.IGMP.ClientGroupStats.{i}.	P

Name	Requirement
GroupAddress	R
TotalStart	R
CurrentDayStart	R
QuarterHourStart	R
.STBService.{i}.Components.FrontEnd.{i}.IP.IGMP.ClientGroupStats.{i}.Total.	P
NumberOfJoins	R
NumberOfLeaves	R
MaxJoinDelay	R
.STBService.{i}.Components.FrontEnd.{i}.IP.IGMP.ClientGroupStats.{i}.CurrentDay.	P
NumberOfJoins	R
NumberOfLeaves	R
MaxJoinDelay	R
.STBService.{i}.Components.FrontEnd.{i}.IP.IGMP.ClientGroupStats.{i}.QuarterHour.	P
NumberOfJoins	R
NumberOfLeaves	R
MaxJoinDelay	R

7.10 BasicPerfMon Profile

Table 11 defines the BasicPerfMon:1 profile for the STBService:1 object. The minimum required version for this profile is STBService:1.0.

Table 11/TR-135 – BasicPerfMon:1 profile definition for STBService:1

Name	Requirement
.STBService.{i}.ServiceMonitoring.	P
SampleEnable	W
SampleState	R
SampleInterval	W
ReportSamples	W
TimeReference	W
ReportStartTime	R
ReportEndTime	R
MainStreamNumberOfEntries	R
.STBService.{i}.ServiceMonitoring.MainStream.{i}.	P
Enable	W
Status	R
ServiceType	W
AVStream	R
Gmin	W
SevereLossMinDistance	W
.STBService.{i}.ServiceMonitoring.MainStream.{i}.Total.	P
Reset	W
ResetTime	R
.STBService.{i}.ServiceMonitoring.MainStream.{i}.Total.DejitteringStats.	P
Overruns	R
Underruns	R

Name	Requirement
.STBService.{i}.ServiceMonitoring.MainStream.{i}.Total.RTPStats.	P
PacketsReceived	R
PacketsLost	R
LossEvents	R
SevereLossIndexCount	R
.STBService.{i}.ServiceMonitoring.MainStream.{i}.Total.MPEG2TSStats.	P
TSPacketsReceived	R
PacketDiscontinuityCounter	R
.STBService.{i}.ServiceMonitoring.MainStream.{i}.Total.VideoDecoderStats.	P
ILostFrames	R
.STBService.{i}.ServiceMonitoring.MainStream.{i}.Total.VideoResponseStats.	P
MinimumVideoSystemResponse	R
MaximumVideoSystemResponse	R
.STBService.{i}.ServiceMonitoring.MainStream.{i}.Sample.	P
SampleSeconds	R
SignificantChanges	R
PVRTimeShift	R
.STBService.{i}.ServiceMonitoring.MainStream.{i}.Sample.DejitteringStats.	P
SampleSeconds	R
Overruns	R
Underruns	R
.STBService.{i}.ServiceMonitoring.MainStream.{i}.Sample.RTPStats.	P
SampleSeconds	R
PacketsReceived	R
PacketsExpected	R
PacketsLost	R
PacketsLostBeforeEC	R
LossEvents	R
SevereLossIndexCount	R
MaximumLossPeriod	R
MinimumLossDistance	R
.STBService.{i}.ServiceMonitoring.MainStream.{i}.Sample.MPEG2TSStats.	P
SampleSeconds	R
TSPacketsReceived	R
TSPacketsDrained	R
PacketDiscontinuityCounter	R
.STBService.{i}.ServiceMonitoring.MainStream.{i}.Sample.VideoDecoderStats.	P
SampleSeconds	R
ILostFrames	R
.STBService.{i}.ServiceMonitoring.MainStream.{i}.Sample.VideoResponseStats.	P
SampleSeconds	R
AverageVideoSystemResponse	R
MinimumVideoSystemResponse	R
MaximumVideoSystemResponse	R

7.11 ECPerfMon Profile

Table 12 defines the ECPerfMon:1 profile for the STBService:1 object. The minimum required version for this profile is STBService:1.0.

Table 12/TR-135 – ECPerfMon:1 profile definition for STBService:1

Name	Requirement
.STBService.{i}.ServiceMonitoring.MainStream.{i}.Total.RTPStats.	P
PacketsReceivedBeforeEC	R
RetransmitTimeouts	R
PacketsLostBeforeEC	R
LossEventsBeforeEC	R
SevereLossIndexCountBeforeEC	R
.STBService.{i}.ServiceMonitoring.MainStream.{i}.Sample.RTPStats.	P
SampleSeconds	R
PacketsReceivedBeforeEC	R
RetransmitTimeouts	R
PacketsLostBeforeEC	R
LossEventsBeforeEC	R
SevereLossIndexCountBeforeEC	R
MinimumLossDistance	R
MinimumLossDistanceBeforeEC	R
MaximumLossPeriod	R
MaximumLossPeriodBeforeEC	R

7.12 VideoPerfMon Profile

Table 13 defines the VideoPerfMon:1 profile for the STBService:1 object. The minimum required version for this profile is STBService:1.0.

Table 13/TR-135 – VideoPerfMon:1 profile definition for STBService:1

Name	Requirement
.STBService.{i}.ServiceMonitoring.MainStream.{i}.Total.VideoDecoderStats.	P
FrameRate	R
LostFrames	R
DecodedFrames	R
IDecodedFrames	R
ConcealedFrames	R
IConcealedFrames	R
.STBService.{i}.ServiceMonitoring.MainStream.{i}.Sample.	P
HighLevelMetricStatsNumberOfEntries	R
.STBService.{i}.ServiceMonitoring.MainStream.{i}.Sample.DejitteringStats.	P
SampleSeconds	R
EmptyBufferTime	R
.STBService.{i}.ServiceMonitoring.MainStream.{i}.Sample.VideoDecoderStats.	P
SampleSeconds	R
FrameRate	R
LostFrames	R
DecodedFrames	R

Name	Requirement
IDecodedFrames	R
ConcealedFrames	R
IConcealedFrames	R
.STBService.{i}.ServiceMonitoring.MainStream.{i}.Sample.HighLevelMetricStats.{i}	R
SampleSeconds	R
MetricName	R
Metric1Threshold	W
Metric1	R

7.13 AudioPerfMon Profile

Table 14 defines the AudioPerfMon:1 profile for the STBService:1 object. The minimum required version for this profile is STBService:1.0.

Table 14/TR-135 – AudioPerfMon:1 profile definition for STBService:1

Name	Requirement
.STBService.{i}.ServiceMonitoring.MainStream.{i}.Total.AudioDecoderStats	P
DecodedFrames	R
DecodingErrors	R
.STBService.{i}.ServiceMonitoring.MainStream.{i}.Sample.AudioDecoderStats	P
SampleSeconds	R
DecodedFrames	R
DecodingErrors	R

7.14 AudienceStats Profile

Table 15 defines the AudienceStats:1 profile for the STBService:1 object. The minimum required version for this profile is STBService:1.0.

Table 15/TR-135 – AudienceStats:1 profile definition for STBService:1

Name	Requirement
.STBService.{i}.Applications.AudienceStats	P
Enable	W
Reset	W
ResetTime	R
ChannelNumberOfEntries	R
.STBService.{i}.Applications.AudienceStats.Channel.{i}	P
Name	R
Duration	R

7.15 AnalogOutput Profile

Table 16 defines the AnalogOutput:1 profile for the STBService:1 object. The minimum required version for this profile is STBService:1.0.

Table 16/TR-135 – AnalogOutput:1 profile definition for STBService:1

Name	Requirement
.STBService.{i}.Components	P
SCARTNumberOfEntries	R

Name	Requirement
.STBService.{i}.Components.VideoOutput.{i}.	P
VideoFormat	W
Macrovision	R
SCARTs	R
.STBService.{i}.Components.SCART.{i}.	P
Status	R
Name	R
Presence	R

7.16 DigitalOutput Profile

Table 17 defines the DigitalOutput:1 profile for the STBService:1 object. The minimum required version for this profile is STBService:1.0.

