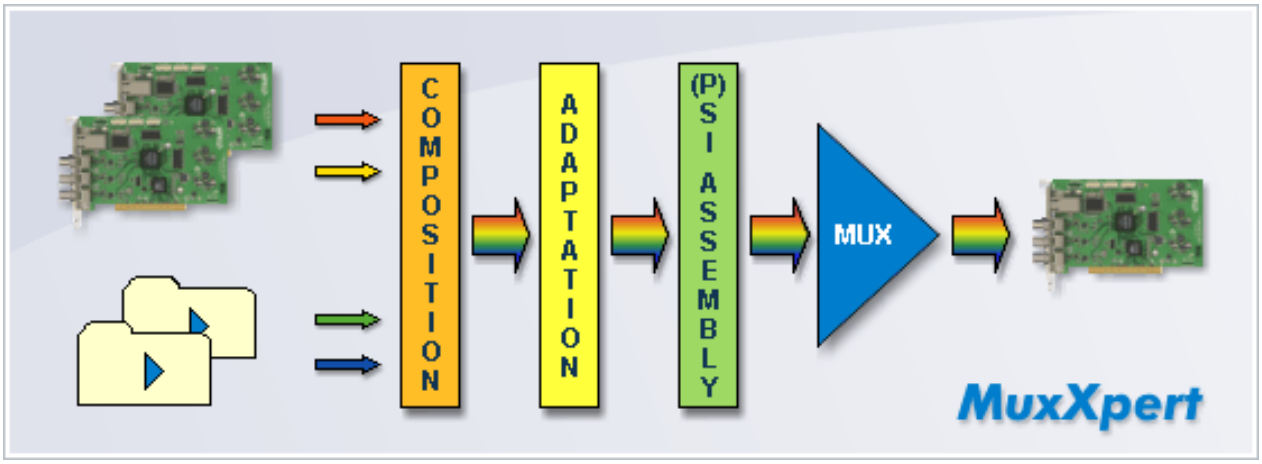
XML Template for RMC Data

FEATURES

- XML-based template for specifying the Remultiplexer Configuration Data.
- Defines the data interface to DekTec's MuxXpert Transport Stream Re-Multiplexer Software.



RELEVANT PRODUCTS

Type	Description
DTC-700-MX	*MuxXpert* MPEG-2 Transport-Stream Re-Multiplexer Software
DTC-705-MR	*MuxXpert API* .NET Class Library for Real-Time Multiplexing

Table of Contents

RMC Data Revision History

Version	Date	Change Description
V2.11.0.2844	2013.10.15	• Added support for DTA-2162 • Increased tolerance for network-jitter on local NIC
V2.10.0.2644	2013.03.29	• Added support for DTA-2138 (DVB-C and DVB-T) • Added support for DTA-2139 • Added support for DVB-S2 roll-off • Fixed DVB-C2/T2 modulation performance • Fixed support for DTA-100 • Fixed bitrate accuracy problem for DTA-2107 • Improved support for large configurations • Improved support for multiple NICs (network interface cards)
V2.9.2.2468	2012.10.03	• Fixed support for DTA-2135 • Fixed support for DTA-2137
V2.9.0.2388	2012.07.16	• Added support for DTA-2107, DTA-2136 and DTE-3137 • Fixed start-up failure of configurations using many IP-outputs • Fixed incorrect license expiration
V2.8.0.2320	2012.05.14	• Based on new DTAPI and drivers
V2.7.0.2244	2012.02.03	• Improved start-up behaviour • Added option to temporary run the MuxXpert API with GUI
V2.6.6.2167	2011.12.08	• Improved robustness for IP-input streams • Added DTA-112 ISDB-T support • Improved indication in case (P)SI is dropped. • Fixed private data specifier > 0x7FFFFFFF bug
V2.6.0.1851	2011.01.26	• Support for single PLP DVB-C2 • Support for single TS, single layer ISDB-S
V2.5.1.1828	2011.01.03	• Fixed DTA-110 support
V2.5.0.1797	2010.12.03	• Support for DTA-2111 • Support for 192 byte TS-files (M2TS-files) • Support for variable rate single program TS-files • Added play-list looping • Fixed large configuration restrictions
V2.4.0.1642	2010.07.02	• Increased number of TS-inputs, TS-outputs and File-players for MuxXpert-API • NIC-only configurations fixed. • Added option to skip extraction and processing of selected SI tables from the TS-input • Added option to share PMT PIDs • Swe.exe memory leak fixed.
V2.3.1.1615	2010.06.04	• Fixed DTA-2160 detection • Fixed installation conflict with StreamXpert
V2.3.0.1563	2010.04.13	• Improved robustness for SDT-actual from SDT-other • Support for DTA-111 and DTA-2160
V2.2.0.1489	2010.01.29	• Support for DTE-3100, DTE-3120

		• Support for Local Network Adapter for TS-over-IP input and output ports. • Support for DVB-T2 output ports.
V2.1.1.1377	2009.10.09	• Fixes output stream interruption after setting same output parameters • Improved robustness for invalid EIT-schedule
V2.1.0.1363	2009.09.25	• Support for DTA-2135 and DTA-2137 • Increased number of outputs for MuxXpert-API • Fixes SDT-other from SDT-actual-out updates • Service_ID 0xFFFF is accepted • Memory leak fixed • Improved robustness in case of invalid PID adaptations
V2.0.2.1179	2009.03.25	• Increased number of TS-input and File-player for MuxXpert-API • Improved robustness for unreliable input Transport Streams • Improved TS-file bitrate estimation for files that are added in play-list
V2.0.1.1139	2009.02.16	• Changed ATSC filter to 32 taps. • Added thumbnails for new supported cards. • Sets variable rate SPTS support from configuration. • Improved MuxXpert restart • Added table version attribute
V2.0.1.1139	2009.02.16	• Changed ATSC filter to 32 taps. • Added thumbnails for new supported cards. • Improved MuxXpert restart • Added table version attribute

V2.0.0.1068	2008.12.02	• MuxXpert-API support included.
		• Support for variable rate single program transport streams over IP.
		• Support for DTA-112,DTA-116, DTA-117 and DTA-2144
		• Fixes problem if TEI is set in TS-packets.
		• Fixes playlist bugs:
		  ○ Failure when no drive-letter is specified.
		  ○ Failure when some but not all items contain a start-time.
		• Check for configuration + play-list file update interval is set to 1 second.
		• Confirmation is asked when closing the MuxXpert.

V1.5.1.757	2008.01.28	• Support for ADTB-T and DMB-T/H output ports.
		• Support for ATSC, QAM-B and DVB-S.2 is improved.
		• Improved support for managing EIT present and EIT following events.
		• Improved Incoming NIT-actual adaptation.
		• Fixes the unreferenced PID selection exception.
		• Fixes the ECM-adaptation exception.
		• Improved robustness in case the IP-cable is not connected.
		• RMC Data template changes:
		  ○ ADTBT and DMBTH replace the DTMB output port.
		  ○ QAMB output parameters are redefined.
		  ○ Redundant OfdmModulationType attribute is removed from DVBH and DVBT output parameters.
		  ○ Non-standard DVBC and QAMC modulation types are removed.
		  ○ DVBS attributes are renamed (old names can still be used).
		  ○ NewPcr element replaces NewPcrPid for changing the PCR of a service (NewPcrPid can still be used).
		  ○ The order of the tables in DvbSiTables is according their table_id.
		  ○ Attribute FilterPfEvents is added to the element EitTimeFilter to control the filtering of EIT present and EIT following events. The FilterPfEvents attribute replaces the PassPastEvents attribute.

		○ Attribute EventType is added to the EventPars element to set the type of an EIT-event.
V1.5.0.620	2007.09.13	• Support for Transport-Stream recording • Support for Custom sub-table generation • Support for disabling of EIT-time-filter for EIT-P/F tables • TS bitrate estimation is done when file is added in play-list • Right mouse click to remove services from output service-list • Maximum play-list size increased to 100 items • Scrollbars are added in behalf of low resolution screens • If necessary ECM(s) are added in case of SvcWideComposition • More double-number-styles are allowed • Fixes the KeepOrigPid and SvcFromScratch combination exception • Fixes the EitActualFromComposition sub-table generation exception • Improved robustness in case an invalid descriptor is specified
V1.4.0.549	2007.07.04	• Support for ATSC, DTMB and DVBS2 output port type. • Support for the DTA107S2 adapter. • Support for the ISDB-T mode 1 segment TV. • Support for selection of a PID range in TsWideComposition. • Support for PcrAdaptation for specifying a new PID for a PCR stream. • Support for creating a NIT-actual from an incoming NIT-actual. • Support for Custom Table generation • Comment lines in SdtActualAdaptation are accepted. • EitOtherFromScratch according to XSD • The number of messages shown in the GUI is limited.
V1.3.2.520	2007.06.05	• Support for 1 segment ISDB-T • Character Code Table selection added in GUI • IP-multicast zero source port is changed • DVB-T Setting fixed • Reading SectionData from RmcData file fixed
V1.3.1.467	2007.04.14	• DVB-T/H configuration start-up error fixed • GUI displays the ISDBT layer of the generated service components
V1.3.0.456	2007.03.30	• Support for ISDBT output port type • Support for QAMB and QAMC output port type • Support for TsTimeOffset for specifying a time offset of an input and output transport stream. • Support for TsTimeCorrection for specifying the applied time correction. • Support for ISDBT transport stream output parameters settings. • Support for KeepOrigPid attribute to keep incoming PIDs unchanged if possible. • Support for simple drag and drop of services. • Maximum number of file players increased to 4. • QamRollOff attribute is removed now DVBC, QAMB and QAMC are different port types • Specification of the Tag element in table descriptor sorting is changed • Specification of the Duration attribute in event is changed.
V1.2.1.331	2006.11.16	• Optional OnwId attribute is added to TsOut
V1.0.285	2006.08.16	• Created

1. Introduction

1.1 Purpose of this Document

This document defines the *Remux Configuration Data* (*RMC-Data*) interface. This interface is used to contribute DVB/MPEG-2 data, including (P)SI tables, custom tables, descriptors and table-assembly parameters, to the MuxXpert.

1.2 Intended Audience

This document is intended to be used by users of the MuxXpert who want to create RMC-Data, e.g. data describing the composition of the outgoing transport streams.

1.3 Interface Specification

The RMC-Data interface is a *file-based interface*. The permissible format of the XML file is defined by XML *Schemas*, written in XML Schema Definition Language [XSDL].

The RMC-Data Schemas are defined in the RmcDataTempl00.xsd file.

The .xsd file enables:

- Conveniently viewing the RMC-Data schemas, using a XML authoring tool;
- Conveniently editing (by hand) of RMC-Data files, using a schema-driven XML editor;
- Automatic validation of the structure of RMC-Data files using a XSDL-enabled validating parser.

The elements and attributes in the schema are described in plain text in this document. The following conventions are used:

Element
XML elements are set in a green, bold font.

Attribute
Attributes of XML elements are set in brown.

Enumeration
Enumeration values are set in blue italics.

Type
Complex types are set in green bold italics.

AttributeType
Attribute types are set in red bold.

The elements and attributes are also shown in a graphical diagram. In this diagram, elements are represented as rectangles; relations between elements are displayed as lines connecting elements. Attributes are represented as rectangles with rounded edges; attributes contained in an element are grouped by a red line. See the example below where element A contains element B and where element B contains the attributes X and Y.

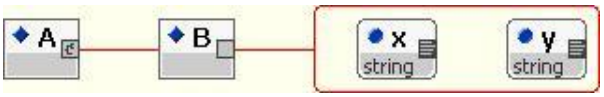


Figure 1. Example of the graphical representation of elements and attributes.

Occurrence indicators are represented graphically in the following way:

no indicator: the element must appear once and only once.
'?' indicator: the element may appear zero or one times.
'+' indicator: the element must appear one or more times.
'*' indicator: the element may appear zero, one or more times.

See the example below.

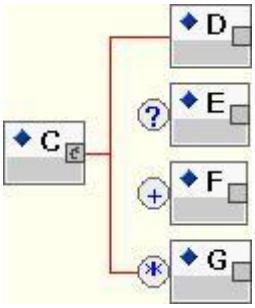


Figure 2. Example of the graphical representation of occurrence indicators.

