TECHNICAL REPORT

TR-135

Data Model for a TR-069 Enabled STB

Issue: 1 Amendment 3
Issue Date: November 2012

© The Broadband Forum. All rights reserved.

Notice

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

This Broadband Forum Technical Report is provided AS IS, WITH ALL FAULTS. ANY PERSON HOLDING A COPYRIGHT IN THIS BROADBAND FORUM TECHNICAL REPORT, OR ANY PORTION THEREOF, DISCLAIMS TO THE FULLEST EXTENT PERMITTED BY LAW ANY REPRESENTATION OR WARRANTY, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, ANY WARRANTY:

(A) OF ACCURACY, COMPLETENESS, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT, OR TITLE;
(B) THAT THE CONTENTS OF THIS BROADBAND FORUM TECHNICAL REPORT ARE SUITABLE FOR ANY PURPOSE, EVEN IF THAT PURPOSE IS KNOWN TO THE COPYRIGHT HOLDER;
(C) THAT THE IMPLEMENTATION OF THE CONTENTS OF THE TECHNICAL REPORT WILL NOT INFRINGE ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.

By using this Broadband Forum Technical Report, users acknowledge that implementation may require licenses to patents. The Broadband Forum encourages but does not require its members to identify such patents. For a list of declarations made by Broadband Forum member companies, please see http://www.broadband-forum.org. No assurance is given that licenses to patents necessary to implement this Technical Report will be available for license at all or on reasonable and non-discriminatory terms.

ANY PERSON HOLDING A COPYRIGHT IN THIS BROADBAND FORUM TECHNICAL REPORT, OR ANY PORTION THEREOF, DISCLAIMS TO THE FULLEST EXTENT PERMITTED BY LAW (A) ANY LIABILITY (INCLUDING DIRECT, INDIRECT, SPECIAL, OR CONSEQUENTIAL DAMAGES UNDER ANY LEGAL THEORY) ARISING FROM OR RELATED TO THE USE OF OR RELIANCE UPON THIS TECHNICAL REPORT; AND (B) ANY OBLIGATION TO UPDATE OR CORRECT THIS TECHNICAL REPORT.

Broadband Forum Technical Reports may be copied, downloaded, stored on a server or otherwise re-distributed in their entirety only, and may not be modified without the advance written permission of the Broadband Forum.

The text of this notice must be included in all copies of this Broadband Forum Technical Report.

 © The Broadband Forum. All rights reserved

Issue History

Issue Number	Approval Date	Publication Date	Issue Editors	Changes
Issue 1	December 2007		Marco Quacchia, Telecom Italia William Lupton, 2Wire Gilles Straub, Thomson	Original
Issue 1 Amendment 1	November 2010		Marco Quacchia, Telecom Italia Gilles Straub, Technicolor	1. Aligned to TR-160 2. Added force monitoring operation mode 3. Added CDS (Content Download Service) objects 4. Added HDMI and S/PDIF modeling
Issue 1 Amendment 2	April 2011		Marco Quacchia, Telecom Italia	1. Modified paragraph 5.6.2 "Service Provider" as proposed by ITU-T Q13/16 incoming liaison bbf12011.108.00
Issue 1 Amendment 3	26 November 2012	16 January 2013	Jean-Didier Ott, France Telecom	Added description of histograms of loss events data (I.3.6.1)

Comments or questions about this Broadband Forum Technical Report should be directed to info@broadband-forum.org.

Editor	Jean-Didier Ott	France Telecom
BroadbandHome™ Working Group Chairs	Jason Walls John Blackford	QA Cafe Pace
Chief Editor	Michael Hanrahan	Huawei Technologies

 © The Broadband Forum. All rights reserved

TABLE OF CONTENTS

 © The Broadband Forum. All rights reserved

 © The Broadband Forum. All rights reserved

 © The Broadband Forum. All rights reserved

List of Figures

 © The Broadband Forum. All rights reserved

Executive Summary

TR-135, *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. TR-135 also includes 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, including for instance the URL of the IPTV Service Discovery server, but its main functions are configuration of STB parameters for trouble management and collection of statistics for QoS / QoE monitoring.

TR-135 specification of monitoring statistics is aligned with Broadband Forum's TR-160, *Requirements for IPTV Performance Monitoring and Diagnostics*.

 © The Broadband Forum. All rights reserved

1 Purpose and Scope

1.1 Purpose

TR-135, *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 inTR-069 [1] and TR-106 [2]. IPTV Services include live IPTV and VoD (Video on Demand). VoD includes, among other services, what is defined in this document as CDS (Content Download Service).

TR-135 covers the data model for describing an STB device as well as rules regarding notifications on parameter value change. General remote management use cases are also described; including standard data model profiles that would typically be seen while remotely managing a device of this nature are included.

TR-135 defines the 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 TR-106. In the context of TR-106, 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 TR-106. 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.

1.2 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. Data flows 1 and 2 are management and control data flows, data flows 3 to 9 are media data flows. The diagram also provides the examples of Section I.8.