Table 17/TR-135 – DigitalOutput:1 profile definition for STBService:1

Name	Requirement
.STBService.{i}.Components.VideoOutput.{i}.	P
HDCP	R

7.17 CA Profile

Table 18 defines the CA:1 profile for the STBService:1 object. The minimum required version for this profile is STBService:1.0.

Table 18/TR-135 – CA:1 profile definition for STBService:1

Name	Requirement
.STBService.{i}.Components.	P
CANumberOfEntries	R
.STBService.{i}.Components.CA.{i}.	P
Status	R
Name	R
SmartCardReader	R
.STBService.{i}.AVStreams.AVStream.{i}.	P
CA	R

7.18 DRM Profile

Table 19 defines the DRM:1 profile for the STBService:1 object. The minimum required version for this profile is STBService:1.0.

Table 19/TR-135 – DRM:1 profile definition for STBService:1

Name	Requirement
.STBService.{i}.Components.	P
DRMNumberOfEntries	R
.STBService.{i}.Components.DRM.{i}.	P
Status	R
Name	R

Name	Requirement
SmartCardReader	R
.STBService.{i}.AVStreams.AVStream.{i}	P
DRM	R

Appendix I – Theory of Operations

This non-normative appendix discusses profile usage, instance number usage, service monitoring, and then re-visits the use cases of section 4, giving detailed examples of how they can be addressed by the STB data model. Finally, it presents various examples of AV Stream and AV Player usage.

I.1 Profile usage

The following profiles are defined in section 7.

- **Baseline:** basic objects and parameters that every STB will probably want to support.
- **PVR:** adds PVR support.
- **DTT:** adds DTT support.
- **IPTVBaseline:** adds IPTV baseline support, including inbound IP streams and basic IGMP support.
- **RTCP:** adds RTCP support.
- **RTPAVPF:** adds RTP/AVPF support.
- **IPTVHomeNetwork:** adds outbound IP stream support.
- **IGMP:** adds more IGMP controls and statistics.
- **BasicPerfMon:** basic IPTV performance monitoring, including RTP, MPEG2-TS, video decoder and video response statistics.
- **ECPerfMon:** adds RTP EC (Error Correction) statistics.
- **VideoPerfMon:** adds de-jittering statistics, more detailed video decoder statistics, and high-level metrics.
- **AudioPerfMon:** adds audio decoder statistics.
- **AudienceStats:** adds audience statistics collection support.
- **AnalogOutput:** adds Macrovision and SCART support.
- **DigitalOutput:** adds HDCP support.
- **CA:** adds CA support.
- **DRM:** adds DRM support.

The following types of STB might support the indicated profiles. In addition, the CA and/or DRM profiles could be supported as needed.

- **IP STB, no PVR, no performance monitoring, analog outputs:** Baseline, IPTV-Baseline, AnalogOutput.
- **IP STB, PVR, no performance monitoring, analog and digital outputs:** Baseline, PVR, IPTVBaseline, AnalogOutput, DigitalOutput.

- **IP STB, PVR, basic performance monitoring, analog and digital outputs:** Baseline, PVR, IPTVBaseline, BasicPerfMon, AnalogOutput, DigitalOutput.
- **DTT STB, PVR, no performance monitoring, digital outputs:** Baseline, PVR, DTT, DigitalOutput.
- **Hybrid DTT / IP STB, PVR, IGMP statistics, basic and video performance monitoring, analog and digital outputs:** Baseline, PVR, DTT, IPTVBaseline, IGMP, BasicPerfMon, VideoPerfMon, AnalogOutput, DigitalOutput.

I.2 Instance number usage

Most STB data model objects are created when the STB boots, and remain in existence for its lifetime. Some objects are created or deleted as the STB configuration changes. No objects are created or deleted by the ACS, which reflects the fact that the STB data model provides primarily a read-only view.

It is important to note that instance numbers are always chosen by the STB, and that the ACS can make no assumptions about them. The ACS must always use object parameters, e.g. Name, in order to determine how a given object instance relates to the STB configuration.

The following sections review the various different types of object.

I.2.1 Fixed objects with fixed purpose

Objects that relate to physical components will exist whenever the corresponding physical component exists. If a component can be added or removed, e.g. a USB-attached tuner card, it is up to the STB implementation to decide whether to create or delete the corresponding object (it might choose just to disable the object when the physical component is removed).

For example, if an STB has two DTT tuners, it might choose to use their OS device names for their Name parameters, e.g. “dvb0” and “dvb1”. The data model might then include the following:

```
.Components.FrontEnd.1.
  Name = "dvb0"
.Components.FrontEnd.2.
  Name = "dvb1"
```

There is no guarantee that the instance numbers would be 1 and 2. This is up to the STB implementation.

I.2.2 Fixed objects with variable purpose

Some objects are created statically and are assigned as needed. The main examples of this in the STB data model are the ServiceMonitoring.MainStream.{i}, AVPlayers.AVPlayer.{i}, AVStreams.AVStream.{i}, Components.FrontEnd.{i}.IP.Inbound.{i} and Components.FrontEnd.-{i}.IP.Outbound.{i} collections. The first two collections are actually rather different from the others, for the following reasons:

Service Monitoring Main Stream objects contain writable parameters: they can be enabled and disabled, and the associated service type can be set, as can several other parameters that affect statistics collection for that instance.

AV Player objects contain writable parameters: they can be enabled and disabled. Therefore, to be useful to the ACS, they should correspond to fixed combinations of Audio and Video Outputs.

AV Stream objects are entirely read-only, and simply reflect what the STB is currently doing. A given AV Stream instance is not likely to be tied to a fixed combination of components, and certainly not to a fixed TV channel (there are just too many combinations).

IP Inbound and Outbound objects are also entirely read-only, and simply reflect what the IP Front End is currently doing.

A given STB may choose to create more Service Monitoring Main Stream, AV Player, AV Stream, IP.Inbound or IP.Outbound objects than can be simultaneously active. For this reason, there are MaxActiveMainStreams, MaxActiveAVPlayers, MaxActiveAVStreams, MaxActive-InboundIPStreams and MaxActiveOutboundIPStreams capability parameters. Again, the reasons are different for the different objects:

For Service Monitoring Main Streams, the STB might pre-configure an instance for each supported service type, but perhaps only one or two MainStream instances can be active at any given time.

For AV Players, there could be a large number of output device combinations, but perhaps only one or two Players can be active at any given time.

For AV Streams, the STB could choose to use different instance number ranges for, for example, DTT- and IP-delivered Streams (note that no information is maintained about streams that are no longer active in the data model).

See section I.8 for AV Stream and AV Player examples.

I.2.3 Objects created as needed

The following objects are created (and deleted) by the STB as needed:

- **DVB service list database:** logical channel and service objects are created as the STB scans for services. Note that this is done only on demand, so these objects are not being created and deleted during normal operation.
- **IGMP group statistics:** group objects are created as the STB joins and leaves Multicast groups. This does happen during normal operation, but the maximum number of such objects is limited by a capability parameter.
- **Audience statistics:** per-channel objects (indexed by channel name) are created as the user watches new channels. This does happen during normal operation, but the maximum number of such objects is limited by a capability parameter.

I.3 Service Monitoring

The STB data model contains a ServiceMonitoring object that is primarily aimed at the Performance Management use cases of section 4.3 but that is also applicable to other use cases. Because it is a complex and potentially confusing object, it is described in its own section.

I.3.1 Key properties

The ServiceMonitoring object has the following key properties:

Statistics are collected only for AV streams that the STB designates as “main streams”; this is a simplification that gives the STB the flexibility to collect statistics only for those streams that it (or the service provider) regards as important.