Sequence and choice indicators are represented graphically in the following way:

Sharp-edged lines: indicate a sequence. All listed elements have to appear in the order indicated. In the example below, element H must contain element I and J and in that order.
Sloping lines: indicate a choice. One and only one of the listed elements has to appear.

In the example below, element J must contain element K or L but not both.

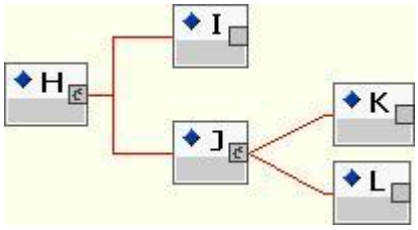


Figure 3. Example of the graphical representation of a sequence and a choice.

1.4 Document Overview

Chapter 2 provides the "context" of the RMC-Data interface within a remultiplexing system. An overview of the interface functions supported by the RMC-Data interface is given.

Chapter 3 describes (for humans) the semantics of elements and attributes defined in the XML Schema files.

Chapter 4 gives an RMC-Data example.

1.5 Definitions, Acronyms and Abbreviations

PDSD
private_data_specifier_descriptor.

Producer
The party that produces RMC-Data. Potential Producers include the system integrator who produces static RMC-Data, or an application that dynamically generates RMC-Data.

Section
Shorthand for *private section*, as defined by clause private_section() in Table 2-30 of [MPEG-2 SYS].

XML
Extensible Markup Language. A markup language defined by the W3C that provides a strict set of standards for document syntax.

XSDL
XML Schema Definition Language. A schema definition language under development by the W3C Schema working group. Expressed in XML document syntax, XSDL is designed to support an extensible data typing system, inheritance, and namespaces.

W3C
The World Wide Web Consortium, which sets Web-oriented standards like XML.

1.6 References

[DVB SI]
EN 300 468, Digital Video Broadcasting (DVB); Specification for Service Information (SI) in DVB systems, V1.6.1 (2004-06).

[MPEG-2 SYS]
ISO/IEC 13818-1, Information technology – Generic coding of moving pictures and associated audio information: Systems, May 27th, 1999.

[XSDL]
XML Schema Definition Language, W3C, Candidate Recommendation, October 24, 2000

2. Context

The objective of the interface is to contribute DVB/MPEG-2 data to a RMC-Data *Consumer*, which is the MuxXpert.

Multiple RMC-Data interfaces may be used in parallel, each with its own directory and its own *Producer*. Typically, one Producer contributes data over one RMC-Data interface. A Producer may use multiple RMC-Data interfaces to contribute its data, e.g. if the data can be separated in a number of logically independent segments. On the other hand, a single RMC-Data interface *cannot* be used by multiple Producers.

In the remultiplexer system, the following types of Producers may occur:

The system integrator, who uses a support tool to produce static RMC-Data, e.g. data describing the composition of the outgoing transport stream.

A (third-party) application that wants to contribute certain (P)SI tables or descriptors.

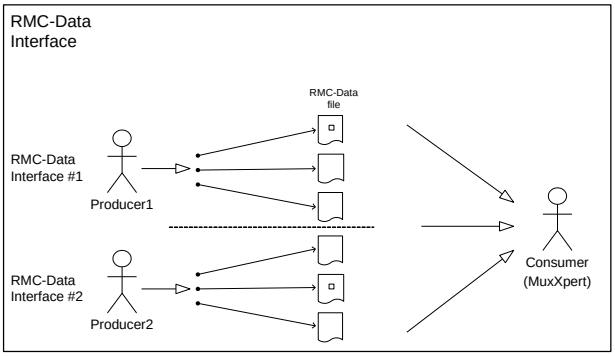


Figure 4. Usage of the RMC-Data Interface.

Note:

The MuxXpert supports now only a single RMC-Data interface. This means that all RMC-Data must reside in one file.

3. Semantics RMC-Data

3.1 RMC-Data – Top-Level

The top-level element of each RMC-Data file is RemuxConfig element.

The syntax of this element is defined entirely by the RmcDataTemplate00.xsd schema.

RemuxConfig

Root-element of a Remux Configuration Data document. The RemuxConfig contains three child elements:

1. LogTsIdIoPortMapping; Optional element that specifies the relation between the logical transport stream IDs (LogTsIds, used within the configuration) and the input and output ports of the re-multiplexer.
2. TsIns; Optional element that specifies the parameters of the transport- stream inputs.
3. TsOuts; Optional element that specifies settings and the contents of the outgoing transport-streams.

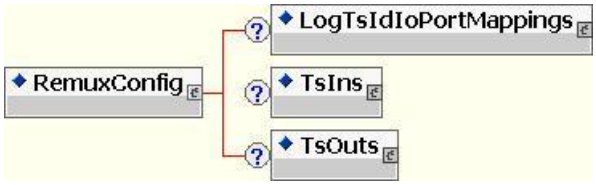


Figure 5. Graphical diagram of RemuxConfig.

3.2 Logical Transport Stream ID to IO Port Mapping

Logical Transport Stream IDs (LogTsIds) are used to uniquely identify a transport stream input or output. Within the RMC-data these LogTsIds are used to refer to an input or output, without having to know the physical location of the transport streams.
The LogTsIdIoPortMappings element is the place where the relation between LogTsIds and the physical input and output streams is specified.

Note:

If the RMC-Data refers to an unspecified LogTsId this is the same as referring to an empty transport-stream.

LogTsIdIoPortMappings

Optional element, used for specifying the relation between LogTsIds and the input and output ports of the re-multiplexing system. The LogTsIdIoPortMappings element contains a sequence of LogTsIdIoPortMapping elements.

Below the graphical diagram of the LogTsIdIoPortMappings element is shown.

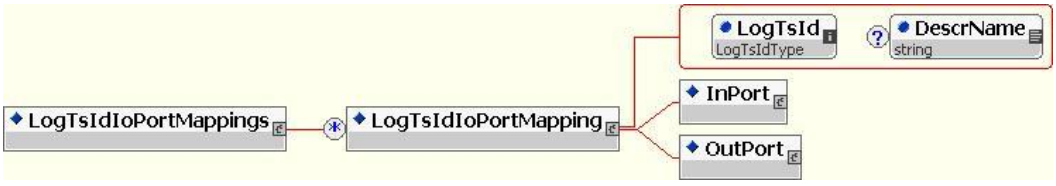


Figure 6. Graphical diagram of LogTsIdIoPortMappings.

LogTsIdIoPortMappings/LogTsIdIoPortMapping

Specifies the relation between one LogTsId and the input or output of the re-multiplexer application. One of the following child elements may occur: OutPort or InPort.

LogTsId

Type: LogTsIdType, Use: required
Uniquely identifies a transport-stream.

DescrName

Type: string, Use: optional
Descriptive name of the transport stream.

3.2.1 Input Port

InPort

Element; to specify an input port. One of the following child elements may occur: ASI, DVBS, DVBS2, DVBT, IP, File or SPI.

Below the graphical diagram of the InPort element is shown. Only the upper branch is expanded.

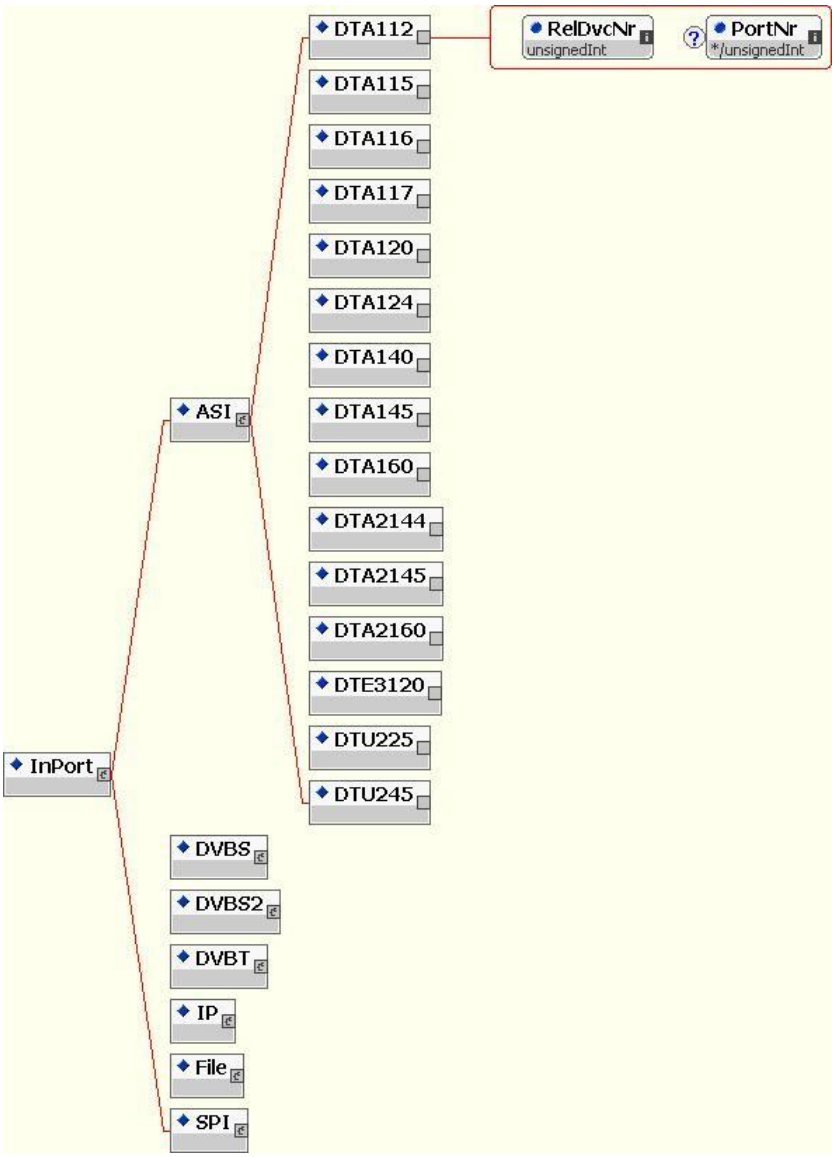


Figure 7. Graphical diagram of Inport.

InPort/ASI

Element; to specify an ASI input port.
In case of an input port for the PCI, PCI-Express or USB-bus, it contains one of the following child elements to specify the adapter type: DTA112, DTA115, DTA116, DTA117, DTA120, DTA124, DTA140, DTA145, DTA160, DTA2144, DTA2145, DTA2160, DTU225 or DTU245. These elements contain the following attributes to specify the adapter in the system and the port:

RelDevcNr
Type: unsignedInt, Use: required.
Specifies the relative position of the adapter in the system.

PortNr
Type: unsignedInt, Use: required if the adapter supports multiple input ports, otherwise optional.
Specifies the port number used.

In case of a Networked ASI input port; it contains the following child elements to specify the adapter type: DTE3120.
This element contains the following attributes to specify the adapter in the system and the port:

IpAddress
Type: IpAddressType, Use: required
Specifies the IP address of the adapter in the system. The format of the IP address must consist of 4 IP-address parts separated by a dot (e.g. "192.168.39.2").

PortNr
Type: unsignedInt, Use: required if the adapter supports multiple input ports, otherwise optional.
Specifies the port number used.

InPort/DVBS
Element; to specify a DVB-S receiver input port; it contains the following child element to specify the adapter type: DTA2137
This element contains the following attributes to specify the adapter in the system and the port:

RelDevcNr
Type: unsignedInt, Use: required.
Specifies the relative position of the adapter in the system.

PortNr
Type: unsignedInt, Use: optional.
Specifies the port number used.

InPort/DVBS2
Element; to specify a DVB-S2 receiver input port; it contains the following child element to specify the adapter type: DTA2137
This element contains the following attributes to specify the adapter in the system and the port:

RelDevcNr
Type: unsignedInt, Use: required.
Specifies the relative position of the adapter in the system.

PortNr
Type: unsignedInt, Use: optional.
Specifies the port number used.

InPort/DVBT
Element; to specify a DVB-T receiver input port; it contains the following child element to specify the adapter type: DTA2135 or DTA2138
This element contains the following attributes to specify the adapter in the system and the port:

RelDevcNr
Type: unsignedInt, Use: required.
Specifies the relative position of the adapter in the system.

PortNr
Type: unsignedInt, Use: optional.
Specifies the port number used.