 © The Broadband Forum. All rights reserved

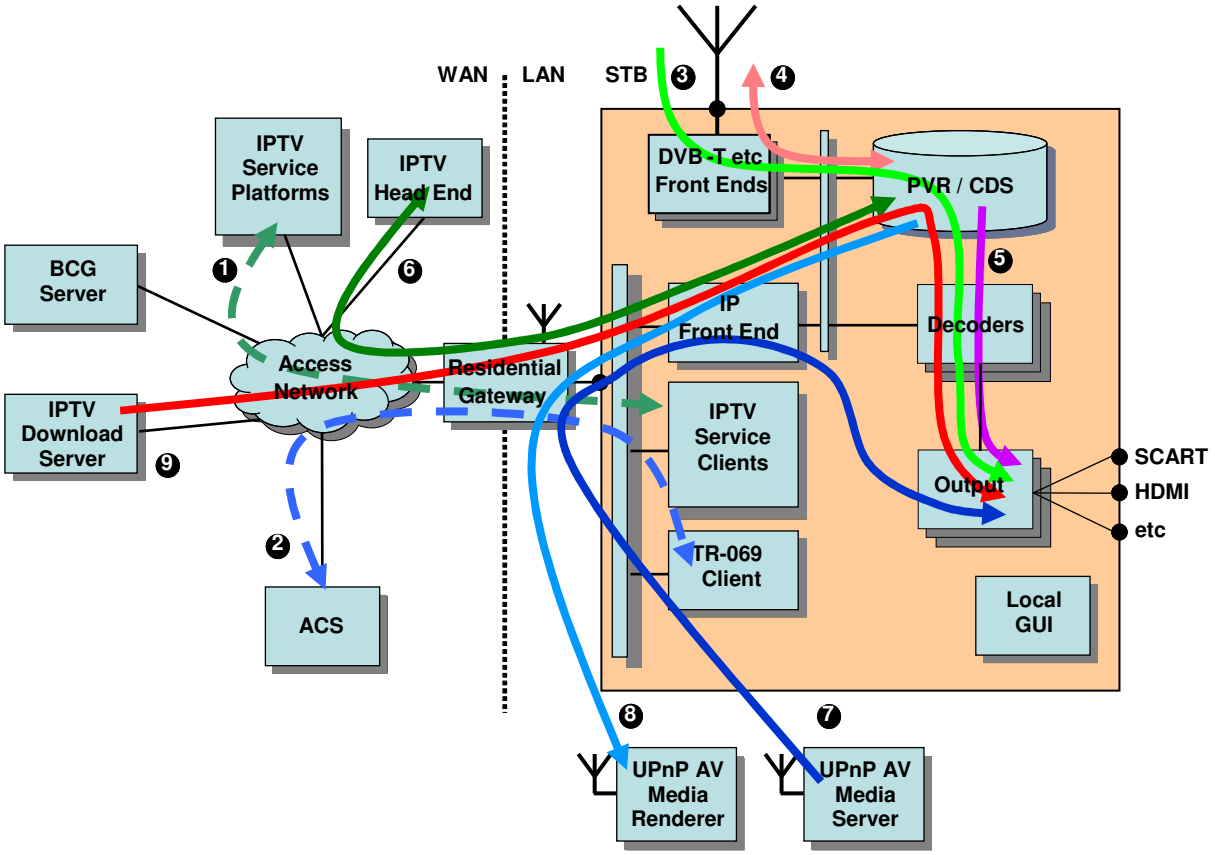


Figure 1– STB Context

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). IPTV content may be provided to the STB in the form of streaming (for instance live IPTV, VoD) or download data (for instance, the CDS service). Content from other networks is provided generally in the form of streams. The STB receives data 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 presented on which display device. 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). Audio outputs too can be single or multi channel analog (headphones, loudspeakers, line), digital (S/PDIF) or networked. 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 Broadband Forum. All rights reserved

the recording of VoD (Video on Demand) events, in which a piece of (protected) content is streamed to the STB, or Push VoD (e.g. push mode CDS), in which a piece of (protected) content is downloaded to the PVR for possible subsequent viewing by the end user. Push VoD download is initiated by the operator. Along with this, Pull VoD is also available (e.g. pull mode CDS), in which content download is carried out by the operator not autonomously but upon user request.

As shown in Figure 1 the IPTV service is provided by one or more (proprietary) IPTV service delivery platforms. TR-069 remote management is a separate platform. There is no precise rule that governs the splitting of functions between TR-069 remote management and IPTV service delivery platforms. As a rule of thumb, though, it is reasonable to assume that IPTV service platforms are mainly involved in media processing, rights management/conditional access and the bulk of provisioning, whereas the ACS is mainly in charge of monitoring operation and performance and providing support to trouble management, i.e. enabling a trained technician to check specific STB parameters and carry out diagnostic tests. The ACS might also carry out a limited amount of provisioning.

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 DLNA Media Server, 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 TR-126 [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 TR-106 [2].
- Accommodate STB devices that are standalone, i.e. implemented in separate hardware devices, as defined in TR-106 [2].

 © The Broadband Forum. All rights reserved

2 References and Terminology

2.1 Conventions

In this Technical Report, several words are used to signify the requirements of the specification. These words are always capitalized. More information can be found be in RFC 2119 [6].

MUST	This word, or the term “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 term “RECOMMENDED”, means that there could exist valid reasons in particular circumstances to ignore this item, but the full implications need to be understood and carefully weighed before choosing a different course.
SHOULD NOT	This phrase, or the phrase "NOT RECOMMENDED" means that there may exist valid reasons in particular circumstances when the particular behavior is acceptable or even useful, but the full implications need to be understood and the case carefully weighed before implementing any behavior described with this label.
MAY	This word, or the term “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.

2.2 References

The following references are of relevance to this Technical Report. At the time of publication, the editions indicated were valid. All 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 references listed below.

A list of currently valid Broadband Forum Technical Reports is published at http://www.broadband-forum.org.

[1] TR-069 Amendment 4, *CPE WAN Management Protocol*, Broadband Forum Technical Report

 © The Broadband Forum. All rights reserved

[2] TR-106 Amendment 6, *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 Issue 1 Amendment 1, *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*

 © The Broadband Forum. All rights reserved

[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

[39] ITU-T Rec. H.770 *"Mechanisms for service discovery and selection for IPTV"*

 © The Broadband Forum. All rights reserved

2.3 Definitions

The following terminology is used throughout this Technical Report.

Auto-Configuration Server	This is a component in the broadband network responsible for auto-configuration of the CPE for advanced services.
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 Player	One or more *AV Streams* (one *AV Main Stream*, possibly several PIP video streams, one audio stream) connected to zero or more *AudioOutput*s and zero or more *VideoOutput*s.
AV Main Stream	This is used in two different contexts: • An *AV Stream* that is destined for the main screen rather than a *PIP*. • An *AV Stream* associated with a .ServiceMonitoring.MainStream.{i} instance.
AV Stream	One of the following chains, corresponding to a channel (or program) that the user is currently watching or recording (see also *AV Player*): • *FrontEnd* → {*AudioDecoder, VideoDecoder}.* • *FrontEnd* → *PVR* → {*AudioDecoder, VideoDecoder}.* • *FrontEnd* → *PVR.* • *PVR* → {*AudioDecoder, VideoDecoder}.* • *PVR* → *FrontEnd.* • *FrontEnd* → *FrontEnd.* • *FrontEnd* → *PVR* → *FrontEnd.*
FrontEnd	A logical component that converts an incoming multiplex from channel format, e.g. *OFDM* modulation for *DTT* or *RTP*/*UDP* encapsulation for IP, to a continuous (still multiplexed) bit stream.
Interconnection Bus	Conceptual bus that interconnects *FrontEnd*s, *AudioDecoder*s, *VideoDecoder*s and the *PVR*, multiplexing/demultiplexing and handling timing and synchronization, as necessary.
Main Stream	*AV Main Stream*.
Parameter	A name-value pair representing a manageable CPE parameter made accessible to an ACS for reading and/or writing.

 © The Broadband Forum. All rights reserved

Player	*AV Player*.
Set Top Box.	This device contains Audio and Video decoders and is intended to be connected to Analog TV and /or Home Theaters.
Stream	*AV Stream*.
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).

2.4 Abbreviations

This Technical Report uses the following abbreviations:

ACS	Auto-Configuration Server
ATSC	Advanced Television Systems Committee. Digital television standards, primarily adopted in North America.
AV	Audio and Video.
BCG	Broadband Content Guide.
BTV	Broadcast Television.
CA	Conditional Access.
CAB	Cable.
CBR	Constant Bit Rate.
CDS	Content Download Service.
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	*DVB* Cable standard
DVB-S	*DVB* Satellite standard
DVB-T	*DVB* Terrestrial; *DVB DTT standard*

 © The Broadband Forum. All rights reserved

DVD	Digital Versatile Disk, or Digital Video Disk.
EC	Error Correction.
ES	*MPEG* Elementary Stream. *An audio, video (or text) stream*
FTTP	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.
IPTV	Internet Protocol Television.
ISDB	Integrated Services Digital Broadcasting. *Japanese digital television and audio broadcasting format.*
MPEG	Moving Picture Experts Group.
MPEG2-TS	MPEG-2 Transport Stream.
OFDM	Orthogonal Frequency Division Multiplexing.
OSS	Operations Support Systems.
PCR	*MPEG2-TS* Program Clock Reference.
PES	MPEG-2 Packetized Elementary Stream.
PIP	Picture In Picture.
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 et 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.