The STB can collect statistics autonomously over an extended period, allowing the ACS to maintain a complete performance record across a population of STBs without the need for frequent STB / ACS communication.

Statistics collection is performed independently for various service types, e.g. IPTV, VoD and DTT, meaning that statistics from one service type will not pollute statistics from another service type.

The STB can collect both Total statistics, e.g. total number of lost packets, and Sample statistics, e.g. numbers of lost packets during the last n sample intervals.

Total statistics are collected since the STB booted, or they were last reset, whichever happened most recently.

Sample statistics are collected during fixed-length sample intervals, and the STB can store the most recent n values for each statistic. Sample intervals can be aligned with absolute time. The ACS can either periodically read stored Sample statistics or else can be notified when it is time to read them. Some Sample statistics can generate a notification when their values go outside a designated range.

The STB can collect various categories of statistics: de-jittering, RTP, MPEG2-TS, video decoder, audio decoder, video response and high-level metrics; the ACS can tell when a given category is not relevant to a particular stream. Some of these statistics categories are described in more detail starting in section I.3.5.

I.3.2 Operational overview

The top-level of the ServiceMonitoring object contains only parameters for configuring and monitoring the collection of Sample statistics. This reflects the fact that all Sample statistics, regardless of service type, are collected over the same sample intervals.

At the next level is a multi-instance MainStream object. Each instance is associated with a service type, can be enabled and disabled, and the associated service type can be set, as can several other parameters that affect statistics collection.

Below the MainStream instances, there is a Total object for Total statistics and a Sample object for Sample statistics. These objects are mostly read-only and contain just a few writable parameters, e.g. a Reset parameter to reset Total statistics, and a few statistics category-specific parameters.

Total statistics are straightforward and require no further explanation.

I.3.3 Service types and MainStream instances

I.3.3.1 Service types

A MainStream instance's service type constrains the AV streams for which statistics will be collected for that instance. The standard service types are as follows (an STB can also support vendor-specific service types):

- **IPTV**: WAN-sourced IPTV stream
- **VoD**: WAN-sourced VoD stream
- **IP**: WAN-sourced IP stream (includes IPTV and VoD)
- **CAB**: Cable, e.g. DVB-C, Front End stream
- **DTT**: DTT, e.g. DVB-T, Front End stream
- **SAT**: Satellite, e.g. DVB-C, Front End stream
- **PVR**: PVR play-out stream

Please note the following:

Only WAN-sourced IP streams are supported. This reflects the fact that this version of the STB data model pays little attention to LAN → LAN streams.

All the above service types are characterized by the *source* of the AV stream, e.g. "DTT" covers any stream that is received by a DTT Front End, regardless of whether it is being recorded, viewed live, or viewed with time-delay.

Service types do not explicitly model PVR recording or PVR time-shift. This is because PVR recording is indicated by the absence of decoder statistics, and PVR time-shift is indicated by a PVRTimeShift comma-separated list parameter.

I.3.3.2 MainStream instances

MainStream instances are created statically. An STB might choose to create an instance for each of the main service types, perhaps with an extra one for other service types. For example:

```
.ServiceMonitoring.  
    MainStreamNumberOfEntries = 3  
  
.ServiceMonitoring.MainStream.1.  
    Enable = True  
    ServiceType = "IPTV"  
    Sample.RTPStats.SevereLossMinDistance = 2  
.ServiceMonitoring.MainStream.2.  
    Enable = True  
    ServiceType = "VoD"  
    Sample.RTPStats.SevereLossMinDistance = 5  
.ServiceMonitoring.MainStream.3.
```

```
Enable = False
ServiceType = ""
```

Such an STB might set `.Capabilities.ServiceMonitoring.MaxActiveMainStreams` to 2, indicating that only two of the above three instances can be active at the same time. In this case, if the ACS wanted to enable DTT statistics collection, it would have to disable either IPTV or VoD statistics collection.

When a `MainStream` instance is enabled, it will probably be in the middle of a sample interval. Furthermore, the statistics parameters for the other enabled `MainStream` instances might already contain values, and the newly-enabled `MainStream` instance needs to remain synchronised with the existing instances. This creates some complications, which are considered in section I.3.5.8.

I.3.4 Sample statistics overview

I.3.4.1 Configuration parameters

The collection of Sample statistics is configured via the following parameters. All of these are shared by all service types and are therefore at the top level of the `.ServiceMonitoring` object:

- **SampleEnable:** Enables and disables all collection of Sample statistics.
- **SampleInterval:** The sample interval over which each statistic is measured.
- **ReportSamples:** The number of samples of each statistic that can be stored on the STB.
- **FetchSamples:** The number of samples between `SampleState` transitions (used only for notifications).
- **TimeReference:** Reference time to which sample intervals and `SampleState` transitions can be aligned.
- **ForceSample:** Explicit request to update the most recent sample (normally samples are updated only at the end of the sample interval).

I.3.4.2 Statistics representation

Each Sample statistic is represented by a comma-separated list of values, of which the first one is the oldest and the last one is the most recent. All of these are specific to the service type and are therefore in the `.ServiceMonitoring.MainStream.{i}.Sample` object.

The following Sample statistics are independent of the statistics category and are therefore at the top level of the `.ServiceMonitoring.MainStream.{i}.Sample` object:

- **SampleSeconds:** Each value is the number of seconds during which data was collected during the sample interval.
- **SignificantChanges:** Each value is the number of significant changes, e.g. channel changes, that might be expected to affect the statistics during the sample interval.
- **PVRTimeShift:** Each value is a Boolean that indicates whether a non-zero PVR time-shift was used during the sample interval (in which case network and decoder statistics might not be correlated).

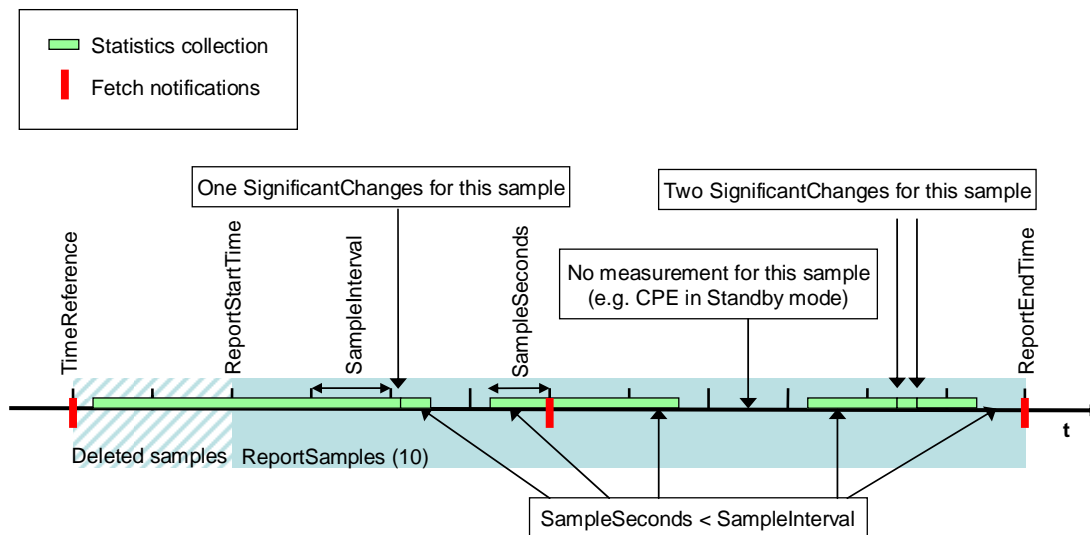
The remaining Sample statistics are specific to the statistics category and are in category sub-objects. For example, the following de-jittering parameters are in the `.ServiceMonitoring.-MainStream.{i}.Sample.DejitteringStats` object:

- **SampleSeconds:** This is similar to the top-level `SampleSeconds` parameter, but this one relates directly to the availability of de-jittering statistics; the value of this parameter, for a given sample interval, will always be less than or equal to the value of the top-level `SampleSeconds` parameter
- **Overruns, Underruns, EmptyBufferTime:** These are the actual de-jittering statistics.