InPort/IP

Element; to specify an IP input port. In case a DekTec IP-Adapter is used, it contains the following child element to specify the adapter type: DTA160 or DTA2160.
This element contains the following attributes to specify the adapter in the system and the port:

RelDevcNr
Type: unsignedInt, Use: required.
Specifies the relative position of the adapter in the system.

PortNr
Type: unsignedInt, Use: optional.
Specifies the port number used.

In case of the local network adapter, it contains the following child elements to specify the adapter type: NIC.
This element contains the following attributes to specify the network adapter.

IpAddress
Type: IpAddressType, Use: optional
Specifies the IP address of the adapter in the system. The address is only relevant in case of multicast-IP-reception, to specify from which adapter to receive.
The format of the IP address must consist of 4 IP-address parts separated by a dot (e.g. "192.168.39.2").

InPort/File

Element; to specify a file player as input port; it contains the following child element to specify the file player: TsFilePlayer
This element contains the following attributes to specify the player used.

PlayerNr
Type: unsignedInt, Use: required.
Specifies the file player used (e.g. 1 for the first player, 2 for the second player).

InPort/SPI

Element; to specify an SPI input port; it contains the following child element to specify the adapter type: DTA122
This element contains the following attributes to specify the adapter in the system and the port:

RelDevcNr
Type: unsignedInt, Use: required.
Specifies the relative position of the adapter in the system.

PortNr
Type: unsignedInt, Use: optional.
Specifies the port number used.

3.2.2 Output Port

OutPort

Element; to specify an output port. One of the following child elements may occur: ASI, ATSC, DTMB, DVBC, DVBC2, DVBH, DVBS, DVBS2, DVBT, DVBT2, IP, ISDBT, QAMB, QAMC or SPI.

Below the graphical diagram of the OutPort element is shown. . Only the upper branch is expanded.

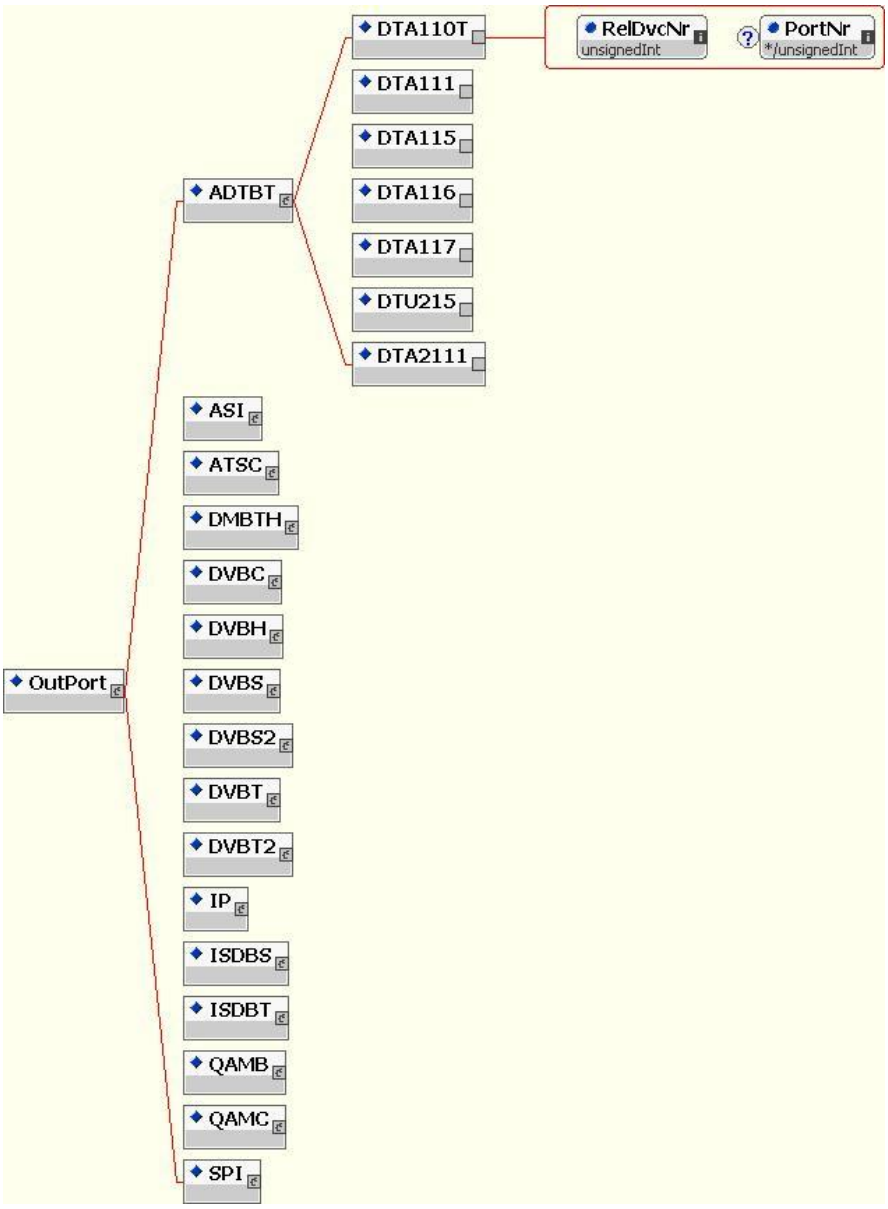


Figure 8. Graphical diagram of Outport.

OutPort/ADTBT

Element; to specify an ADTB-T output port; it contains one of the following child elements to specify the adapter type: DTA110T, DTA111, DTA115, DTA116, DTA117, DTU215 or DTA2111. These elements contain the following attributes to specify the adapter in the system and the port:

RelDevcNr
Type: unsignedInt, Use: required.
Specifies the relative position of adapter in the system.

PortNr
Type: unsignedInt, Use: optional.
Specifies the port number used.

OutPort/ASI

Element; to specify an ASI output port.
In case of an output port for the PCI, PCI-Express or USB-bus, it contains one of the following child elements to specify the adapter type: DTA100, DTA105, DTA112, DTA115, DTA116, DTA117, DTA140, DTA145, DTA160, DTA2137, DTA2144, DTA2145, DTA2160, DTU205 or DTU245.
These elements contain the following attributes to specify the adapter in the system and the port:

RelDevcNr
Type: unsignedInt, Use: required.
Specifies the relative position of the adapter in the system.

PortNr
Type: unsignedInt, Use: required if the adapter supports multiple output ports, otherwise optional.
Specifies the port number used.

In case of a Networked ASI output port; it contains the following child elements to specify the adapter type: DTE3100.
This element contains the following attributes to specify the adapter in the system and the port:

IpAddress
Type: IpAddressType, Use: required
Specifies the IP address of the adapter in the system. The format of the IP address must consist of 4 IP-address parts separated by a dot (e.g. "192.168.39.2").

PortNr
Type: unsignedInt, Use: required if the adapter supports multiple output ports, otherwise optional.
Specifies the port number used.

OutPort/ATSC

Element; to specify an ATSC output port; it contains one of the following child elements to specify the adapter type: DTA110T, DTA111, DTA112, DTA115, DTA116, DTA117, DTU215 or DTA2111.
These elements contain the following attributes to specify the adapter in the system and the port:

RelDevcNr
Type: unsignedInt, Use: required.
Specifies the relative position of adapter in the system.

PortNr
Type: unsignedInt, Use: optional.
Specifies the port number used.

OutPort/DMBTH

Element; to specify a DMB-T/H output port; it contains one of the following child elements to specify the adapter type: DTA110T, DTA111, DTA115, DTA116, DTA117, DTU215 or DTA2111. These elements contain the following attributes to specify the adapter in the system and the port:

RelDevcNr

Type: unsignedInt, Use: required.
Specifies the relative position of adapter in the system.

PortNr

Type: unsignedInt, Use: optional.
Specifies the port number used.

OutPort/DVBC

Element; to specify a DVB-C output port; it contains one of the following child elements to specify the adapter type: DTA110, DTA110T, DTA111, DTA112, DTA115, DTA116, DTA117, DTU215 or DTA2111.
These elements contain the following attributes to specify the adapter in the system and the port:

RelDevcNr

Type: unsignedInt, Use: required.
Specifies the relative position of adapter in the system.

PortNr

Type: unsignedInt, Use: optional.
Specifies the port number used.

OutPort/DVBC2

Element; to specify a DVB-C2 output port; it contains the following child element to specify the adapter type: DTA111, DTA115, DTA116, DTA117, DTU215 or DTA2111.
This element contains the following attributes to specify the adapter in the system and the port:

RelDevcNr

Type: unsignedInt, Use: required.
Specifies the relative position of the adapter in the system.

PortNr

Type: unsignedInt, Use: optional.
Specifies the port number used.

OutPort/DVBH

Element; to specify a DVB-H output port; it contains the following child element to specify the adapter type: DTA110T, DTA111, DTA112, DTA115, DTA116, DTA117, DTU215 or DTA2111.
This element contains the following attributes to specify the adapter in the system and the port:

RelDevcNr

Type: unsignedInt, Use: required.
Specifies the relative position of the adapter in the system.

PortNr

Type: unsignedInt, Use: optional.
Specifies the port number used.

OutPort/DVBS

Element; to specify a DVB-S output port; it contains the following child element to specify the adapter type: DTA107, DTA107S2 or DTA2107.
This element contains the following attributes to specify the adapter in the system and the port:

RelDevcNr

Type: unsignedInt, Use: required.
Specifies the relative position of the adapter in the system.

PortNr

Type: unsignedInt, Use: optional.
Specifies the port number used.

OutPort/DVBS2

Element; to specify a DVB-S2 output port; it contains the following child element to specify the adapter type: DTA107S2 or DTA2107.
This element contains the following attributes to specify the adapter in the system and the port:

RelDevcNr

Type: unsignedInt, Use: required.
Specifies the relative position of the adapter in the system.

PortNr

Type: unsignedInt, Use: optional.
Specifies the port number used.

OutPort/DVBT

Element; to specify a DVB-T output port; it contains the following child element to specify the adapter type: DTA110T, DTA111, DTA112, DTA115, DTA116, DTA117, DTU215 or DTA2111.
This element contains the following attributes to specify the adapter in the system and the port:

RelDevcNr

Type: unsignedInt, Use: required.
Specifies the relative position of the adapter in the system.

PortNr

Type: unsignedInt, Use: optional.
Specifies the port number used.

OutPort/DVBT2

Element; to specify a DVB-T2 output port; it contains the following child element to specify the adapter type: DTA111, DTA115, DTA116, DTA117, DTU215 or DTA2111.
This element contains the following attributes to specify the adapter in the system and the port:

RelDevcNr

Type: unsignedInt, Use: required.
Specifies the relative position of the adapter in the system.

PortNr

Type: unsignedInt, Use: optional.
Specifies the port number used.

OutPort/IP

Element; to specify an IP output port. In case a DekTec IP-Adapter is used, it contains the following child element to specify the adapter type: DTA160 or DTA2160.
This element contains the following attributes to specify the adapter in the system and the port:

RelDevcNr

Type: unsignedInt, Use: required.
Specifies the relative position of the adapter in the system.

PortNr

Type: unsignedInt, Use: optional.
Specifies the port number used.

In case of the local network adapter, it contains the following child elements to specify the adapter type: NIC.
This element contains the following attributes to specify the network adapter.

IpAddress

Type: IpAddressType, Use: optional
Specifies the IP address of the adapter in the system. The address is only relevant in case of multicast-IP-transmission, to specify which adapter is used.
The format of the IP address must consist of 4 IP-address parts separated by a dot (e.g. "192.168.39.2").

OutPort/ISDBS

Element; to specify an ISDB-S output port; it contains the following child element to specify the adapter type: DTA107, DTA107S2 or DTA2107.
This element contains the following attributes to specify the adapter in the system and the port:

RelDevcNr

Type: unsignedInt, Use: required.
Specifies the relative position of the adapter in the system.

PortNr

Type: unsignedInt, Use: optional.
Specifies the port number used.

OutPort/ISDBT

Element; to specify an ISDB-T output port; it contains the following child element to specify the adapter type: DTA110T, DTA111, DTA115, DTA11 DTA116, DTA117, DTU215 or DTA2111.
This element contains the following attributes to specify the adapter in the system and the port:

RelDevcNr

Type: unsignedInt, Use: required.
Specifies the relative position of the adapter in the system.