 © The Broadband Forum. All rights reserved

TFTP	Trivial File Transfer Protocol, *IETF RFC 1350*.
TR	Technical Report
TS	*MPEG2-TS*.
UPnP	Universal Plug'n'Play.
USB	Universal Serial Bus.
VCR	Video Cassette Recorder.
VoD	Video on Demand.
WG	Working Group
WLAN	Wireless Local Area Network.

 © The Broadband Forum. All rights reserved

3 Technical Report Impact

3.1 Energy Efficiency

TR-135 has no impact on energy efficiency.

3.2 IPv6

TR-135 has no impact on IPv6.

3.3 Security

TR-135 has no impact on security.

3.4 Privacy

TR-135 has no impact on privacy.

 © The Broadband Forum. All rights reserved

4 Use Cases

A number of remote management use cases can be considered for the scenario of Section 1.2. 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 signaling 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 Platforms.

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

 © The Broadband Forum. All rights reserved

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. The technician can also request the STB to display color bars, to check the output modules and the STB-display connection.

Depending on the cases, the technician can carry out actions, such as forcing operation of error correction to try and improve network performance, 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 “passive” measurements done at the service level. These might be “in service”, i.e. the measurements are carried out on the stream that contains the user content, as well as “out of service”, i.e. the measurements are carried out on a reference stream to which the STB is forced to connect when the user is not consuming content. Both of these are of fundamental importance to an operator in a number of cases, for example:

 © The Broadband Forum. All rights reserved

- 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 that 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.

 © The Broadband Forum. All rights reserved

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 modeled 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.

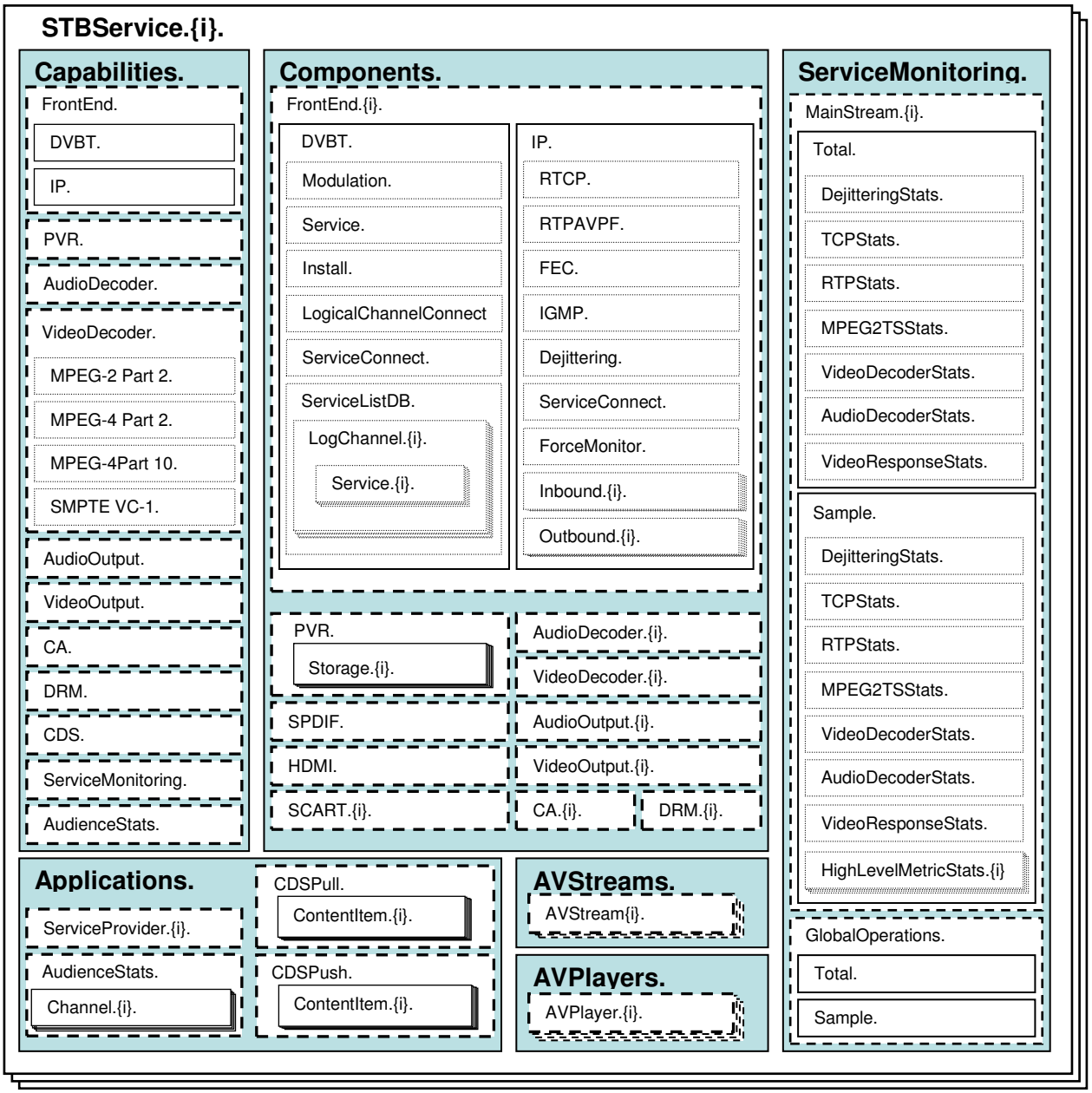


Figure 2– TR-135 – STBService object structure

 © The Broadband Forum. All rights reserved

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 modeled in .Capabilities.CCC.; each CCC instance is modeled 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 modeled 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 Broadband Forum. All rights reserved

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 modeled.

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 interfaces (which are modeled by the generic objects defined in TR-106 [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 modeled 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 MEPG2-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 modeled via TR-140 [4] StorageService instances within the STB. As noted in Section 1.2, the PVR also performs standard (VCR-like) recording functions as well as advanced

 © The Broadband Forum. All rights reserved

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/PDIF 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 modeled).

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). Video Outputs can also provide color bar test patterns to check the operation of the display device and the existence and quality of the connection between STB and display. Adaptation of the video to the specified

 © The Broadband Forum. All rights reserved

output format can include various actions like digital-to-analog conversion, resizing, aspect ratio conversion and 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 modeled).

5.2.7 SCART

The SCART (Société des Constructeurs d'Appareils Radiorécepteurs et Téléviseurs) 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).

5.2.10 HDMI

A HDMI component models features of the STB HDMI functionality, such as the output resolution mode and value. It also contains a sub object providing details on the connected HDMI display device (typically a TV set): for instance, the delay the STB is requested to apply to audio to preserve "lip sync" synchronization. This is a read-only parameter for the ACS, and is communicated to the STB through the HDMI interface.

5.2.11 SPDIF

A SPDIF component models features of the S/PDIF functionality of the STB. It allows the ACS to force the audio format to be downmixed into stereo PCM in the case an external audio

 © The Broadband Forum. All rights reserved

amplifier does not support multi-channel audio format decode. It also contains some parameters to manage the audio delay (to preserve lip-sync).

5.3 AV Streams

An AV Stream is modeled 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 modeled 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 modeling and thus is not present in the STB data model.