If the STB is in Standby mode, there will be no active AV streams, but sample intervals will continue to be timed as normal. The fact that the STB was in Standby mode will be indicated by the fact that all the `SampleSeconds` values will be zero when the STB was in Standby mode for the entire sample interval, and will be less than `SampleInterval` when the STB was in Standby mode for only part of the sample interval.

I.3.4.3 Example configuration

Figure 6 illustrates all the above parameters.



Example parameter values:

(Input parameters)

SampleEnable = True

SampleInterval = 3600

ReportSamples = 10

FetchSamples = 6

TimeReference = "2007-06-29T00:00:00Z"

(Output parameters)

SampleSeconds = "3600,3600,1800,2800,3600,2500,0,2870,3600,1200"

SignificantChanges = "0,0,1,0,0,0,0,2,0"

PVRTimeShift = "0,0,0,0,1,0,0,0,0,0"

DejitteringStats.SampleSeconds =
"3600,3600,1800,2800,3600,2500,0,2870,3600,1200"

DejitteringStats.OVERRUNS = "0,2,0,0,5,0,0,0,0,0"

DejitteringStats.Underruns = "0,1,0,0,0,0,0,0,0,0"

DejitteringStats.EmptyBufferTime = "0,5,0,0,0,0,0,0,0,0"

Figure 6/TR-135 – Sample statistics parameters

There is a lot of detail in the figure. Here is some explanation:

- The most important part of the configuration is straightforward: (SampleEnable, SampleInterval, ReportSamples) = (True, 3600, 10).
- This indicates that Sample statistics collection is enabled, the sample interval is 3600 (an hour) and the STB can store 10 values for each statistic.
- The region with the solid background represents the 10 samples for which values are currently stored on the STB. ReportStartTime and ReportEndTime indicate the corresponding range of absolute times.
- The “Statistics collection” periods are also indicated, with the 5 samples during which SampleSeconds was less than SampleInterval, and the 1 sample during which the STB was in Standby mode and there were no measurements. The SampleSeconds comma-separated list contains this information.
- Similarly, the significant changes, e.g. channel changes, are shown, as are the sample intervals during which PVR time-shift was used. The SignificantChanges and PVRTimeShift comma-separated lists contain this information.

- TimeReference is set to UTC midnight, which means that, because SampleInterval is 3600, samples will always end on UTC hour boundaries.
- FetchSamples is set to 6. Together with TimeReference, this means that SampleState state transitions (“Fetch notifications”) will occur every 6 hours, at UTC 00:00, 06:00, 12:00 and 18:00.
- If the ACS enables Active notifications on SampleState and if it reads the statistics every time it gets a notification, it will read 10 hours worth of statistics every 6 hours, i.e. there is an overlap of 4 hours worth of statistics.

I.3.5 Sample statistics details

In this section, detailed Sample statistics behaviour is explored via a series of examples. These examples merely illustrate the behavior that is already defined in the data model; they do not add any new requirements.

Suppose that the following settings are present in the STB’s configuration:

```
.ServiceMonitoring.
    SampleEnable = True
    SampleInterval = 3600
    ReportSamples = 10
    FetchSamples = 0
    TimeReference = "0001-01-01T00:00:00Z"
```

I.3.5.1 How Sample statistics collection is initiated on boot

When the STB boots, it will, once it is ready, begin to collect Sample statistics. TimeReference has been set to the Unknown Time, so there is no requirement to align sample intervals with absolute time, and the first sample interval will begin immediately.

For example, if statistics collection begins at 2007-06-29T01:02:03Z, then the sample intervals will be as follows:

```
2007-06-29T01:02:03Z → 2007-06-29T02:02:03Z
2007-06-29T02:02:03Z → 2007-06-29T03:02:03Z
2007-06-29T03:02:03Z → 2007-06-29T04:02:03Z
...
```

I.3.5.2 ACS options for reading Sample statistics

There are various ways in which the ACS can read Sample statistics. For example:

- It can poll them at some fixed interval, perhaps as part of a daily periodic Inform, in which case the TimeReference parameter will have to be configured appropriately; see section I.3.5.5.
- It can enable Active Notification on the SampleState parameter, in which case the FetchSamples parameter will have to be configured appropriately; see section I.3.5.6.
- It can read them as needed, driven by some other logic.

I.3.5.3 Why reading statistics doesn't update them in the data model

Suppose that the ACS reads some statistics at 2007-06-29T01:32:03Z, i.e. half way through the second sample interval. The returned values might be as follows:

```
.ServiceMonitoring.
  ReportStartTime = "2007-06-29T01:02:03Z"
  ReportEndTime = "2007-06-29T02:02:03Z"
.ServiceMonitoring.MainStream.1.Sample.
  SampleSeconds = "3600"
  SignificantChanges = "0"
  PVRTimeShift = "0"
.ServiceMonitoring.MainStream.1.Sample.DejitteringStats.
  Overruns = "0"
  Underruns = "5"
  EmptyBufferTime = "0"
```

There is no data from the second sample because Sample statistics are updated in the data model only at the end of each sample interval (unless ForceSample is used). To see why this is the correct behaviour, consider the parameter values at the end of the 10th sample interval:

```
.ServiceMonitoring.
  ReportStartTime = "2007-06-29T01:02:03Z"
  ReportEndTime = "2007-06-29T11:02:03Z"
.ServiceMonitoring.MainStream.1.Sample.
  SampleSeconds =
"3600,3600,1800,2800,3600,2500,0,2870,3600,1200"
  SignificantChanges = "0,0,1,0,0,0,0,0,0,2,0"
  PVRTimeShift = "0,0,0,0,1,0,0,0,0,0"
.ServiceMonitoring.MainStream.1.Sample.DejitteringStats.
  SampleSeconds =
"3600,3600,1800,2800,3600,2500,0,2870,3600,1200"
  Overruns = "0,2,0,0,5,0,0,0,0,0"
  Underruns = "0,1,0,0,0,0,0,0,0,0"
  EmptyBufferTime = "0,5,0,0,0,0,0,0,0,0"
```

Suppose that the ACS reads a statistics parameter at 2007-06-29T11:02:04Z, i.e. one second after the end of the 10th sample interval. If this caused the Sample statistics to be updated in the data model, then *every* Sample statistic for every MainStream instance would have to be updated, *every* oldest sample would have to be discarded, and ReportStartTime and ReportEndTime would have to be updated. The result would be as follows:

```
.ServiceMonitoring.
  ReportStartTime = "2007-06-29T02:02:03Z"
  ReportEndTime = "2007-06-29T11:02:04Z"
.ServiceMonitoring.MainStream.1.Sample.
  SampleSeconds =
"3600,1800,2800,3600,2500,0,2870,3600,1200,1"
  SignificantChanges = "0,1,0,0,0,0,0,0,2,0,0"
  PVRTimeShift = "0,0,0,1,0,0,0,0,0,0"
```

```
.ServiceMonitoring.MainStream.1.Sample.DejitteringStats.
  SampleSeconds =
"3600,1800,2800,3600,2500,0,2870,3600,1200,1"
  Overruns = "2,0,0,5,0,0,0,0,0,0"
  Underruns = "1,0,0,0,0,0,0,0,0,0"
  EmptyBufferTime = "5,0,0,0,0,0,0,0,0,0"
```