PortNr

Type: unsignedInt, Use: optional.
Specifies the port number used.

OutPort/QAMB

Element; to specify a QAM-B output port; it contains one of the following child elements to specify the adapter type: DTA110, DTA110T, DTA111, DTA112, DTA115, DTA116, DTA117, DTU215 or DTA2111.
These elements contain the following attributes to specify the adapter in the system and the port:

RelDevcNr

Type: unsignedInt, Use: required.
Specifies the relative position of adapter in the system.

PortNr

Type: unsignedInt, Use: optional.
Specifies the port number used.

OutPort/QAMC

Element; to specify a QAM-C output port; it contains one of the following child elements to specify the adapter type: DTA110, DTA110T, DTA111, DTA112, DTA115, DTA116, DTA117, DTU215 or DTA2111.
These elements contain the following attributes to specify the adapter in the system and the port:

RelDevcNr

Type: unsignedInt, Use: required.
Specifies the relative position of adapter in the system.

PortNr

Type: unsignedInt, Use: optional.
Specifies the port number used.

OutPort/SPI

Element; to specify an SPI output port; it contains the following child element to specify the adapter type: DTA102.
This element contains the following attributes to specify the adapter in the system and the port:

RelDevcNr

Type: unsignedInt, Use: required.
Specifies the relative position of the adapter in the system.

PortNr

Type: unsignedInt, Use: optional.
Specifies the port number used.

3.3 Transport Stream Inputs

The TsIns element is used to specify the parameter settings of the transport-stream inputs.

TsIns

Element; specifying the settings of incoming transport-stream inputs by means of a sequence of TsIn elements.

Below the graphical diagram of the TsIns element is shown.

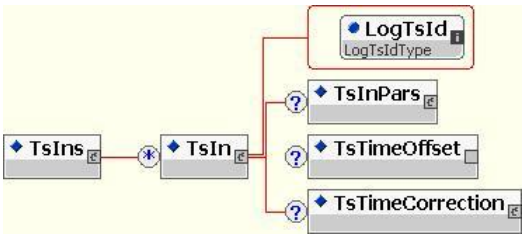


Figure 9. Graphical diagram of TsIns.

TsIns/TsIn

Specifies the settings of a single transport-stream input. The input stream is described with the following child elements: TsInPars, TsTimeOffset and TsTimeCorrection.

LogTsId

Type: LogTsIdType, Use: required
Uniquely identifies a transport-stream input.

TsIn/TsInPars

Optional element, specifying the settings for the transport-stream input port. See section 3.3.1.

TsIn/TsTimeOffset

Optional element, specifying the time offset that has to be used in case the time in the incoming tables is not based on UTC-time. See section 3.3.2.

TsIn/TsTimeCorrection

Optional element, specifying whether and how times in the incoming tables are corrected. If this element is absent no time correction is applied. See section 3.3.3.

3.3.1 TsInPars

TsInPars

Optional element. This element specifies the settings for the transport-stream input port; it contains one of the following child elements: ASI, SPI, IP or File.

Below the graphical diagram of the TsInPars element is shown.

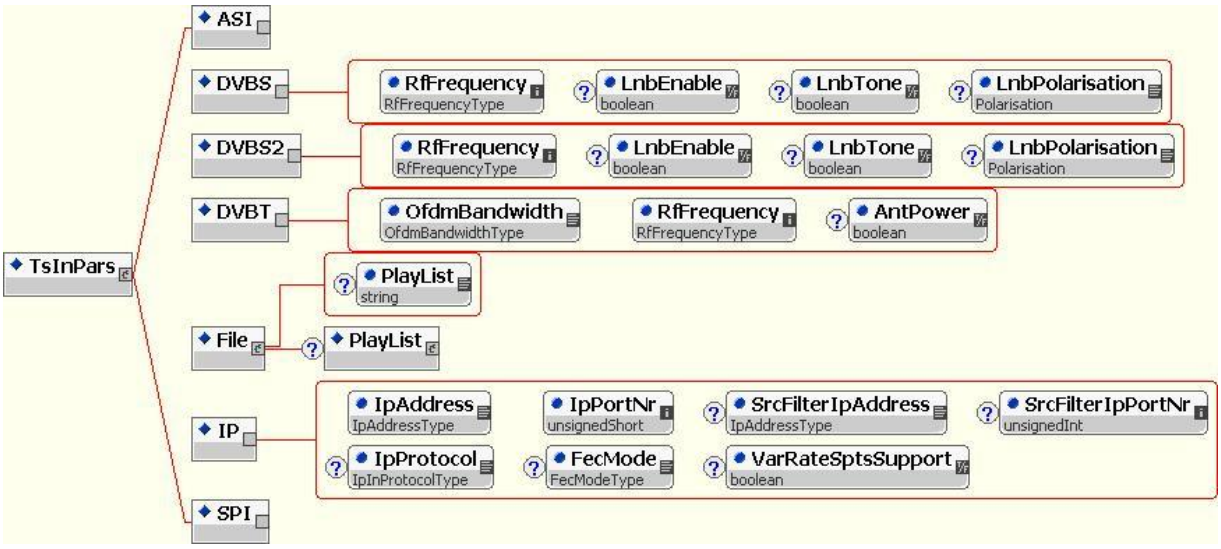


Figure 10. Graphical diagram of TsInPars.

TsInPars/ASI

Element; Empty, no input parameters need to be specified.

TsInPars/SPI

Element; Empty, no input parameters need to be specified.

TsInPars/DVBS

Element; to specify the parameters of a DVB-S receiver input port.

RfFrequency

Type: RfFrequencyType, Use: required
Specifies the tuner frequency for DVB-S receiver.

LnbEnable

Type: boolean, Use: optional (default: "*false*")
Specifies whether the LNB control is enabled.

LnbTone

Type: boolean, Use: optional (default: "*false*")
Specifies whether the LNB tone is switched on. Only applicable if LNB control is enabled.

LnbPolarisation

Type: Polarisation, Use: optional (default: "*horz*")
Specifies the LNB polarisation. This can be set to "*horz*" or "*vert*". Only applicable if LNB control is enabled.

TsInPars/DVBS2

Element; to specify the parameters of a DVB-S2 receiver input port.

RfFrequency

Type: RfFrequencyType, Use: required
Specifies the tuner frequency for DVB-S2 receiver.

LnbEnable

Type: boolean, Use: optional (default: "*false*")

Specifies whether the LNB control is enabled.

LnbTone
Type: boolean, Use: optional (default: "*false*")
Specifies whether the LNB tone is switched on. Only applicable if LNB control is enabled.

LnbPolarisation
Type: Polarisation, Use: optional (default: "*horz*")
Specifies the LNB polarisation. This can be set to "*horz*" or "*vert*". Only applicable if LNB control is enabled.

TsInPars/DVBT
Element; to specify the parameters of a DVB-T receiver input port.

OfdmBandwidth
Type: DvbhOfdmBandwidthType, Use: required
Specifies the bandwidth of the modulated signal. This can be set to: "*6MHz*", "*7MHz*" or "*8MHz*".

RfFrequency
Type: RfFrequencyType, Use: required
Specifies the tuner frequency for DVB-T receiver.

AntPower
Type: boolean, Use: optional (default: "*false*")
Specifies whether the antenna power is on.

TsInPars/IP
Element; to specify the parameters of an IP input port.

IpAddress
Type: IpAddressType, Use: required
Specifies the IP address from which to receive the packets. The format of the IP address must consist of 4 IP-address parts separated by a dot (e.g. "192.168.39.2").

IpPortNr
Type: unsignedShort, Use: required
Specifies the port number at which to receive IP packets.

SrcFilterIpAddress
Type: IpAddressType, Use: optional (default: 0.0.0.0).
Relevant for multicast reception only: for specifying a specific IP address for listening to a single source. The format of the IP address must consist of 4 IP-address parts separated by a dot (e.g. "192.168.39.2").

IpPortNr
Type: unsignedShort, Use: optional (default: 0)
Specifies a specific source port number, or to 0 for accepting IP packets from any source port.

Protocol
Type: IpInProtocolType, Use: optional (default: "*auto*")
Specifies the expected protocol used for encapsulating the transport packets. This can be set to "*auto*", "*udp*" or "*rtp*".

FecMode
Type: FecModeType, Use: optional (default: "*disable*")
Specifies the Error-correction mode used. This can be set to "*disable*" or "*2D*".

VarRateSptsSupport

Type: boolean, Use: optional (default: "*false*")
Specifies the whether variable rate single program transport stream support is enabled. This attribute should be set "*true*" only in case of a variable rate single program transport stream.

TsInPars/File

Element; to specify the file player's playlist. The playlist can be specified by the PlayList attribute that refers to an external file or by specifying the playlist by means of the PlayList element.

Note:
If the playlist is frequently updated then it is recommended to specify the playlist in an external file and to refer to this file with the PlayList attribute.

PlayList

Type: string, Use: optional (default: "")
Specifies the path and filename of the external play-list.

TsInPars/File/PlayList

Optional element; to specify the playlist within the configuration file.

The PlayList contains the following child elements:

1. PlaySettings; Optional element that specifies the settings of the file-player.
2. A sequence of PlayListItem elements, each element specifies one item in the play-list.

Below the graphical diagram of the PlayList element is shown.

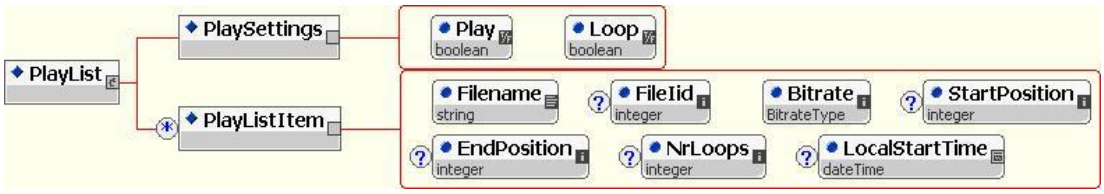


Figure 11. Graphical diagram of PlayList.

PlayList/PlaySettings

Optional element, used for specifying the state of the play-out operation after loading the play-list. It has the following attributes:

Play

Type: boolean, Use: optional, Default: false (=not playing)
Specifies whether the file-player starts or continues the play-out operation.

Loop

Type: boolean, Use: optional, Default: false (=not looping)
Specifies whether the file-player starts play-out of a file to occur in a continuous loop.

LoopList

Type: boolean, Use: optional, Default: false (=not looping)
Specifies whether the file-player starts play-out of the playlist to occur in a loop.

PlayList/PlayListItem

Element that can occur multiple times, the PlayListItem element is used for specifying one item in the play-list. It has the following attributes:

Filename

Type: string, Use: required
Specifies the path of the file to be played-out. The path must start with a drive-letter.

FileIid

Type: integer, Use: optional
The FileIid uniquely identifies an instance of a file (play-list-item) within the play-list and within two subsequent loaded versions of the play-list. The file-iids used in a new play-list determine the behaviour of the file-player. There are three possible cases:

a. The file-iid of the currently playing file is still in the new play-list: the file-player will continue the play-out operation of the current file without interruption.
b. The file-player finished the previous play-list and the file-iid of the lastly played-out file is still in the new play-list: the file-player will start playing-out the file coming after the lastly played-out file.
c. Other situations: the file-player will start playing-out the first file in the new play-list.

Specifying the file-iid, enables the user to extend the play-list with "new" play-list-items and to remove the "old" play-list-items, without interrupting the current play-out operation.

Note(s):

In two subsequent versions of the play-list the user should not re-use the same file-iid for another instance of a file.The user must specify a new file-iid in case one of the play-list-item-attributes of the file changes (a new play-list-item is created).

Bitrate

Type: integer, Use: required
Specifies the bit-rate used for playing-out the file. In case VarTsRate is set to "true" Bitrate specifies the peak bit-rate.

VarTsRate

Type: boolean, Use: optional, Default: false (=constant rate)
Specifies whether the file is contains a variable rate single program Transport Stream.

StartPosition

Type: integer, Use: optional, Default: 0 (= begin of file)
Specifies a byte offset from the start of the file at which to start the file play-out.

EndPosition

Type: integer, Use: optional, Default: 0 (= end of file)
Specifies a byte offset from the start of the file at which to stop the file play-out. If the EndPosition is set to "0" the file is played-out till the end of the file.