 © The Broadband Forum. All rights reserved

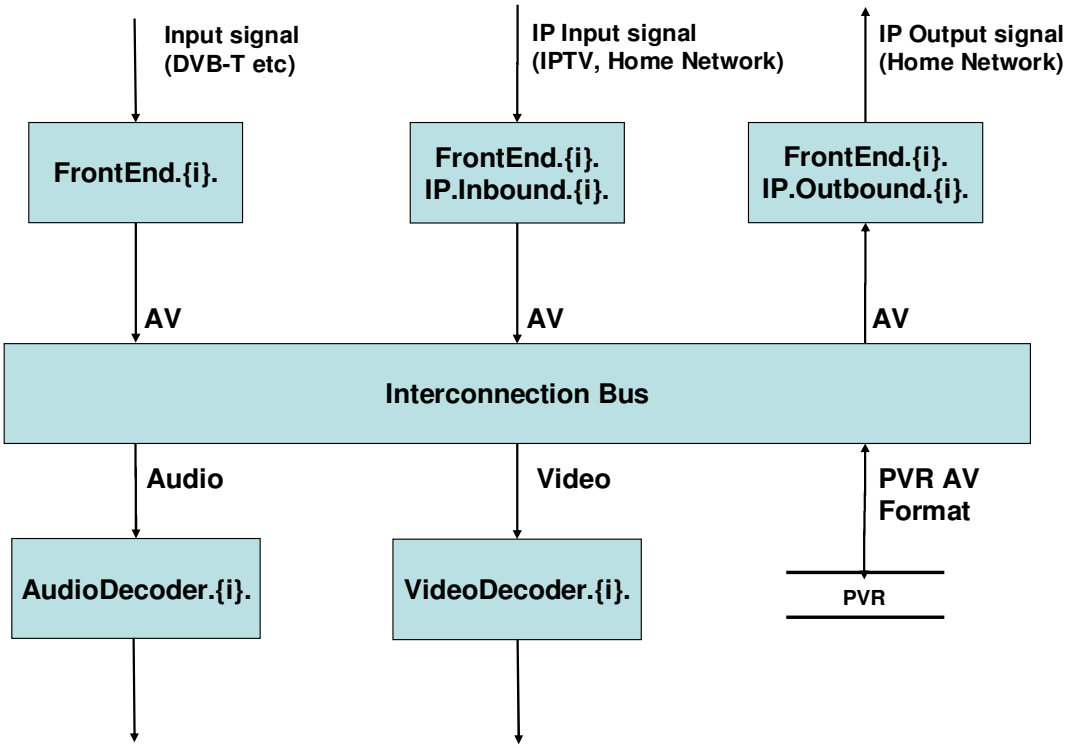


Figure 3– Logical AV Streaming Model

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

 © The Broadband Forum. All rights reserved

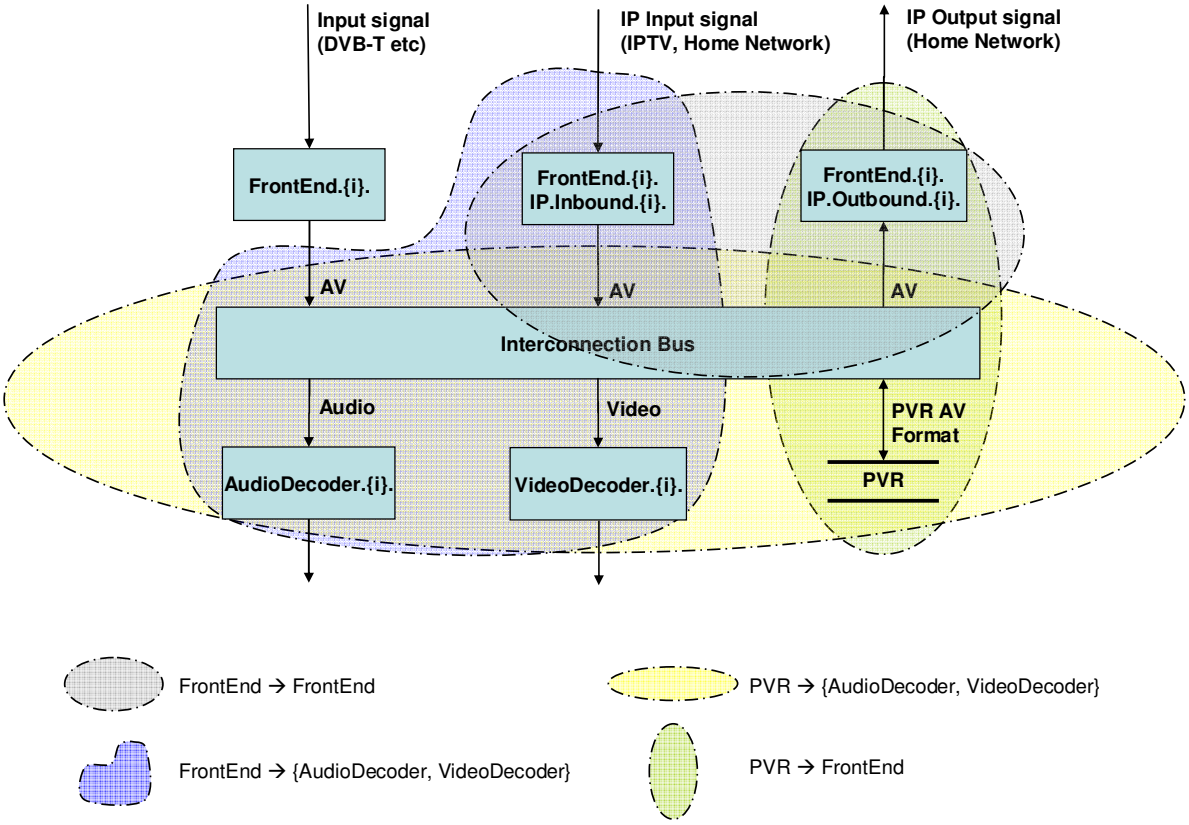


Figure 4 – Logical AV Streaming Model with Example Streams

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 modeled 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 modeled by means of a single AV Player referencing the AV Stream object and multiple Audio Output and Video Output objects. A case

 © The Broadband Forum. All rights reserved

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.