I.3.5.4 How to force update of Sample statistics in the data model

The ForceSample parameter can be used to force Sample statistics to be updated in the data model. If ForceSample was set to True at 2007-06-29T11:02:04Z, i.e. 1 second after the start of the 11th sample interval, the Sample parameters would be updated as shown above. If ForceSample was again set to True at 2007-06-29T11:02:14Z, i.e. 11 seconds after the start of the 11th sample interval, then only ReportEndTime and the final values in the comma-separated lists would be updated:

```
.ServiceMonitoring.
  ReportStartTime = "2007-06-29T02:02:03Z"
  ReportEndTime = "2007-06-29T11:02:14Z"
.ServiceMonitoring.MainStream.1.Sample.
  SampleSeconds =
"3600,1800,2800,3600,2500,0,2870,3600,1200,11"
  SignificantChanges = "0,1,0,0,0,0,0,2,0,0"
  PVRTimeShift = "0,0,0,1,0,0,0,0,0,0"
.ServiceMonitoring.MainStream.1.Sample.DejitteringStats.
  SampleSeconds =
"3600,1800,2800,3600,2500,0,2870,3600,1200,11"
  Overruns = "2,0,0,5,0,0,0,0,0,0"
  Underruns = "1,0,0,0,0,0,0,0,0,1"
  EmptyBufferTime = "5,0,0,0,0,0,0,0,0,0"
```

I.3.5.5 How to align sample intervals with absolute time

Sample intervals can be aligned with absolute time by setting TimeReference to a time other than the Unknown Time.

```
.ServiceMonitoring.
  SampleEnable = True
  SampleInterval = 3600
  ReportSamples = 10
  FetchSamples = 0
  TimeReference = "2007-06-29T00:00:00Z"
```

If statistics collection begins, as before, at 2007-06-29T01:02:03Z, then the first sample interval begins immediately but, because the configuration requires the sample intervals to complete at 00:00, 01:00, 02:00 etc, sample intervals will be as follows:

```
2007-06-29T01:02:03Z → 2007-06-29T02:00:00Z
2007-06-29T02:00:00Z → 2007-06-29T03:00:00Z
2007-06-29T03:00:00Z → 2007-06-29T04:00:00Z
```

...

In other words, the first sample interval will be shorter, by 123 seconds, than the second and subsequent sample intervals. The ACS can determine this from the ReportStartTime and ReportEndTime parameters.

I.3.5.6 How to configure SampleState Active Notifications

SampleState notifications can be configured by setting FetchSamples to a value other than 0.

```
.ServiceMonitoring.
  SampleEnable = True
  SampleInterval = 3600
  ReportSamples = 10
  FetchSamples = 6
  TimeReference = "2007-06-29T00:00:00Z"
```

FetchSamples does not affect the operation of Sample statistics in any way. Its only purpose is to control when the SampleState Enabled → Trigger → Enabled transitions will take place. If the ACS enables Active Notification on SampleState, this transition can trigger the ACS to read Sample statistics. FetchSamples specifies the number of samples between such triggers. In the above example, FetchSamples is 6, so triggers will occur on completion of every 6th sample. These triggers are aligned with TimeReference, so they will occur at 00:00, 06:00, 12:00 and 18:00 every day.

I.3.5.7 Enabling and disabling Sample statistics

Up until now, the examples have assumed that Sample statistics are being collected all the time. However, there are various writable parameters that, when changed, cause Sample statistics to be reset.

- **SampleEnable:**
 - If changed from True → False, Sample statistics collection ceases, and the values of all comma-separated list Sample statistics parameters become undefined.
 - If changed from False → True, the values of all comma-separated list Sample statistics parameters are set to empty strings, and Sample statistics collection begins exactly as it does when the STB boots.
- **SampleInterval:** If changed when SampleEnable is True, Sample statistics collection is reset exactly as if SampleEnable was changed from True → False → True.
- **ReportSamples:** Does not cause reset of Sample statistics (just causes the STB's statistics buffers to be truncated or extended as appropriate).
- **FetchSamples:** Does not cause reset of Sample statistics (comes into immediate effect if changed while SampleEnable is True).
- **TimeReference:** Exactly as for SampleInterval.

- **ForceSample:** Does not cause reset of Sample statistics (has no effect if changed while SampleEnable is False).

I.3.5.8 Enabling and disabling MainStream instances

When a MainStream instance is disabled, Sample statistics collection on that instance (if enabled) ceases, and the values of all comma-separated list Sample statistics parameters become undefined.

When a MainStream instance is enabled, and if Sample statistics collection is currently enabled, the situation is complicated by the following:

- The current time is probably in the middle of a sample interval.
- The statistics parameters for the other enabled MainStream instances probably already contain values, and the newly-enabled MainStream instance needs to remain synchronised with the existing instances.

For example, suppose that Sample statistics collection is currently enabled, MainStream instance #1 is enabled, and it's currently 1800 seconds into the 5th 3600 second sample interval. MainStream #2 is now enabled. The situation at the end of the 5th sample interval might be as follows:

```
.SampleMonitoring.MainStream.1.
    SampleSeconds = "3600,3600,3600,3600,3600"
...
.SampleMonitoring.MainStream.2.
    SampleSeconds = "0,0,0,0,1800"
...
```

This illustrates how 4½ sample intervals have to be “invented” for MainStream #2, so that it can remain synchronised with MainStream #1.

I.3.5.9 How to use Active Notifications for individual statistics

A small number of individual statistics support the generation of Active Notifications if their values during a given sample interval fall outside a designated range.

For example, suppose that an STB supports a MOS-V user-perceived video quality high-level metric. The relevant part of the configuration might be as follows:

```
.ServiceMonitoring.MainStream.1.Sample.
    HighLevelMetricStatsNumberOfEntries = 1
.ServiceMonitoring.MainStream.1.Sample.
    HighLevelMetricStats.1.
    MetricName = "MOS-V"
    Metric1Threshold = 50000
```

Metric1Threshold indicates that, each time that the value of the metric is measured as less than 50000, Metric1Failures should be incremented. The ACS can enable Active Notification on Metric1Failures and so can be notified of such occurrences.

Note that the STB data model does not specify details of any particular high-level metrics. In the above example, it would be up to the STB implementation to ensure that notifications were generated only at appropriate times and at appropriate rates.

I.3.5.10 How to handle service configuration changes

If there is a change to a service configuration, e.g. the IPTV error correction mechanism is changed, statistics collected before the change might not be directly comparable with those collected after the change.

It is assumed that the ACS is aware of the times of such changes and so is able to identify any sample intervals that might be affected.

I.3.6 RTP statistics

RTP statistics are based on the concept of *Loss Event*, which is defined as a sequence of lost packets, possibly including *islands* of received packets. Each island consists of up to $G_{min} - 1$ received packets (a sequence of G_{min} received packets terminates the Loss Event, and so is not an island). G_{min} is writable, so that the Loss Event can be configured as best suits the operator.

For example, if G_{min} is 1 then there can be no islands and any sequence of lost packets is a loss event. If G_{min} is 2 then the loss event can (but need not) contain islands of 1 received packet and is terminated by 2 consecutive received packets. The following are all examples of loss events in the $G_{min} = 2$ case (where the loss event begins with the first shown lost packet and ends with the last shown lost packet):

- R, L, R, R (length = 1)
- R, L, L, R, R (length = 2)
- R, L, R, L, R, R (length = 3)

It is important to note that this definition considers single errors as error bursts of length 1. Loss Events are characterized by their length and by the value of G_{min} that defines them.

Service Monitoring statistics capture the length (in RTP packets) of Loss Events and their distribution over time. For both parameters the data model provides general statistics (minimum, average and maximum) and a comparison against a configurable threshold, in the form of the count of packets that exceeded this threshold. There are writable parameters in the data model for both thresholds, describing the maximum “non severe” length for a Loss Event and the minimum “non severe” distance between adjacent Loss Events. Loss Events longer than the length threshold or closer than the distance threshold are classified as “severe”. This way the data model provides a flexible insight into how well the STB is performing.