NrLoops

Type: integer, Use: optional, Default: 0
Specifies the number of times the play-out of the file is repeated. If the NrLoops is set to "-1" this file is repeated until the play-out start-time of the next file in the play-list.

LocalStartTime

Type: dateTime, Use: optional, Default: As Soon As Possible.
Specifies the play-out start-time of this file. Subsequent play-list-items that specify a start-time must have increasing start-times. If this attribute is absent, the file will be played-out as soon as possible after the previous file in the list is completed.

3.3.2 TsTimeOffset

TsTimeOffset

Optional element. This element specifies the time offset that has to be used in case the time in the incoming tables is not based on UTC-time. The time offset is positive in case the used time in the tables is in advance of the UTC-time. If this element is absent, no time offset is used.

TimeOffset

Type: TimeOffsetType, Use: required

Specifies the time difference between the stream's standard time and UTC-time. The valid range is -24 hours to +24 hours.

For example: when the stream's time is based on JST (Japanese Standard Time) a time offset of +9 hours has to be specified (= "PT9H").

Below the graphical diagram of the TsTimeOffset element is shown.

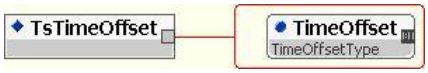


Figure 12. Graphical diagram of TsTimeOffset.

3.3.3 TsTimeCorrection

TsTimeCorrection

Optional element. This element specifies whether and how times in the incoming tables are corrected. This is especially related to the EIT-event's start-times. The applied correction is dependent on the difference between the current UTC-time and the stream's original creation time.

The stream's original creation time can be:

- Derived from the TDT table in the incoming transport stream.
- Set to a fixed date and time.

If the TsTimeCorrection element is absent, no time correction is applied.

The TsTimeCorrection element contains one of the following child elements: ByTsTdt or ByFixedTsTime.

Below the graphical diagram of the TsTimeCorrection element is shown.

Figure 13. Graphical diagram of TsTimeCorrection.

TsTimeCorrection/ByTsTdt

Empty element, specifying that the time correction that is applied is dependent on the difference between current UTC-time and the time in the incoming TDT-table.

TsTimeCorrection/ByFixedTsTime

Element; specifying a fixed date and time for the transport stream.

TsTime

Type: dateTime, Use: required

Specifies the transport stream's original time at the moment the RMC-Data is loaded or re-loaded.

Note:

In case of an 'old' incoming transport stream, the transport stream's original time is in the past but continuously increasing. It is hard to predict accurately the transport stream's original time at the moment the RMC-Data is loaded or re-loaded. Therefore, this option is only preferred in case the incoming transport stream doesn't contain a TDT table.

3.4 Transport Stream Outputs

The TsOuts element is used to specify the contents of the outgoing transport-streams.

TsOuts

The element specifying the settings and the contents of the outgoing transport-streams by means of a sequence of TsOut elements.

Below the graphical diagram of the TsOuts element is shown.

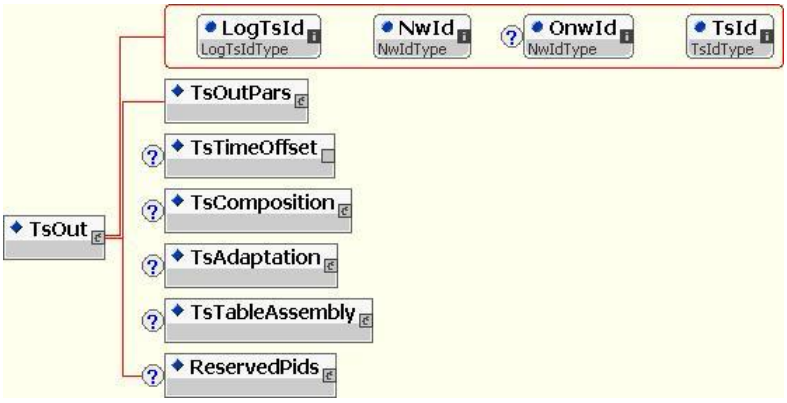


Figure 14. Graphical diagram of TsOuts.

TsOuts/TsOut

Element; specifies the content of a single transport-stream output. The output stream is described with the following child elements: TsOutPars, TsComposition, TsAdaptation, TsTableAssembly and ReservedPids.

LogTsId

Type: LogTsIdType, Use: required
Uniquely identifies a transport-stream output.

NwId

Type: NwIdType, Use: required
Specifies the `network_id` of the transport-stream output.

OnwId

Type: NwIdType, Use: optional (default equal to NwId)
Specifies the `original_network_id` of the outgoing transport-stream. If it is not specified, the `original_network_id` is equal to the `network_id` in the outgoing transport-stream.

TsId

Type: TsdType, Use: required
Specifies the `transport_stream_id` of the transport-stream output.

TsOut/TsOutPars

Optional element, specifying the parameters of the transport-stream output channel. For a single output this element must be present exactly once in the collection of files that describes the RMC-Data. See section 3.4.1.

TsOut/TsTimeOffset

Optional element, specifying the time offset that has to be used in case the time in the outgoing tables is not based on UTC-time. See section 3.4.2.

TsOut/TsComposition

Optional element, specifying the composition of the outgoing transport-stream. For a single output this element must be present exactly once in the collection of files describing the RMC-Data. See section 3.4.3.

TsOut/TsAdaptation

Optional element, specifying the adaptations of the selected services and components. Multiple files that describe the RMC-Data may contribute to the transport-stream adaptations. See section 3.4.4.

TsOut/TsTableAssembly

Optional element, specifying the assembly of the tables for the transport-stream output. Multiple files that describe the RMC-Data may contribute to the transport-stream table assembly. See section 3.4.5.

TsOut/ReservedPids

Optional element, specifying the PIDs which are reserved and will not be allocated by the MuxXpert. However, mandatory PID assignments may overrule the reserved PIDs. See section 3.4.6.

3.4.1 Transport Stream Output Parameters

TsOutPars

Optional element, however for a single output this element must be present exactly once in the collection of files that describe the RMC-Data. This element specifies the settings for the transport-stream output port; it contains one of the following child elements: ASI, ATSC, DTMB, DVBC, DVBH, DVBS, DVBS2, DVBT, IP, QAMB, QAMC or SPI.

Below the graphical diagram of the TsOutPars element is shown.

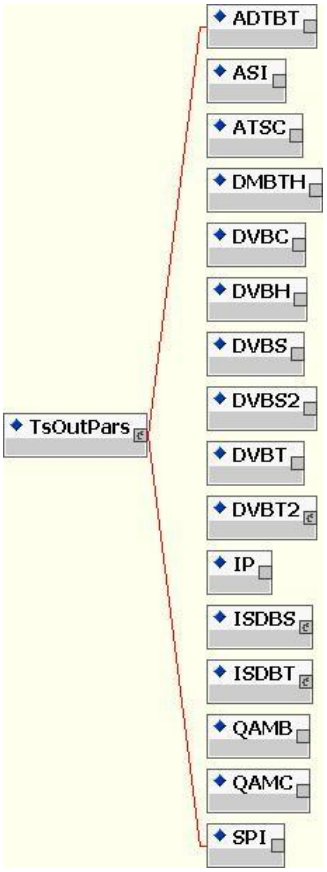


Figure 15. Graphical diagram of TsOutPars.

TsOutPars/ADTBT

Element; to specify the parameters of an ADTB-T output port.

DtmbBandwidth

Type: DtmbBandwidthType, Use: required
Specifies the constellation type. This can be set to "*5MHz*", "*6MHz*", "*7MHz*" or "*8MHz*".

DtmbConstellation

Type: DtmbConstellationType, Use: required
Specifies the constellation type. This can be set to "*qam4-nr*", "*qam4*", "*qam16*", "*qam32*", or "*qam64*". **Note that the values "*qam4-nr*" and "*qam32*"are only allowed in combination with FEC code rate value "*0.8*".**

DtmbFrameHeader

Type: DtmbFrameHeaderType Use: required
Specifies the frame header mode. This can be set to "*pn420*", "*pn595*" or "*pn945*".

DtmbCodeRate

Type: DtmbCodeRateType, Use: required
Specifies the FEC code rate. This can be set to "*0.4*", "*0.6*" or "*0.8*".

DtmbInterleaverMode

Type: DtmbInterleaverModeType, Use: required
Specifies the interleaver mode. This can be set to "*1*" or "*2*".

DtmbFrameNumbering
Type: OnOffType, Use: required
Specifies whether frame numbering must be used. This can be set to "*off*" or "*on*".

DtmbPilots
Type: OnOffType, Use: required
Specifies whether pilots must be added. This can be set to "*off*" or "*on*".

RfFrequency
Type: RfFrequencyType, Use: optional (required for adapters with RF up-converter).
Specifies the carrier frequency for the RF up-converter

RfLevel
Type: dBmType, Use: optional (default: 27.5 dBm for adapters that support adjustable RF-level)
Specifies the output level of the RF-signal

Below the graphical diagram of the ADTBT element is shown.

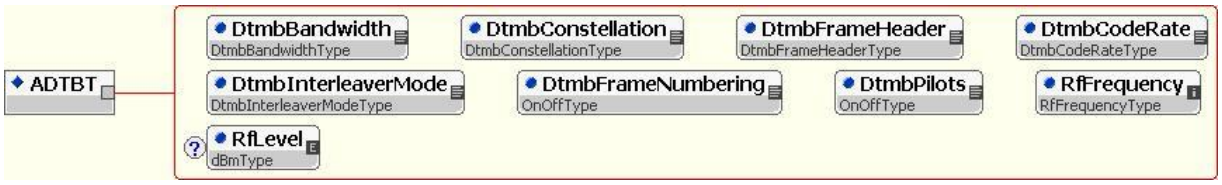


Figure 16. Graphical diagram of ADTBT.

TsOutPars/ASI
Element; to specify the parameters of an ASI output port.

Bitrate
Type: BitrateType, Use: required
Specifies the transport-stream output bitrate.

TransmitMode
Type: TransmitModeType, Use: required
Specifies the channel's transmit mode, this can be set to 188-byte transmit mode ("*188*") or 204-byte transmit mode by adding 16 bytes ("*add16*").

Below the graphical diagram of the ASI element is shown.

Figure 17. Graphical diagram of ASI.

TsOutPars/ATSC
Element; to specify the parameters of an ATSC output port.

AtscConstellation
Type: AtscConstellationType, Use: required
Specifies the constellation type. This can be set to "*8-vsb*" or "*16-vsb*".

RfFrequency
Type: RfFrequencyType, Use: optional (required for adapters with RF up-converter).
Specifies the carrier frequency for the RF up-converter

RfLevel
Type: dBmType, Use: optional (default: 27.5 dBm for adapters that support adjustable RF-level)

Specifies the output level of the RF-signal

Below the graphical diagram of the ATSC element is shown.

Figure 18. Graphical diagram of ATSC.

TsOutPars/DMBTH

Element; to specify the parameters of a DMB-T/H output port.

DtmbBandwidth

Type: DtmbBandwidthType, Use: required
Specifies the constellation type. This can be set to "*5MHz*", "*6MHz*", "*7MHz*" or "*8MHz*".

DtmbConstellation

Type: DtmbConstellationType, Use: required
Specifies the constellation type. This can be set to "*qam4-nr*", "*qam4*", "*qam16*", "*qam32*", or "*qam64*". **Note that the values "*qam4-nr*" and "*qam32*"are only allowed in combination with FEC code rate value "*0.8*".**

DtmbFrameHeader

Type: DtmbFrameHeaderType Use: required
Specifies the frame header mode. This can be set to "*pn420*" or "*pn945*".

DtmbCodeRate

Type: DtmbCodeRateType, Use: required
Specifies the FEC code rate. This can be set to "*0.4*", "*0.6*" or "*0.8*".

DtmbInterleaverMode

Type: DtmbInterleaverModeType, Use: required
Specifies the interleaver mode. This can be set to "*1*" or "*2*".

DtmbFrameNumbering

Type: OnOffType, Use: required
Specifies whether frame numbering must be used. This can be set to "*off*" or "*on*".

RfFrequency

Type: RfFrequencyType, Use: optional (required for adapters with RF up-converter).
Specifies the carrier frequency for the RF up-converter

RfLevel

Type: dBmType, Use: optional (default: 27.5 dBm for adapters that support adjustable RF-level)
Specifies the output level of the RF-signal

Below the graphical diagram of the DMBTH element is shown.