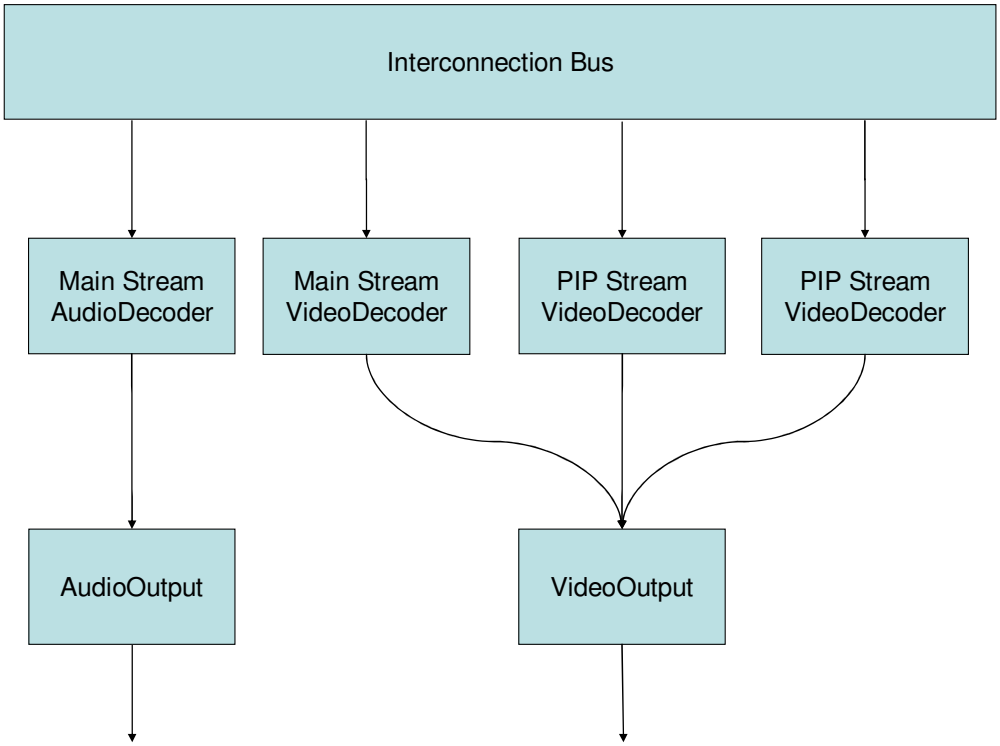


Figure 5 – Logical AV Player Model

5.5 Service Monitoring

Service Monitoring statistics are collected on parameters describing the STB global operation and 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 eight categories: De-jittering, RTP, MPEG2-TS, TCP, 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:

 © The Broadband Forum. All rights reserved

- 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, organized by channel.

5.6.2 ServiceProvider

This object contains Service Provider specific parameters. At the moment it is used by the ACS to configure the URL of the Service Discovery server. An example of Service Discovery server functionality is contained in [39]. By connecting to the Service Discovery server the STB gets information about one or more Service Providers. The object shows the service provider description retrieved by the STB. From this the user can get information about the Service Provider programming.

5.6.3 CDSPush and CDS Pull

The CDS (Content Download Service) allows the operator to download pieces of content (typically audiovisual files, but not limiting to this) to an operator-managed area of the STB disk. The CDSPush and CDSPull objects model the Push and Pull mode, respectively, of this service. Basically these objects provide the ACS with a list of the content items downloaded into the STB. The ACS can also configure these lists to request deletion of specific items for troubleshooting purposes. For the sake of privacy the STB display of the CDSPull object's content items list might be subject to the user authorization, that could be granted for instance via the STB local user interface.

 © The Broadband Forum. All rights reserved

6 STBService Parameter Definitions

The normative definition of the STBService data model is split between several DM Instance documents (see Annex A/TR-106 [2]) and is published at http://www.broadband-forum.org/cwmp. For a given revision of the data model, the corresponding TR-135 XML document defines the STBService model itself and imports additional components from the other XML documents listed. Each TR-135 HTML document is a report generated from the XML files, and lists a consolidated view of the STBService data model in human-readable form.

 © The Broadband Forum. All rights reserved

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. It then presents various examples of AV Stream and AV Player usage and, finally, a description of the CDSPush and CDSPull objects operation.

I.1 Profile usage

The following profiles are defined in the Data Model. The minimum STBService object Required Version: is 1.1 for DiagPerfMon:1 and 1.0 for all the other profiles defined here.

- **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.
- **HistoPerfMon**: adds histogram-style RTP statistics.
- **DiagPerfMon**: adds response time and transaction completion statistics.
- **ECPerfMon**: adds RTP EC (Error Correction) statistics.
- **HistoECPerfMon**: adds histogram-style RTP EC 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.

 © The Broadband Forum. All rights reserved

- **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 top-level 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.

Objects that relate to specific applications, such as .Applications.CDSPush or .Applications.-CDSPull, usually get created on a firmware upgrade or possibly software module install. When

 © The Broadband Forum. All rights reserved

these objects contain tables, as is the case of the CDS objects, the STB autonomously determines how many rows (object instances) the tables should contain. For instance, a STB supporting the CDSPush service might expose a content item table with 10 content items with a data model like this:

```
.Applications.CDSPush
.Applications.CDSPush.Reference
.Applications.CDSPush.ItemNumberOfEntries
.Applications.CDSPush.ContentItem.1.ContentReferenceID
.Applications.CDSPush.ContentItem.1.VersionNumber
.Applications.CDSPush.ContentItem.1.DeleteItem
...
.Applications.CDSPush.ContentItem.10.ContentReferenceID
.Applications.CDSPush.ContentItem.10.VersionNumber
.Applications.CDSPush.ContentItem.10.DeleteItem
```

The same considerations hold for CDSPull object.

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}, AV-Streams.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-

 © The Broadband Forum. All rights reserved

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 DVB 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 globally for the STB as well as for specific AV streams that the STB designates as “main streams”; the latter 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. Global Operation statistics are collected regardless of which stream is the Active Stream.

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 can be collected only while the end user is consuming content or all the time, regardless of when the end user is doing. To achieve this the ACS can configure the STB to be forced to connect to a test URL when no content stream is available. This functionality is called the force monitoring mode

 © The Broadband Forum. All rights reserved

Statistics collection is performed independently for parameters describing the STB global operation as well as 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.

For global operation and main streams 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 Sample statistics mechanism is active even when no stream is active for a configured service type.

The STB can collect various categories of statistics: de-jittering, RTP, MPEG2-TS, TCP, 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 there are the single instance GlobalOperations object and a multi-instance MainStream object. The GlobalOperations object can never be disabled. Each MainStream 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 GlobalOperations object and 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. GlobalOperations and MainStream.{i}. Sample statistics have common configuration parameters.

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.

 © The Broadband Forum. All rights reserved

- **VoD**: WAN-sourced VoD stream.
- **IP**: WAN-sourced IP data (includes IPTV and VoD).
- **TEST**: WAN-sourced force monitoring IP stream from ForceMonitorURI (may be either IPTV or VoD).
- **IP-All**: Any WAN-sourced IP stream (includes user and force monitoring streams, 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.

CDS content shall fit under the VoD ServiceType. VoD service is not limited to CDS.

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"
```

 © The Broadband Forum. All rights reserved

```
        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 synchronized 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.
- **EventsPerSampleInterval**: Event specific parameters (samples are made up of events).
- **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 items, of which the first one is the oldest and the last one is the most recent. Each item may be, in turn, either one single value or (in case events per sample are listed) a comma-separated list of values (the way to represent lists of lists is specified in TR-106 [2]). 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:

 © The Broadband Forum. All rights reserved

- **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:

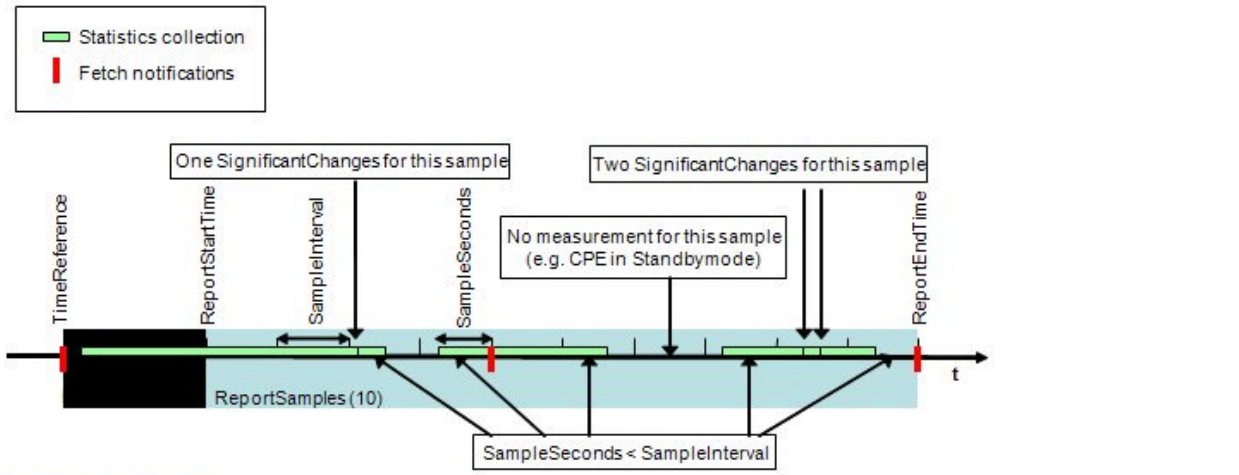
- **SampleSeconds**: This is similar to the top-level SampleSeconds parameter, but this one relates directly to the availability of de-jittering statistics; the value of this parameter, for a given sample interval, will always be less than or equal to the value of the top-level SampleSeconds parameter.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