The ACS can change these parameters at any time. Changes take place immediately and are not recorded in any statistics, so the sample interval during which these parameters change will contain polluted statistics. Statistics will be reliable again from the next sample interval.

The STB data model also provides a number of statistics that are measured both before and after Error Correction (EC). The ACS can understand whether any EC is being applied by comparing the performance before EC, captured by parameters with suffix “BeforeEC”, and after EC,

captured by parameters with no specific suffix. If no EC is being applied, both statistics will have the same values.

Finally, the STB data model also provides statistics about *Discarded* packets, including *Late* and *Out of Sequence* packets. *Late* is defined as “too late for playout”. *Out of sequence* is defined with reference to the RTP sequence count. Late packets may or may not be out of sequence, and out of sequence packets may or may not be too late for playout.

I.3.7 MPEG2-TS statistics

MPEG2-TS is used as a multiplexing format in a number of services: it is mandatory in DTT, where MPEG2-TS is transported directly over the physical layer, whereas it is optional, though quite common at the moment, in IPTV and VoD, where it can be transported over RTP/UDP or simply over UDP. For these services RTP could be used instead of MPEG2-TS. To account for common practice, two counters are provided, TSPacketsReceived and TSPacketsDrained, which account for the state of the de-jittering buffer from a MPEG2-TS perspective. Another parameter of general interest is TSDiscontinuityCounter, which provides some information about whether a discontinuity, e.g. packet loss, took place. The discontinuity counter can be evaluated before or after the CA, if present, in order to account for packet loss that might be introduced by the CA block. Parameters significant only for DTT are TSSyncLossCount, which indicates loss of synchronization, and TSSyncByteErrorCount, which counts how many errored TS Synchronization Bytes were received [22].

I.4 Configuration

The STB data model permits only a small number of parameters to be written, because, as was explained in section 2, it is assumed that the Media Delivery Platform takes responsibility for configuring the STB for media processing.

The following STB data model parameters are writable:

- **Enable/disable:** overall STB object, components, applications (also some others listed below)
- **DVB-T Front End modulation:** type, frequency, channel bandwidth, constellation etc
- **DVB-T Front End service list:** scan control, frequency range, reset, explicit tune to logical channel or service
- **RTCP:** enable/disable, transmission repeat interval
- **RTP/AVPF:** re-transmit timeout, re-transmit criteria
- **IGMP client:** enable/disable, logging enable/disable, QoS markings, robustness, unsolicited report interval
- **IP de-jittering:** buffer size, buffer play-out initial level
- **IP service connect:** explicit connect to specified service
- **Audio Output:** audio level, cancel mute
- **Video Output:** video format

- **SCART:** “presence” control signal
- **AV Players:** preferred audio and subtitling languages; enable/disable of individual players
- **Service Monitoring:** see section I.3 for a general description; in addition there are some statistics category-specific settings (e.g. severe loss minimum distance) and thresholds for high-level metric event generation
- **Audience viewing statistics:** enable/disable and reset

Most of these parameters are writable in support of specific Trouble Management (section I.4.2), Performance Management (section I.6) and Fault Management (section I.7) use cases.

However, the following parameters are likely to be configured when a STB is first deployed, and are described in more detail in the following sections.

- **DVB-T Front End service list:** initial scan of service list
- **Service Monitoring:** all of the writable parameters; see section I.3 for general examples
- **AV Players:** preferred audio and subtitling languages

I.4.1 DVB-T Front End service list

It is common for the user to have to manually initiate a channel scan on a newly installed STB. For DVB-T Front Ends, the STB data model allows the ACS to do this on behalf of the user.

The procedure, which proceeds independently for each DVB-T Front End, is as follows (the examples assume Front End #1):

```
.Components.FrontEnd.1.DVBT.ServiceListDatabase
    Reset = True
.Components.FrontEnd.1.DVBT.Install.
    StartFrequency = <start-frequency>
    StopFrequency = <stop-frequency>
    Start = True
```

The initial Reset will clear out any existing entries in service list database. If the STB is newly installed, the service list database is presumably empty but the ACS might choose to issue the Reset just in case.

When the scan starts, two variables will change to indicate that it is active, and its progress:

```
.Components.FrontEnd.1.DVBT.Install.
    Status = "Enabled"
    Progress = <percentage-complete>
```

As channels are found, entries are added to the service list database, which is indexed by logical channel number (corresponds to the channel, e.g. “BBC1”). A given logical channel may be available from more than one transmitter, and each (logical channel, transmitter) combination is referred to as a “service” (more rigorously, a service is identified by a DvbId, which is a triplet of (Original Network Id, Service Id, Transport Stream Id)).

For example, in an area of very poor reception, perhaps only two logical channels were found, “BBC1” from a single transmitter and “BBC2” from two transmitters:

```
.ServiceListDatabase.
  TotalServices = 3
  LogicalChannelNumberOfEntries = 2

.ServiceListDatabase.LogicalChannel.1.
  LogicalChannelNumber = 1
  ServiceNumberOfEntries = 1
.ServiceListDatabase.LogicalChannel.1.Service.1.
  DvbId = "20FA00040101"
  Frequency = 100000
  BER = 10
  Preferred = True

.ServiceListDatabase.LogicalChannel.2.
  LogicalChannelNumber = 2
  ServiceNumberOfEntries = 2
.ServiceListDatabase.LogicalChannel.2.Service.1.
  DvbId = "20FA00040102"
  Frequency = 101000
  BER = 10
  Preferred = True
.ServiceListDatabase.LogicalChannel.2.Service.2.
  DvbId = "20FA00040103"
  Frequency = 102000
  BER = 20
  Preferred = False
```

The Preferred parameter indicates the preferred service. A scan will always set the preferred service for a given logical channel number to be that with the lowest BER. However, the ACS can override this.

I.4.2 AV Players

The ACS can set the preferred audio and subtitling languages:

```
.AVPlayers.
  PreferredAudioLanguage = "en-UK"
  PreferredSubtitlingLanguage = "en-UK"
```

I.5 Trouble Management

Trouble management is the action taken by a trained technician when the user reports a problem.

As already noted, only a relatively small number of STB data model parameters can be modified. Many of these correspond directly to Trouble Management use cases:

- **Enable/disable:** A component can be temporarily disabled in order to test whether it was responsible for a problem. For example, if one nominally operational DVB-T Front End indicates a very low SNR, but the other one works fine, then perhaps the aerial connector has fallen off the first one?

```
.Components.FrontEnd.1.Enable = False
```

- **DVB-T Front End modulation:** DVB-T Front Ends offer a lot of configurable parameters, and provide feedback on BER, SNR etc. The technician can ask the user to watch live TV and can then check and adjust parameters while monitoring the results.

```
.Components.FrontEnd.2.DVBT.Modulation.  
  ChannelBandwidth = "8MHz"  
  GuardInterval = "1/32"
```

- **DVB-T Front End service list:** The technician can force a DVB-T Front End to tune to a specified logical channel or service. If this fails to result in a reasonable SNR, the technician can initiate a full or partial channel re-scan, and can monitor its progress. See section I.5.1.

```
.Components.FrontEnd.2.DVBT.LogicalChannelConnect.  
  LogicalChannelNumber = 1
```

```
.Components.FrontEnd.2.DVBT.ServiceConnect.  
  DvbId = "20FA00040101"
```

- **IP service connect:** as with DVB-T, the technician can force an IP Front End to connect to a specified service. See section I.5.1.

```
.Components.FrontEnd.1.IP.ServiceConnect.  
  URI = "udp://224.112.15.18:3257?StrCtrl=IGMP  
        ?StrTrspCtrl=AL-FEC?MuxType=MPEG2-TS"
```