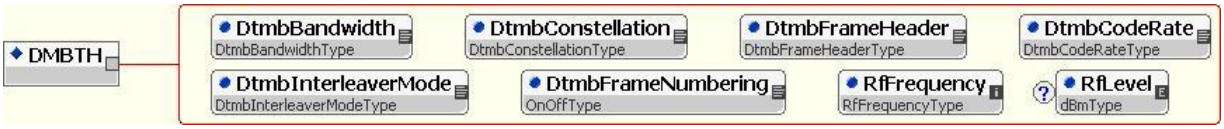


Figure 19. Graphical diagram of DMBTH.

TsOutPars/DVBC

Element; to specify the parameters of a DVB-C output port.

SymbolRate

Type: SymbolRateType, Use: required
Specifies the symbol-rate of the DVB-C output stream.

QamModulationType

Type: QamModulationTypeType, Use: required
Specifies the modulation type. This can be set to "*qam16*", "*qam32*", "*qam64*", "*qam128*" or "*qam256*".

RfFrequency

Type: RfFrequencyType, Use: optional (required for adapters with RF up-converter).
Specifies the carrier frequency for the RF up-converter

RfLevel

Type: dBmType, Use: optional (default: 27.5 dBm for adapters that support adjustable RF-level)
Specifies the output level of the RF-signal

Below the graphical diagram of the DVBC element is shown.

Figure 20. Graphical diagram of DVBC.

TsOutPars/DVBC2

Element; specifying the modulation parameters of a DVB-C2 output port. The DVB-C2 modulation is described with the following child elements: General, DSlice0 and Rf

TsOutPars/DVBC2/General

Element; specifying the general DVB-C2 parameters

Bandwidth

Type: Dvbc2BandwidthType, Use: required
Specifies the bandwidth of channel raster of the network. This can be set to "*6MHz*" or "*8MHz*".

C2Bandwidth

Type: unsignedInt, Use: required
Specifies the bandwidth of the generated signal in multiples of pilot carrier spacing. The valid range is 0 ... 65535.

StartFrequency

Type: unsignedInt, Use: required
Start frequency of the C2-System by means of the distance from 0Hz in multiples of the carrier spacing. The valid range is 0 ... 0xFFFFFF and multiples of D_X. (D_X=24 for guard interval 1/128 and D_X=12 for guard interval 1/64).

GuardInterval

Type: Dvbc2GuardIntervalType, Use: required
Specifies the guard interval used for the modulated signal. This can be set to "*1_128*or "*1_64*".

ReservedTone

Type: Boolean, Use: required
Specifies whether one or more tones (carriers) are reserved. This can be set to "*true*" or "*false*".

L1TiMode
Type: Dvbc2L1TiModeType, Use: required
Specifies the L1 time interleaving mode. This can be set to "*none*", "*best_fit*", "*4_symbols*" or "*8_symbols*".

NetworkId
Type: unsignedInt, Use: required
Specifies the DVB-C2 network ID.

SystemId
Type: unsignedInt, Use: required
Specifies the DVB-C2 system ID.

TsOutPars/DVBC2/DSlice0
Element; specifying the DVB-C2 parameters of Data Slice 0 and containing the child element: Plp0.

Type
Type: Dvbc2DSliceType, Use: required
Specifies the data slice type. This can be set to "*type1*" or "*type2*".

TunePosition
Type: unsignedInt, Use: required
Tune position of the associated data slice relative to the start frequency of the C2-System in multiples of pilot carrier spacing.
The valid range is 0 ... 8191 if the guard interval is 1/128.
The valid range is 0 ... 16383 if the guard interval is 1/64.

OffsetLeft
Type: unsignedInt, Use: required
Start position of the associated data slice by means of the distance to the left from the tuning position in multiples of the pilot carrier spacing.
The valid range is -128 ... 127 if the guard interval is 1/128.
The valid range is -256 ... 255 if the guard interval is 1/64.

OffsetRight
Type: unsignedInt, Use: required
End position of the associated data slice by means of the distance to the right from the tuning position in multiples of the pilot carrier spacing.
The valid range is -128 ... 127 if the guard interval is 1/128.
The valid range is -256 ... 255 if the guard interval is 1/64.
If OffsetLeft equals OffsetRight, the data slice is empty and no input streams are created for the PLPs of the data slice.

FecHdr
Type: Dvbc2DSliceFecHdrType, Use: required
Specifies the FEC frame header type. This can be set to "*hem*" or "*robust*".

TiDepth
Type: Dvbc2DSliceTiDepthType, Use: required
Specifies the time interleaving depth within the data slice. This can be set to "*none*", "*4_symbols*", "*8_symbols*" or "*16_symbols*".

TsOutPars/DVBC2/DSlice0/Plp0
Element; specifying the DVB-C2 parameters of Physical Layer Pipe 0.

Modulation
Type: Dvbc2PlpModulationType, Use: required
Specifies the modulation type used by the PLP. This can be set to "*qam16*", "*qam64*", "*qam256*", "*qam1024*" or "*qam4096*".

CodeRate
Type: Dvbc2PlpCodeRateType, Use: required
Specifies the code rate used by the PLP. This can be set to "*2_3*", "*3_4*", "*4_5*", "*5_6*", "*8_9*" or "*9_10*".

Fec
Type: Dvbc2PlpFecType, Use: required
Specifies the FEC type used by the PLP: 16K LDPC or 64K LDPC. This can be set to "*16k*" or "*64k*".

Issy
Type: Boolean, Use: required
Specifies whether Input Stream Synchronization mechanism is used. This can be set to "*true*" or "*false*".

IssyBufs
Type: unsignedInt, Use: required
Specifies the ISSY 'BUFS' value. The valid range is 0 … 2097151.

Hem
Type: Boolean, Use: required
Specifies whether High Efficiency Mode or Normal Mode is used. This can be set to "*true*" or "*false*".

Npd
Type: Boolean, Use: required
Specifies whether null-packet deletion is used. This can be set to "*true*" or "*false*".

HeaderCounter
Type: unsignedInt, Use: required
Specifies the header counter field, the number of FEC-frames following the FEC-frame header. This can be set to:
"*0*": One FEC-frame follows after the FEC-frame header.
"*1*": Two FEC-frames follow after the FEC-frame header.

TsRate
Type: unsignedInt, Use: required
Specifies the PLP transport stream rate in bps.
If it is set to '0', and no ISSY is used and null-packet deletion is not active then the transport stream rate is computed from the PLP parameters.

TsOutPars/DVBC2/Rf
Optional element; specifying the RF parameters of the DVB-C2 capable modulator card with RF up-converter.

RfFrequency
Type: RfFrequencyType, Use: required
Specifies the carrier frequency for the RF up-converter

RfLevel
Type: dBmType, Use: optional (default: 27.5 dBm for adapters that support adjustable RF-level)

Specifies the output level of the RF-signal

Below the graphical diagram of the DVBC2 element is shown.

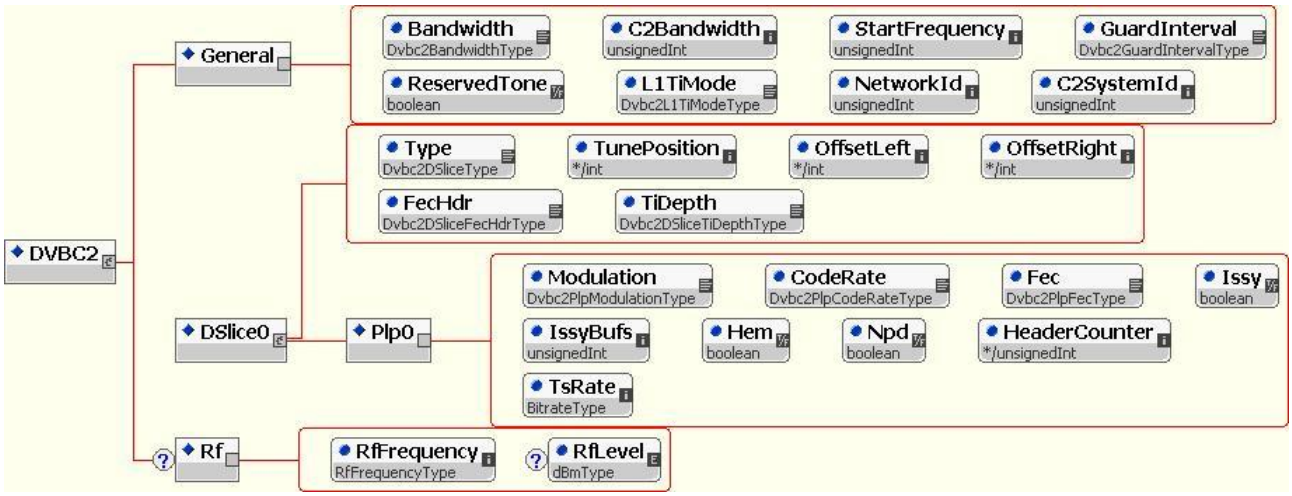


Figure 21. Graphical diagram of DVBC2.

TsOutPars/DVBH

Element; to specify the parameters of a DVB-H output port.

OfdmCodeRate

Type: OfdmCodeRateType, Use: required
Specifies the convolutional rate used for the modulated signal. This can be set to "*1_2*", "*2_3*", "*3_4*", "*5_6*" or "*7_8*".

OfdmBandwidth

Type: DvbhOfdmBandwidthType, Use: required
Specifies the bandwidth used for the modulated signal. This can be set to "*5MHz*", "*6MHz*", "*7MHz*" or "*8MHz*".

OfdmConstellation

Type: OfdmConstellationType, Use: required
Specifies the constellation used for the modulated signal. This can be set to "*qpsk*", "*qam16*" or "*qam64*".

OfdmGuardInterval

Type: OfdmGuardIntervalType, Use: required
Specifies the guard interval used for the modulated signal. This can be set to "*1_32*", "*1_16*", "*1_8*" or "*1_4*".

OfdmTransmissionMode

Type: OfdmTransmissionModeType, Use: required
Specifies the transmission mode used for the modulated signal. This can be set to "*2k*", "*4k*" or "*8k*".

OfdmInterleaving

Type: OfdmInterleavingType, Use: required
Specifies the interleaving type used for the modulated signal. This can be set to "*indepth*" or "*native*".

OfdmCellIdentifier

Type: unsignedShort, Use: optional
Specifies the DVB-H cell identifier.

OfdmTimeSlicing
Type: UsedType, Use: optional
Specifies whether the DVB-H time slicing is used. This can be set to "*used*" or "*unused*".

OfdmMpeFec
Type: UsedType, Use: optional
Specifies whether the DVB-H MPE-FEC is used. This can be set to "*used*" or "*unused*".

RfFrequency
Type: RfFrequencyType, Use: optional (required for adapters with RF up-converter).
Specifies the carrier frequency for the RF up-converter

RfLevel
Type: dBmType, Use: optional (default: 27.5 dBm for adapters that support adjustable RF-level)
Specifies the output level of the RF-signal

Below the graphical diagram of the DVBH element is shown.

Figure 22. Graphical diagram of DVBH.

TsOutPars/DVBS
Element; to specify the parameters of a DVB-S output port.

SymbolRate
Type: SymbolRateType, Use: required
Specifies the symbol-rate of the DVB-S output stream.

DvbsModulationType
Type: DvbsModulationTypeType, Use: required
Specifies the modulation type. This must be set to QPSK ("*qpsk*").

DvbsCodeRate
Type: DvbsCodeRateType, Use: required
Specifies the convolutional rate used for the modulated signal. This can be set to "*1_2*", "*2_3*", "*3_4*", "*5_6*" or "*7_8*"

RfFrequency
Type: RfFrequencyType, Use: required
Specifies the carrier frequency for the RF up-converter

Below the graphical diagram of the DVBS element is shown.

Figure 23. Graphical diagram of DVBS.

TsOutPars/DVBS2
Element; to specify the parameters of a DVB-S.2 output port.

SymbolRate
Type: SymbolRateType, Use: required

Specifies the symbol-rate of the DVB-S.2 output stream.

DvbS2ModulationType
Type: DvbS2ModulationTypeType, Use: required
Specifies the modulation type. This can be set to QPSK ("*qpsk*") or 8-PSK ("*8-psk*").