If the STB is not receiving any data (because for instance in Standby mode and the force monitoring mode is not enabled), there will be no active AV streams, but sample intervals will continue to be timed as normal. The fact that the STB was not receiving any data will be indicated by the fact that all the SampleSeconds values will be zero when the STB was not receiving any data for the entire sample interval, and will be less than SampleInterval when the STB was not receiving any data for only part of the sample interval.

I.3.4.3 Example configuration

Figure 6 illustrates all the above parameters.

 © The Broadband Forum. All rights reserved



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

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"

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

VideoResponseStats.VoDControlResponse = "125%2C357%2C88%2C596,127%2C964%2C1579,473,512%2C124,,,375"

Figure 6 – Sample statistics Parameters

In Figure 6 parameters from two groups of statistics are reported (DejitteringStats and VideoResponseStats). SampleSeconds is assumed to be the same for both statistics though this need not be true in all cases.

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.

 © The Broadband Forum. All rights reserved

- 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.
- Note that the escape mechanism used in VoDControlResponse to represent a list of lists is defined in TR-106 [2] and subject to change in future versions of that document.

I.3.5 Sample statistics details

In this section, detailed Sample statistics behavior 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
…
```

 © The Broadband Forum. All rights reserved

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 does not 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 behavior, 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 = “0,0,1,0,0,0,0,0,2,0”
```

 © The Broadband Forum. All rights reserved

```
        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 = “0,1,0,0,0,0,0,2,0,0”
        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”
```

 © The Broadband Forum. All rights reserved

```
        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.

 © The Broadband Forum. All rights reserved

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)

 © The Broadband Forum. All rights reserved

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 synchronized with the existing instances.

For example, suppose that Sample statistics collection is currently enabled, MainStream instance #1 is enabled, and it is 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 synchronized 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.
     Enable = True
     MetricName = "MOS-V"
     MetricThreshold = 50000
```

 © The Broadband Forum. All rights reserved

MetricThreshold indicates that, each time that the value of the metric is measured as less than 50000, MetricFailures should be incremented. The ACS can enable Active Notification on MetricFailures 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 the STB implementation to ensure that notifications were generated only at appropriate times and at appropriate rates.

I.3.5.10 Event collection

Events are the elementary observations that are accounted for in ServiceMonitoring statistics: for instance, events accounted for by parameter PortalResponse are the individual portal response times measured by the STB for each portal access. For a restricted number of parameters (PortalResponse, VideoSystemResponse, VoDControlResponse) the data model lists the individual events, i.e. the values on which the statistics are calculated, in addition to the usual statistical indicators (for the above parameters there are Maximum, Average and Minimum over a sample interval). The ACS may configure a maximum number of events per sample interval the STB is requested to store. If the ACS determines that the maximum number of events per sample interval for a given class of STBs should be e.g. 3, then the ACS configures the following parameter as indicated:

```
.ServiceMonitoring.EventsPerSampleInterval = 3
```

The STB can use the Capabilities Object to indicate the maximum number of events per sample interval that it can support.

```
.Capabilities.ServiceMonitoring.
     MaxEventsPerSampleInterval = 10
```

It is not mandatory for the STB to specify this parameter's value. In case the STB does not put any restriction on this parameter then it should set this value to -1.

The following example describes the case of three accesses to the Operator's Portal in the first Sample Interval, no access in the second, one in the third and three again in the fourth:

```
.STBService{i}.ServiceMonitoring.GlobalOperation.Sample.
     PortalResponse = 123%2C134%2C247,,214,126%2C189%2C235
```
[1]

If one more access had taken place in the fourth sample interval, with delay of 357 ms, and maximum three events per sample interval were configured then the recorded data would possibly be modified this way:

```
.STBService{i}.ServiceMonitoring.GlobalOperation.Sample.
     PortalResponse = 123%2C134%2C247,,214,189%2C235%2C357
```

[1] Note that the escape mechanism used in PortalResponse to represent a list of lists is defined in TR-106 [2] and subject to change in future versions of that document.

 © The Broadband Forum. All rights reserved

I.3.5.11 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 Gmin - 1 received packets (a sequence of Gmin received packets terminates the Loss Event, and so is not an island). Gmin is writable, so that the Loss Event can be configured as best suits the operator.

For example, if Gmin is 1 then there can be no islands and any sequence of lost packets is a loss event. If Gmin 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 Gmin = 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 Gmin 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.

 © The Broadband Forum. All rights reserved

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.3.6.1 Histograms

A *Histogram* is a parameter derived from another parameter, which is a counter of *Loss Events*. Whereas that counter counts all such events, the histogram counts the events separately based on some property of those events (e.g. the number of packets lost during the event or the duration of the event). For each histogram, another parameter defines ranges of values of that property. The resulting histogram is a comma-separated list of counters; each of those counters corresponds to one of the defined ranges and counts the events for which the property falls in that range. Thus, the sum of the values of the counters in the histogram equals to the value of the counter the histogram is derived from.

Three histograms are defined in the STB data model. They provide detailed information about the frequency and amplitude of packet losses:

- PacketsLostByEventHistogram: distribution of the *Loss Events* according to their size (in packets). PacketsLostByEventHistogramIntervals defines the size ranges.
- DelayBetweenLossEventsHistogram: distribution of the *Loss Events* according to the delay (in ms) since the last *Loss Event*. DelayBetweenLossEventsHistogramIntervals defines the delay ranges.
- DurationSevereLossEventsHistogram: distribution of the *Severe Loss Events* according to their duration (in ms). DurationSevereLossEventsHistogramIntervals defines the duration ranges.

For each of those histograms, another one gives the same statistics measured before error correction. Each histogram appears in the Total and in the Sample statistics. The ranges definitions are common for Total and Sample.

For example, if PacketsLostByEventHistogramIntervals is set to "1,2,4,8,", a value of "30,35,13,24,3" for PacketsLostByEventHistogram denotes 30 *Loss Events* of exactly 1 packet, 35 of exactly 2 packets, 13 of 3 or 4 packets, 24 of 5 to 8 packets, 3 of 9 packets or more.