- **Audio Output:** Perhaps the user inadvertently turned the volume down or forgot to cancel mute? The technician can determine this, and can fix it.

```
.Components.AudioOutput.1.CancelMute = True
```

- **Video Output:** Perhaps the video format is wrong for the output device? The technician can try different settings.

```
.Components.VideoOutput.1.VideoFormat = "YPrPb"
```

- **SCART:** Related to the above, perhaps the SCART “presence” control signal is not asserted? The technician can try different settings by forcing the “presence” to be asserted.

```
.Components.SCART.1.Presence = True
```

Related to the above, the technician could also try to force S-Video or sync-on-green RGB on the SCART output. To do this he might set the value of the VideoFormat parameter in the AVOutput.*{i}*. object that references the SCART to “S-Video”.

```
.Components.AVOutput.1.VideoFormat = "S-Video"
```

- **Service Monitoring:** If the system is working but there is a QoE issue, the technician can monitor the various classes of statistics and can assess likely problems. For example, if the user reports that the video is blocky, the technician can look at Overruns and Underruns in order to determine whether there might be a packet loss or jitter problem. The technician can try changing the de-jittering buffer size. If this works, it was a jitter problem; if not, it is maybe a packet loss problem and the fault lies in the home network, the residential gateway, or beyond.

```
.Components.FrontEnd.1.IP.Dejittering.BufferSize = 100
```

I.5.1 Explicit connect to service

The technician can force the STB to connect to a specified service in both the DVB-T and IP cases. The following is a typical sequence of events.

- User has a problem with the IPTV service and calls customer support.
- Technician decides to monitor performance data while connecting to an IPTV test service.
- Performance data is currently being collected with a sample interval of 3600 (an hour) so technician decides to force current sample to complete early, to read the existing collected data, and to change the sample interval to 30.

```
.ServiceMonitoring.ForceSample = True
<read the existing collected data>
.ServiceMonitoring.SampleInterval = 30
```

- Technician connects to an IPTV test service. The STB ensures that the test service is delivered in an AV stream that is regarded as a “main stream” for performance data collection purposes.

```
.Components.FrontEnd.1.IP.ServiceConnect.
    URI = "udp://224.112.15.18:3257?StrCtrl=IGMP
        ?StrTrspCtrl=AL-FEC?MuxType=MPEG2-TS"
```

- Technician determines, if not already known, that the IPTV ServiceMonitoring Main-Stream instance is #1.
- Technician monitors performance data; a new sample is added every 30 seconds.
- Technician diagnoses and fixes the problem.
- Technician restores sample interval of 3600.

```
.ServiceMonitoring.SampleInterval = 3600
```

Alternatively, the technician could choose to monitor Total statistics, in which case the Sample statistics collection could continue uninterrupted. However, this would involve more network traffic, since the technician would have to simulate sample intervals by reading Total statistics at regular intervals, and would involve more complex ACS logic.

I.6 Performance Management

Performance management is automated monitoring for the purpose of identifying and avoiding possible future faults and service calls.

In an end-to-end AV performance monitoring model, the STB is not the only node that provides performance measurements, and it is likely that STB-based performance monitoring will be targeted at the performance of the last mile and the home network. The BasicPerfMon profile is targeted at such applications, and the ECPeefMon, VideoPerfMon and AudioPerfMon profiles include a wider range of statistics.

I.7 Fault Management

Fault Management is the action taken by the ACS when it receives a fault notification or when it detects a fault via regular polling.

For example, all Component, AV Stream and AV Player objects have a Status parameter that indicates when a component is in an error state and which in some cases can indicate further details of the error. The ACS can monitor or poll such parameters and can initiate further investigations. Such investigations can be similar to the trouble management investigations but, because no user is involved (indeed the user may be using the STB), need to be less invasive. In some cases it might be appropriate for a technician to call a user, in which case the investigation would become a trouble management investigation.

I.8 AV Stream and AV Player examples

The following examples correspond to the flows in Figure 7, which is nearly the same as Figure 1 (flows 1 and 2 were management flows that are not relevant here and have been omitted).

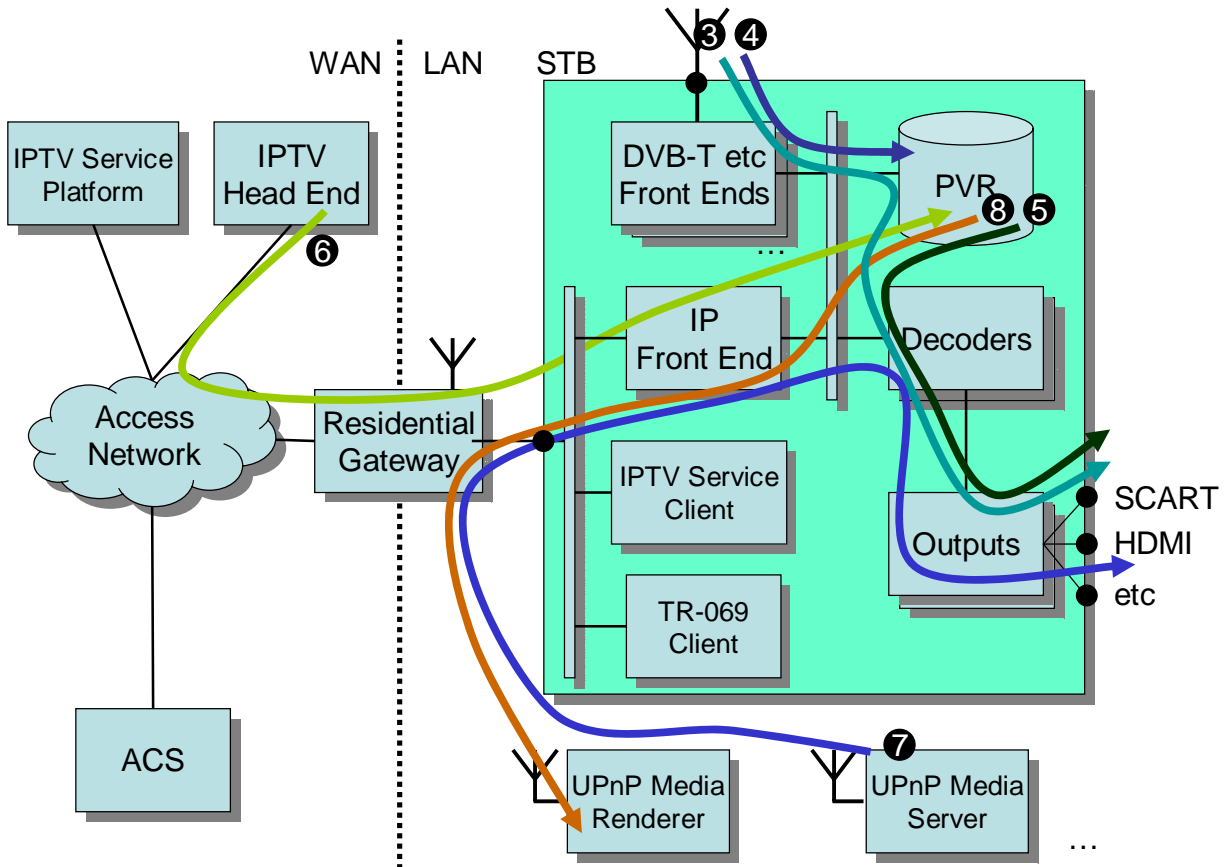


Figure 7/TR-135 – STB Context (same as Figure 1/TR-135)