DvbS2CodeRate
Type: DvbS2CodeRateType, Use: required
Specifies the convolutional rate used for the modulated signal. This can be set to "*1_4*", "*1_3*", "*2_5*", "*1_2*", "*3_5*", "*2_3*", "*3_4*", "*4_5*", "*5_6*", "*8_9*" or "*9_10*"

RfFrequency
Type: RfFrequencyType, Use: required
Specifies the carrier frequency for the RF up-converter

Below the graphical diagram of the DVBS2 element is shown.

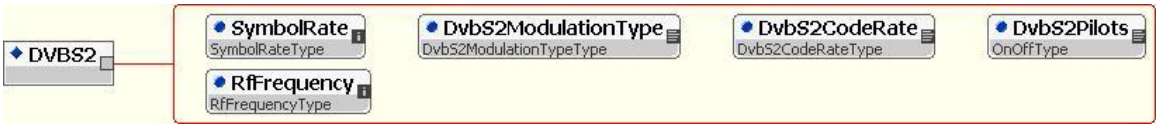


Figure 24. Graphical diagram of DVBS2.

TsOutPars/DVBT
Element; to specify the parameters of a DVB-T output port.

OfdmCodeRate
Type: OfdmCodeRateType, Use: required
Specifies the convolutional rate used for the modulated signal. This can be set to "*1_2*", "*2_3*", "*3_4*", "*5_6*" or "*7_8*".

OfdmBandwidth
Type: OfdmBandwidthType, Use: required
Specifies the bandwidth used for the modulated signal. This can be set to "*5MHz*", "*6MHz*", "*7MHz*" or "*8MHz*".

OfdmConstellation
Type: OfdmConstellationType, Use: required
Specifies the constellation used for the modulated signal. This can be set to "*qpsk*", "*qam16*" or "*qam64*".

OfdmGuardInterval
Type: OfdmGuardIntervalType, Use: required
Specifies the guard interval used for the modulated signal. This can be set to "*1_32*", "*1_16*", "*1_8*" or "*1_4*".

OfdmTransmissionMode
Type: DvbtOfdmTransmissionModeType, Use: required
Specifies the transmission mode used for the modulated signal. This can be set to "*2k*" or "*8k*".

RfFrequency
Type: RfFrequencyType, Use: optional (required for adapters with RF up-converter).
Specifies the carrier frequency for the RF up-converter

RfLevel
Type: dBmType, Use: optional (default: 27.5 dBm for adapters that support adjustable RF-level)
Specifies the output level of the RF-signal

Below the graphical diagram of the DVBT element is shown.

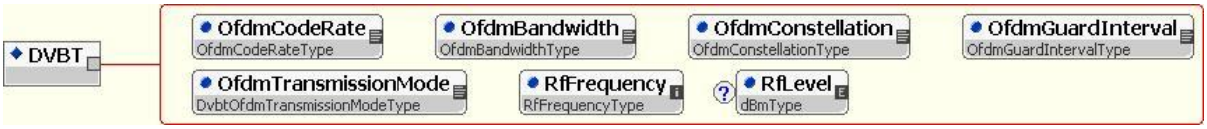


Figure 25. Graphical diagram of DVBT.

TsOutPars/DVBT2

Element; specifying the modulation parameters of a DVB-T2 output port. The DVB-T2 modulation is described with the following child elements: General, T2Frame, Fef, Plp0 and Rf

TsOutPars/DVBT2/General

Element; specifying the general DVB-T2 parameters

Bandwidth

Type: Dvbt2BandwidthType, Use: required
Specifies the bandwidth used for the modulated signal. This can be set to "*1_7MHz*", "*5MHz*", "*6MHz*", "*7MHz*", "*8MHz*" or "*10MHz*".

ExtBandwidth

Type: Boolean, Use: required
Specifies the whether extended carrier mode is used. This can be set to "*true*" or "*false*".

FftMode

Type: Dvbt2FftModeType, Use: required
Specifies the FFT mode (or size) in the channel. This can be set to "*1k*", "*2k*", "*4k*", "*8k*", "*16k*" or "*32k*".

GuardInterval

Type: Dvbt2GuardIntervalType, Use: required
Specifies the guard interval used for the modulated signal. This can be set to "*1_128*", "*1_32*", "*1_16*", "*1_8*", "*1_4*", "*19_128*" or "*19_256*".

L1Modulation

Type: Dvbt2L1ModulationType, Use: required
Specifies the L1 Modulation, the constellation of the L1-post signalling block. This can be set to "*bpsk*", "*qpsk*", "*qam16*" or "*qam64*".

PilotPattern

Type: unsignedShort, Use: required
Specifies the Pilot Pattern used for the data OFDM symbols. The valid range is 1..8.

Papr

Type: DvbT2PaprType, Use: required
Specifies the PAPR reduction used: None, ACE only, TR only or both ACE and TR constellation. Note that PAPR reduction is only signalled. This can be set to "*none*", "*ace*", "*tr*" or "*ace_tr*".

Miso

Type: DvbT2MisoType, Use: required
Specifies the Multiple Input Single Output mode: Off (=SISO), Tx1 only, Tx2 only or the sum of Tx1 and Tx2. This can be set to "*off*", "*tx1*", "*tx2*" or "*sum*".

NetworkId

Type: unsignedInt, Use: required

Specifies the DVB-T2 network ID.

SystemId
Type: unsignedInt, Use: required
Specifies the DVB-T2 system ID.

CellId
Type: unsignedInt, Use: required
Specifies the DVB-T2 cell ID.

Frequency
Type: unsignedInt, Use: required
Specifies the DVB-T2 frequency in the L1 signalling.

TsOutPars/DVBT2/T2Frame
Element; specifying the DVB-T2 frame structure parameters

NumT2Frames
Type: unsignedInt, Use: required
Specifies the number of T2-frames in a super frame.

NumDataSymbols
Type: unsignedInt, Use: required
Specifies the number of data OFDM symbols per T2-frame, excluding P1 and P2.

TsOutPars/DVBT2/Fef
Optional element; to specify the insertion of Future Extension Frames (FEFs).

Type
Type: unsignedInt, Use: required
Specifies the FEF-type. The valid range is 0..15.

S1
Type: unsignedInt, Use: required
Specifies the S1-field value in the P1 signalling data. The valid range is 2..7.

S2
Type: unsignedInt, Use: required
Specifies the S2-field value in the P1 signalling data. The valid values: 1, 3, 5, 7, 9, 11, 13 and 15.

Signal
Type: Dvbt2FefSgnalType, Use: required
Specifies the generated signal during the FEF period can be zero or a random test signal made from 1k OFDM transformations of a PRBS signal. This can be set to "*zero*" or "*prbs*".

Length
Type: unsignedInt, Use: required
Specifies the number of T2-frames in a super frame.

Interval
Type: unsignedInt, Use: required
Specifies the number of T2-frames between two FEF-parts.

TsOutPars/DVBT2/Plp0

Element; specifying the DVB-T2 parameters of Physical Layer Pipe 0.

Modulation

Type: Dvbt2PlpModulationType, Use: required
Specifies the modulation type used by the PLP. This can be set to "*qpsk*", "*qam16*", "*qam64*" or "*qam256*".

CodeRate

Type: Dvbt2PlpCodeRateType, Use: required
Specifies the code rate used by the PLP. This can be set to "*1_2*", "*2_3*", "*3_4*", "*3_5*", "*4_5*" or "*5_6*".

Fec

Type: Dvbt2PlpFecType, Use: required
Specifies the FEC type used by the PLP: 16K LDPC or 64K LDPC. This can be set to "*16k*" or "*64k*".

IlType

Type: unsignedInt, Use: required
Specifies the Time IL Type. This can be set to:
"*0*": One Interleaving Frame corresponds to one T2-frame
"*1*": One Interleaving Frame is carried in multiple T2-frames.

Length

Type: unsignedInt, Use: required
Specifies the Time IL Length.
If the Time IL Type is set to '0', this parameter specifies the number of TI-block per Interleaving Frame
If the Time IL Type is set to '1', this parameter specifies the number of T2-frames to which each Interleaving Frame is mapped

Issy

Type: Boolean, Use: required
Specifies whether Input Stream Synchronization mechanism is used. This can be set to "*true*" or "*false*".

Hem

Type: Boolean, Use: required
Specifies whether High Efficiency Mode or Normal Mode is used. This can be set to "*true*" or "*false*".

Rotation

Type: Boolean, Use: required
Specifies whether constellation rotation is used. This can be set to "*true*" or "*false*".

NumBlocks

Type: unsignedInt, Use: required
Specifies the number of FEC blocks contained in an Interleaving.

TsOutPars/DVBT2/Rf

Optional element; specifying the RF parameters of the DVB-T2 capable modulator card with RF up-converter.

RfFrequency

Type: RfFrequencyType, Use: required

Specifies the carrier frequency for the RF up-converter

RfLevel

Type: dBmType, Use: optional (default: 27.5 dBm for adapters that support adjustable RF-level)
Specifies the output level of the RF-signal

Below the graphical diagram of the DVBT2 element is shown.

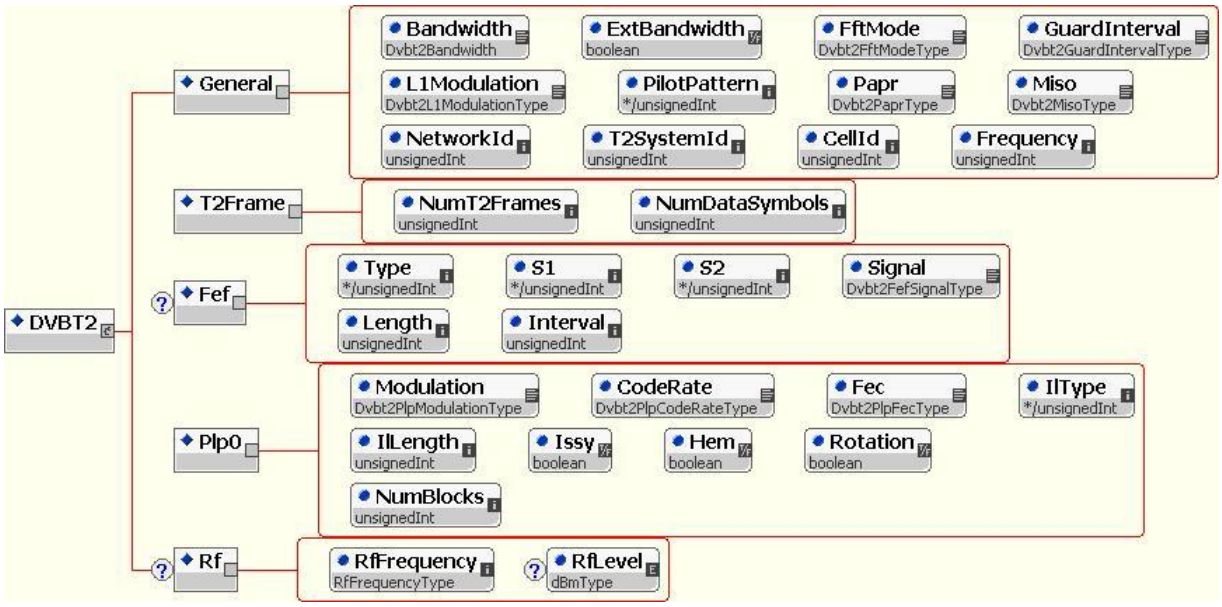


Figure 26. Graphical diagram of DVBT2.

TsOutPars/IP

Element; to specify the parameters of an IP output port.

Bitrate

Type: BitrateType, Use: required
Specifies the transport-stream output bitrate.

IpAddress

Type: IpAddressType, Use: required
Specifies the destination IP address for transmission. The format of the IP address must consist of 4 IP-address parts separated by a dot (e.g. "192.168.39.2").

IpPortNr

Type: unsignedShort, Use: required
Specifies the destination port number.

Protocol

Type: IpInProtocolType, Use: required
Specifies the protocol used for encapsulating the transport packets. This can be set to "*udp*" or "*rtp*".

FecMode

Type: FecModeType, Use: required
Specifies the Error-correction mode used. This can be set to "*disable*" or "*2D*".

FecNumRows

Type: unsignedInt, Use: optional (default: 0)
Specifies the number of rows in the FEC matrix[1].