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

 © The Broadband Forum. All rights reserved

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 1.2 , 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**: enable/disable, operation mode (auto or forced), re-transmit timeout, re-transmit criteria.
- **FEC**: enable/disable, operation mode (auto or forced).
- **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.
- **IP force monitor**: enable/disable force monitoring mode, explicit service to connect to.
- **Audio Output**: audio level, cancel mute.
- **SPDIF**: Forces audio format over S/PDIF to be in PCM mode, audio delay operation mode (auto or manual).
- **Video Output**: video format.
- **HDMI**: resolution mode (auto or manual) and value.
- **SCART**: “presence” control signal.
- **CDSPush**, **CDSPull**: Delete specific content items.
- **AV Players**: preferred aspect ratio behavior, 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.

 © The Broadband Forum. All rights reserved

- **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:

 © The Broadband Forum. All rights reserved

```
.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.
```

 © The Broadband Forum. All rights reserved

```
PreferredAudioLanguage = “en-UK”
PreferredSubtitlingLanguage = “en-UK”
```

I.4.3 Error Correction operation mode

This section applies to both FEC and RTP/AVPF correction features. The STB may implement an automatic error correction mode, where it determines autonomously whether error correction must be enabled or not. The methods used by the STB to make this decision are not specified: for instance, the STB could base its decision on packet error rate measurements. The service provider may however decide to override the STB decision and either disable or force the error correction. This is the purpose of the FEC and RTPAVPF objects.

Assuming IP FrontEnd is number 3, setting the FEC operation mode in automatic mode would require:

```
.Components.FrontEnd.3.IP.FEC.Enable = True
.Components.FrontEnd.3.IP.FEC.OperationMode = “Auto”
```

Setting the FEC in forced mode (i.e. FEC has to run regardless of e.g. the packet loss measured in the STB) would require:

```
.Components.FrontEnd.3.IP.FEC.Enable = True
.Components.FrontEnd.3.IP.FEC.OperationMode = “Forced”
```

Disabling the FEC (i.e. no FEC regardless of e.g. the packet loss measured in the STB) would require:

```
.Components.FrontEnd.3.IP.FEC.Enable = “False”
```

The same applies to the RTPAVPF object.

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
  ```

 © The Broadband Forum. All rights reserved

- **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 Error Control - RTP AVPF:** In case of poor streaming performance on the IP front end the technician can enable error control features on the STB. For example, he can decide to force the STB to send RTPAVPF (AV Profile) feedback reports to the RTP Server. This is done by enabling the RTPAVPF object and then configuring RTPAVPF parameter OperationMode to "Force". Assuming IP FrontEnd is number 3 then configuration of the RTPAVPF object would be as follows:

  ```
  .Components.FrontEnd.3.IP.RTPAVPF.Enable = True
  .Components.FrontEnd.3.IP.RTPAVPF.OperationMode = Force
  ```

  The technician can then decide to enable the STB to determine autonomously whether to send RTPAVPF reports or not. There is no need to re-enable the RTPAVPF object, since this was just done. RTPAVPF parameter OperationMode is set to "Auto":

  ```
  Components.FrontEnd.3.IP.RTPAVPF.OperationMode = Auto
  ```

- **IP Error Control - AL-FEC:** In the same way as with RTP AVPF, the technician can force the STB to always use the Application Layer FEC.

  ```
  .Components.FrontEnd.3.IP.FEC.Enable = True
  .Components.FrontEnd.3.IP.FEC.OperationMode = Force
  ```

 © The Broadband Forum. All rights reserved

The technician can then decide to enable the STB to determine autonomously whether to use AL-FEC or not. There is no need to re-enable the FEC object, since was just done. FEC parameter OperationMode is set to "Auto":

```
Components.FrontEnd.3.IP.FEC.OperationMode = Auto
```

- **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"
```

The STB connects to the specified URI immediately after the ACS configured the URI Parameter. To stop the connection the URI parameter is set to an empty string.

```
.Components.FrontEnd.1.IP.ServiceConnect.URI = ""
```

- **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

 © The Broadband Forum. All rights reserved

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

 © The Broadband Forum. All rights reserved

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 ECPerfMon, VideoPerfMon and AudioPerfMon profiles include a wider range of statistics.

As described in I.3, performance management is done by the Service Monitoring object by accumulating statistical data over time. To allow the STB to accumulate statistics whenever requested by the ACS, regardless of the STB receiving content from the network or not, both "in service" and "out of service" modes of statistics collection need to be implemented.

The "out of service" operation of the STB is achieved by the force monitoring mode. When the STB is set to force monitoring mode, it switches to a pre-determined service whenever the STB is likely to stop receiving data from the network (for instance, because the user is playing video from his PVR, or because the STB is going in standby mode). Therefore, a STB in force monitoring mode will continue to collect statistics even when the STB stops receiving content from the network.

Assuming IP FrontEnd is number 3, activating the force monitoring mode would require:

```
.Components.FrontEnd.3.IP.ForceMonitor.ForeceMonitorEnable = True
.Components.FrontEnd.3.IP.ForceMonitor.ForceMonitorURI =
     rtp://224.112.15.18:3257?StrCtrl=IGMP?MuxType=MPEG2-TS
```

Reading `.Components.FrontEnd.3.IP.ForceMonitor.OperationMode` parameter allows determining whether the force monitoring mode is on.

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.

 © The Broadband Forum. All rights reserved

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 and control flows that are not relevant here and have been omitted).