For the purpose of these examples, assume that a hybrid IPTV / DTT STB has the following components:

```
.Components.FrontEnd.1.
  Name = "ip"
.Components.FrontEnd.2.
  Name = "dvb0"

.Components.AudioDecoder.1.
  Name = "MPEG2"
  AudioStandard = "MPEG2-AAC-LC"
.Components.AudioDecoder.2.
  Name = "Dolby"
  AudioStandard = "DTS"

.Components.VideoDecoder.1.
  Name = "MPEG2"
  MPEG2Part2 = .Capabilities.VideoDecoder.MPEG2Part2.
  ProfileLevel.1
```

```

.Components.VideoDecoder.2.
    Name = "VC1"
    SMPTEVC1 = .Capabilities.VideoDecoder.SMPTEVC1.
                                                ProfileLevel.1

.Components.AudioOutput.1.
    Name = "Digital"
    AudioFormat = "DIGITAL-OPTICAL-SP/DIF"
.Components.AudioOutput.2.
    Name = "Analog"
    AudioFormat = "ANALOG-5.1-CHANNELS"
.Components.AudioOutput.3.
    Name = "RF"
    AudioFormat = "RF"

.Components.VideoOutput.1.
    Name = "Digital"
    VideoFormat = "DVI"
.Components.VideoOutput.2.
    Name = "Analog"
    VideoFormat = "CVBS"
.Components.VideoOutput.3.
    Name = "RF"
    VideoFormat = "RF"

```

Also assume that it supports two AV Players, the first of which is tied to the digital Outputs, and the second of which is tied to the Analog and RF Outputs:

```

.AVPlayers.AVPlayer.1.
    Name = "Digital"
    AudioOutputs = .Components.AudioOutput.1
    VideoOutputs = .Components.VideoOutput.1

.AVPlayers.AVPlayer.2.
    Name = "Analog"
    AudioOutputs = .Components.AudioOutput.2,
                  .Components.AudioOutput.3
    VideoOutputs = .Components.VideoOutput.2,
                  .Components.VideoOutput.3

```

Finally, assume that it supports two AV Streams (real implementations will undoubtedly support a lot more than this), which will initially be disabled.

```

.AVStreams.AVStream.1.Status = "Disabled"
.AVStreams.AVStream.2.Status = "Disabled"

```

All the above would be statically configured, and (if the parameters are supported by the STB) the ACS is able to enable and disable Components and AV Players. It cannot enable and disable

AV Streams because these are enabled and disabled as a result of user interactions with the STB or with other home network devices.

I.8.1 Flow 3: DVB-T Front End to Analog AV Player

The user chooses to use the DVB-T Front End (#2) to watch BBC1 on the Analog AV Player (#2). Assuming that AV Stream #1 is used for this, this might result in the following parameter changes:

```
.AVStreams.AVStream.1.
  Status = "Enabled"
  Name = "BBC1"
  PVRState = "Disabled"
  FrontEnd = .Components.FrontEnd.2
  Inbound = <Empty>
  Outbound = <Empty>
  AudioDecoder = .Components.AudioDecoder.1
  VideoDecoder = .Components.VideoDecoder.1

.AVPlayers.AVPlayer.2.MainStream = .AVStreams.AVStream.1
```

Note that the flow is cunningly shown as just touching the PVR. In the above example, PVRState is "Disabled". However, it will be common for trick modes such as pause and time delay to be supported for such streams. For example, the following would indicate that live TV is being buffered by the PVR:

```
.AVStreams.AVStream.1.PVRState = "Play"
```

I.8.2 Flow 4: DVB-T Front End to PVR

If the program from the previous example was instead being recorded, no AV Player would have been involved, and the parameter changes might instead have been the following:

```
.AVStreams.AVStream.1.
  Status = "Enabled"
  Name = "BBC1"
  PVRState = "Stopped"
  FrontEnd = .Components.FrontEnd.2
  Inbound = <Empty>
  Outbound = <Empty>
  AudioDecoder = <Empty>
  VideoDecoder = <Empty>
```

Here the PVR must be recording because (a) the AV Stream is associated with a Front End, and (b) PVRState is not "Disabled".

I.8.3 Flow 5: PVR to Analog AV Player

If the program from the previous example is played on the Analog AV Player (#2), the parameter changes might have been the following:


```
.AVStreams.AVStream.1.
  Status = "Enabled"
  Name = "BBC1"
  PVRState = "Play"
  FrontEnd = <Empty>
  Inbound = <Empty>
  Outbound = <Empty>
  AudioDecoder = .Components.AudioDecoder.1
  VideoDecoder = .Components.VideoDecoder.1

.AVPlayers.AVPlayer.2.MainStream = .AVStreams.AVStream.1
```

It should be no surprise that the only difference between Flow 3 (in the time delay case) and Flow 5 is that there is no Front End. This simply reflects the fact that (loosely speaking):

- Flow 3 = Flow 4 + Flow 5

I.8.4 Flow 6: IPTV to PVR

This is the IPTV equivalent of Flow 4 and is therefore the same apart from the Front End (#1):

```
.AVStreams.AVStream.1.
  Status = "Enabled"
  Name = "BBC1"
  PVRState = "Stopped"
  FrontEnd = .Components.FrontEnd.1
  Inbound = .Components.FrontEnd.1.Inbound.1
  Outbound = <Empty>
  AudioDecoder = <Empty>
  VideoDecoder = <Empty>
```

There is only a single IP Front End, and it can handle multiple Inbound and Outbound streams. The Inbound object (#1) might include the following parameters:

```
.Components.FrontEnd.1.Inbound.1.
  Status = "Enabled"
  StreamingTransportProtocol = "RTP"
  StreamingTransportControlProtocol = "RTCP"
  MultiplexType = "MPEG2Program"
  URI = "urn:ietf:tbd"
```

The format of the URI parameter is not specified as part of the STB data model, but should as far as possible follow existing standards.

I.8.5 Flow 7: Home Network Media Server to Digital AV Player

The scenario here is that the STB is a UPnP Media Renderer. This is the UPnP AV equivalent of Flow 3 and is therefore similar apart from the Front End (#1). Also, as it happens, it uses the other AV Stream (#2) and the Digital AV Player (#1):

```
.AVStreams.AVStream.2.
  Status = "Enabled"
  Name = "The Princess Bride"
  PVRState = "Disabled"
  FrontEnd = .Components.FrontEnd.1
  Inbound = .Components.FrontEnd.1.Inbound.2
  Outbound = <Empty>
  AudioDecoder = .Components.AudioDecoder.2
  VideoDecoder = .Components.VideoDecoder.2

.AVPlayers.AVPlayer.1.MainStream = .AVStreams.AVStream.2
```

The Inbound object (#2) might include the following parameters:

```
.Components.FrontEnd.1.Inbound.2.
  Status = "Enabled"
  StreamingTransportProtocol = "HTTP"
  MultiplexType = "VOB"
  URI = "urn:upnp-org:tbd"
```

I.8.6 Flow 8: PVR to Home Network Media Renderer

The scenario here is that the STB is a UPnP Media Server. This flow is different from the others in that it involves an IP Outbound flow.

```
.AVStreams.AVStream.2.
  Status = "Enabled"
  Name = "The Princess Bride"
  PVRState = "Play"
  FrontEnd = <Empty>
  Inbound = <Empty>
  Outbound = .Components.FrontEnd.1.Outbound.1
  AudioDecoder = <Empty>
  VideoDecoder = <Empty>

.Components.FrontEnd.1.Outbound.1.
  Status = "Enabled"
  URI = "urn:upnp-org:tbd"
```

I.8.7 Flow 3 + Flow 7: Two AV Streams in Digital AV Player

The user might choose to view Flow 3 as the main picture, and to view Flow 7 as a PIP, both in the Digital AV Player (#1). This doesn't involve any change to the AV Streams, just to the AV Player:

```
.AVPlayers.AVPlayer.1.
  MainStream = .AVStreams.AVStream.1
  PIPStreams = .AVStreams.AVStream.2
```