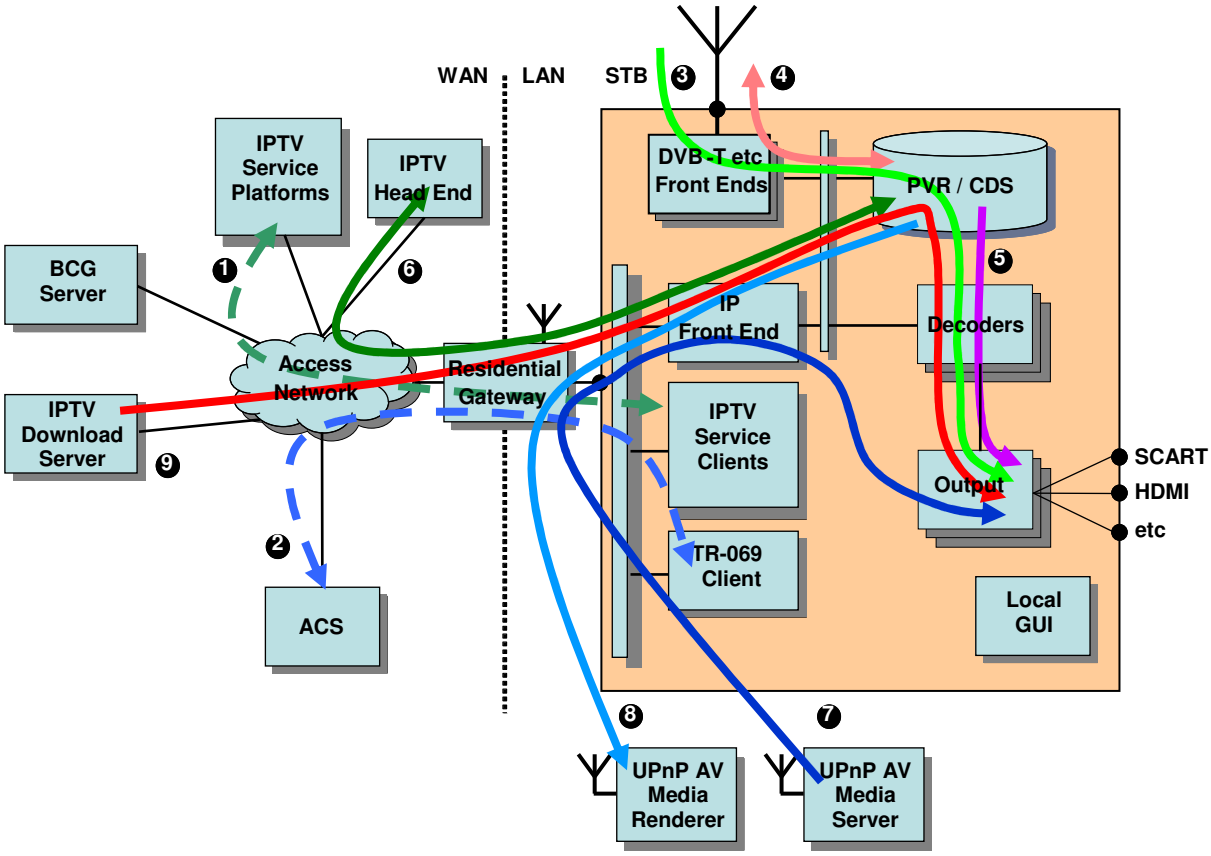


Figure 7 – STB Context (same as Figure 1)

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"
```

 © The Broadband Forum. All rights reserved

```
        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- S/PDIF”
    .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:

 © The Broadband Forum. All rights reserved

```
.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
```

 © The Broadband Forum. All rights reserved

```
.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
```

 © The Broadband Forum. All rights reserved

```
        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.

 © The Broadband Forum. All rights reserved

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
```

 © The Broadband Forum. All rights reserved

```
        AudioDecoder = <Empty>
        VideoDecoder = <Empty>

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

I.8.7 Flow 9 IPTV Download Server to CDS and Display

The user might choose to request download of content "Star Wars" from a BBC CDS server. Since download takes place through the Internet, IP Front End (#1) is going to be used,

The user intends to store this piece of content on his external storage system and as well watch the BBC piece of content being downloaded on digital audio / video devices.

There is only a single IP Front End, and it can handle multiple Inbound and Outbound streams. Since content download is requested by the user, it is managed by CDSPull object. Assuming CDSPull service Inbound object is #3, the Inbound object data model could be:

```
    .Components.FrontEnd.1.Inbound.3.
        Status = "Enabled"
        StreamingTransportProtocol = "HTTP"
        StreamingTransportControlProtocol = <Empty>
        MultiplexType = "MPEG2-TS"
        URI = "urn:bbc:starwars"
```

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.

Assuming the external storage system is modeled as Device.StorageService.1 and the CDSPull content is stored in logical volume 1 then the CDSPull object could be modeled this way:

```
    .Applications.CDSPull.Reference =
        Device.StorageService.LogicalVolume.1
```

Assuming the user intends to resort to ACS management services and this is the first content item downloaded, the CDSPull object would look like:

```
    .Applications.CDSPull.ItemNumberOfEntries
    .Applications.CDSPull.ContentItem.1.
        ContentReferenceId = "1234"
```

 © The Broadband Forum. All rights reserved

```
        VersionNumber = 1.0
        DeleteItem = False
```

The specification for ContentReferenceId is out of scope for this Technical Report

For display, assuming that AV Stream #1 is used for content download, this might result in the following parameter configurations:

```
.AVStreams.AVStream.1.
        Status = "Enabled"
        Name = "Download1"
        PVRState = "Disabled"
        FrontEnd = .Components.FrontEnd.1
        Inbound = .Components.FrontEnd.1.Inbound.3
        Outbound = <Empty>
        AudioDecoder = .Components.AudioDecoder.1
        VideoDecoder = .Components.AudioDecoder.1
```

Note that the PVR object is disabled since this content download is not managed by PVR but by CDS.

The AVStream.{i}. object contains no explicit indication that any CDS system is being used.

Finally, the data model indicates that AVStream #1 is mapped onto AVPlayer # 1:

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

I.8.8 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 does not involve any change to the AV Streams, just to the AV Player:

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

I.9 Content Download Service

The Content Download Service allows the operator to download content to the end user's STB. A possible system architecture to perform this task was recently standardized by DVB [18] and

 © The Broadband Forum. All rights reserved

named the CDS System. The CDS data model of this document may apply to the DVB CDS system, but is not limited to it.

The Content Download Service comes in two operation modes: Push and Pull. In Push Mode the operator autonomously determines which content items should be downloaded to the end user's STB and initiates the process. In Pull Mode, instead, it is the end user that initiates the process by requesting the operator to download a specific content item. The download is then carried out by the operator as in Push Mode.

These operation modes are managed by means of TR-069 objects CDSPush and CDSPull. These can be implemented independently of each other. They allow the operator's ACS to manage the (portion of) end user storage used by the relevant service. Management is the same in both cases: there is no configuration that the ACS needs to do on these objects but, as a support to troubleshooting, the data model objects allow the ACS to mark content items for deletion by the STB.

The only difference CDSPush and CDSPull operation is that for the sake of privacy in Pull Mode the user might be allowed to decide whether to hide his own content list from the ACS, and manage it locally, or to expose it and resort to the ACS management services. This option is mainly an implementation issue and is out of scope for TR-069 management.

The STB indicates to the ACS which kind of CDS service it supports by setting the relevant .Capabilities.CDS. parameter. A STB that supports both CDSPush and CDSPull would indicate

```
.Capabilities.CDS.PushCapable = True
.Capabilities.CDS.PullCapable = True
```

At boot time the STB populates the CDSPush / CDSPull content item lists. To do this the STB needs to connect to the user's hard disks. The pathnames of these are exposed in the data model. If, for instance, the user's Storage service is named StorageService.1 and, on this system, pushed pieces of content are stored on logical volume 1 and pulled pieces of content are stored on logical volume 2 then the data model parameters would be:

```
.Applications.CDSPush.Reference =
     Device.Services.StorageService.1.LogicalVolume.1
.Applications.CDSPull.Reference =
     Device.Services.StorageService.1.LogicalVolume.2
```

After connecting, the STB determines which content items are stored on which disk and populates the corresponding object tables. If, among the pushed pieces of content, item number 1 for instance models a stored content item that has ContentReferenceID = "1234" and Version Number = 1.0 then the data model would look like this:

```
.Applications.CDSPush.ContentItem.1.ContentReferenceID =
      "1234"
.Applications.CDSPush.ContentItem.1.VersionNumber = 1.0
```

 © The Broadband Forum. All rights reserved

```
.Applications.CDSPush.ContentItem.1.DeleteItem = False
```

ContentReferenceID specification is out of scope for this Technical Report.

The STB sets the ContentItem instance DeleteItem flag to False to indicate the corresponding content item is present on the disk. It sets the DeleteItem flag to true to indicate that there is no content item on the disk for that ContentItem instance.

The ACS monitors the exposed lists of content items created / deleted. In case one or more specific pieces of content cause trouble, for instance because they cannot be played for some reason, the ACS requests deletion of these content items by configuring the DeleteItem flag in the ContentItem object to True. To request deletion of ContentItem number 5, for instance:

```
.Components.CDSPush.ContentItem.5.DeleteItem = True
```

The ACS also reads from the Data Model the pathnames of the storage systems where the downloaded content items reside.

End of Broadband Forum Technical Report TR-135

 © The Broadband Forum. All rights reserved