FecNumCols
Type: unsignedInt, Use: optional (default 0)
Specifies the number of columns in the FEC matrix[1].

IpTimeToLive
Type: unsignedByte, Use: optional (default 0)
Specifies the Time-To-Live (TTL) value to be used for multicast transmission.

NumTpPerIp
Type: unsignedInt, Use: optional (default 7)
Specifies the number of Transport Packets (TPs) stored in one IP packet. The valid range is 1..7.

Below the graphical diagram of the IP element is shown.

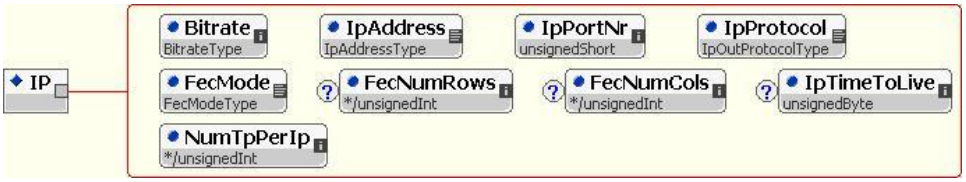


Figure 27. Graphical diagram of IP.

TsOutPars/ISDBS
Element; specifying the modulation parameters of an ISDB-S output port. The ISDB-S modulation for the three hierarchical layers is described with the following child elements: LayerPArs, RelTs2TsId and Slot2RelTs.

Emergency
Type: Boolean, Use: required
Specifies whether the switch-on control flag for emergency broadcast should be turned on. This can be set to "*true*" or "*false*".

RfFrequency
Type: RfFrequencyType, Use: optional (required for adapters with RF up-converter).
Specifies the carrier frequency for the RF up-converter

TsOutPars/ISDBS/LayerPars
Element; specifying the modulation parameters of an ISDB-S layer. Although the schema allows you to specify multiple layers, maximum one layer is supported.

Layer
Type: unsignedInt, Use: required
Specifies the ISDB-S layer. The valid range is 0..3.

NumSlots
Type: unsignedInt, Use: required
Specifies the number of slots allocated for this layer. The valid range is 0..48. The total number of slots allocated shall be 48.

ModCod
Type: IsdbsModCodType, Use: required

[1] In the *Code of Practice #3* the rows and columns parameters are called D and L respectively. The following restrictions apply to L and D: $4 \le D \le 20$, $1 \le L \le 20$ and $L*D \le 100$

Specifies the modulation type and code rate used for the layer. This can be set to "*bpsk_1_2*", "*qpsk_1_2*", "*qpsk_2_3*", "*qpsk_3_4*", "*qpsk_5_6*", "*qpsk_7_8*", "*8psk_2_3*" or "*not_alloc*".

TsOutPars/ISDBS/RelTs2TsId

Element; specifying the mapping between a relative transport stream and a transport stream ID. Although the schema allows you to specify multiple mappings, maximum one mapping is supported.

RelativeTs

Type: unsignedInt, Use: required
Specifies the value of the relative transport stream. The valid range is 0..7.

TsId

Type: TsdType, Use: required
Specifies the `transport_stream_id` related to the relative transport stream.

TsOutPars/ISDBS/Slot2RelTs

Element; specifying the mapping between a range of slot numbers and a relative transport stream.

MinSlotNr

Type: unsignedInt, Use: required
Specifies the value of the relative transport stream. The valid range is 1..48.

MinSlotNr

Type: unsignedInt, Use: required
Specifies the value of the relative transport stream. The valid range is 1..48.

RelativeTs

Type: unsignedInt, Use: required
Specifies the value of the relative transport stream. The valid range is 0..7.

Below the graphical diagram of the ISDBS element is shown.

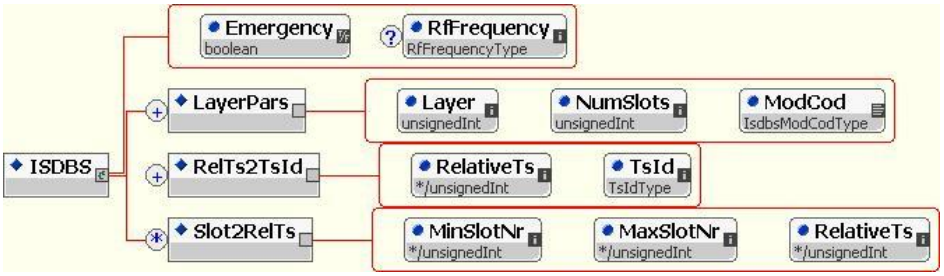


Figure 28. Graphical diagram of ISDBS.

TsOutPars/ISDBT

Element; specifying the modulation parameters of an ISDB-T output port. The ISDB-T modulation for the three hierarchical layers is described with the following child elements: LayerA, LayerB and LayerC (of type *IsdbtLayerPars* see section 3.4.1.1)

BroadcastType

Type: IsdbtBroadcastTypeType, Use: required

Specifies the ISDB-T broadcast type. This can be set to "*tv*" (for 13-segment TV broadcast), "*rad1*" (for 1-segment radio broadcast), "*rad3*" (for 3-segment radio broadcast) or "*tv1*" (for 1-segment TV broadcast).

TransmissionMode

Type: IsdbtTmModeType, Use: required
Specifies the ISDB-T transmission mode. This can be set to "*1*" (for Mode 1: 2k), "*2*" (for Mode 2: 4k) or "*3*" (for Mode 3: 8k).

GuardInterval

Type: IsdbtGuardIntervalType, Use: required
Specifies the ISDB-T guard interval length. This can be set to "*1_32*", "*1_16*", "*1_8*" or "*1_4*".

PartialRx

Type: Boolean, Use: required
Specifies whether layer A is used for partial reception. This can be set to "*true*" or "*false*".

Emergency

Type: Boolean, Use: required
Specifies whether the switch-on control flag for emergency broadcast should be turned on. This can be set to "*true*" or "*false*".

IipPid

Type: PidType, Use: required
Specifies the PID value used for multiplexing the IIP packet.

DefaultLayer

Type: IsdbtLayerType, Use: optional (default: "*none*")
Specifies the default ISDB-T layer. All transport stream elements that are not explicitly mapped in an ISDB-T layer are mapped in the default layer. This can be set to "*a*", "*b*", "*c*" or "*none*".

RfFrequency

Type: RfFrequencyType, Use: optional (required for adapters with RF up-converter).
Specifies the carrier frequency for the RF up-converter

RfLevel

Type: dBmType, Use: optional (default: 27.5 dBm for adapters that support adjustable RF-level)
Specifies the output level of the RF-signal

Below the graphical diagram of the ISDBT element is shown.

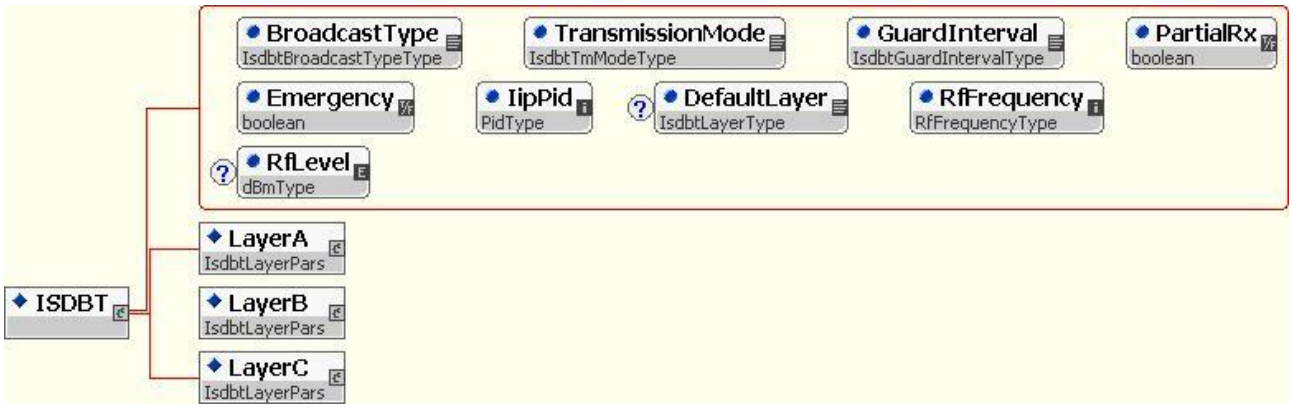


Figure 29. Graphical diagram of ISDBT.

TsOutPars/QAMB

Element; to specify the parameters of a QAM-B output port.

QamModulationType

Type: QambModulationTypeType, Use: optional (required for adapters with RF up-converter).
Specifies the modulation type. This can be set to "*qam64*" or "*qam256*".

QambInterleaving

Type: QambInterleavingType, Use: required
Specifies the QAM-B interleaving mode used. The possible interleaving values are specified in the table below.

Value	CW (Code Word)	I (# of taps)	J (increment)
"*i128_j1d*"	0001	128	1
"*i64_j2*"	0011	64	2
"*i32_j4*"	0101	32	4
"*i16_j8*"	0111	16	8
"*i8_j16*"	1001	8	16
"*i128_j1*"	0000	128	1
"*i128_j2*"	0010	128	2
"*i128_j3*"	0100	128	3
"*i128_j4*"	0110	128	4
"*i128_j5*"	1000	128	5
"*i128_j6*"	1010	128	6
"*i128_j7*"	1100	128	7
"*i128_j8*"	1110	128	8

RfFrequency

Type: RfFrequencyType, Use: required
Specifies the carrier frequency for the RF up-converter

RfLevel

Type: dBmType, Use: optional (default: 27.5 dBm for adapters that support adjustable RF-level)
Specifies the output level of the RF-signal

Below the graphical diagram of the QAMB element is shown.

Figure 30. Graphical diagram of QAMB.

TsOutPars/QAMC

Element; to specify the parameters of a QAM-C output port.

SymbolRate

Type: SymbolRateType, Use: required
Specifies the symbol-rate of the QAM-C output stream.

QamModulationType

Type: QamModulationTypeType, Use: required
Specifies the modulation type. This can be set to "*qam16*", "*qam32*", "*qam64*", "*qam128*" or "*qam256*".

RfFrequency

Type: RfFrequencyType, Use: optional (required for adapters with RF up-converter).
Specifies the carrier frequency for the RF up-converter

RfLevel

Type: dBmType, Use: optional (default: 27.5 dBm for adapters that support adjustable RF-level)
Specifies the output level of the RF-signal

Below the graphical diagram of the QAMC element is shown.

Figure 31. Graphical diagram of QAMC.

TsOutPars/SPI
Element; to specify the parameters of an SPI output port.

Bitrate

Type: BitrateType, Use: required
Specifies the transport-stream output bitrate.

TransmitMode

Type: TransmitModeType, Use: required
Specifies the channel's transmit mode, this can be set to 188-byte transmit mode ("*188*") or 204-byte transmit mode by adding 16 bytes ("*add16*").

Below the graphical diagram of the SPI element is shown.

Figure 32. Graphical diagram of SPI.

3.4.1.1 IsdbtLayerPars Type

The IsdbtLayerPars type is used for the specification of ISDB-T modulation for a single hierarchical layer.

IsdbtLayerPars
An element of this type specifies the ISDB-T modulation for a single hierarchical layer and it specifies which items from the outgoing transport stream have to be mapped to this layer.
Which services, service components, EMM streams, elementary streams and tables are mapped to his layer is specified by the following child elements: TsSvcComponents, SvcsComponents, EmmStreams, ElemStreams or Tables.

NumSegments

Type: unsignedInt, Use: required
Specifies the number of segments used in this layer. The sum of the NumSegments of the three ISDB-T layers must be 13.

ModulationType

Type: IsdbtModulationTypeType, Use: required
Specifies the modulation type applied to the segments in this layer. This can be set to "*dpsk*", "*qpsk*", "*qam16*" or "*qam64*".

CodeRate

Type: IsdbtCodeRateType, Use: required
Specifies the convolutional coding rate applied to the segments in this layer. This can be set to "*1_2*", "*2_3*", "*3_4*", "*5_6*" or "*7_8*".

TimeInterleave

Type: unsignedInt, Use: required
Specifies the time interleaving. The valid range is 0..4.

Below the graphical diagram of the *IsdbtLayerPars* is shown.

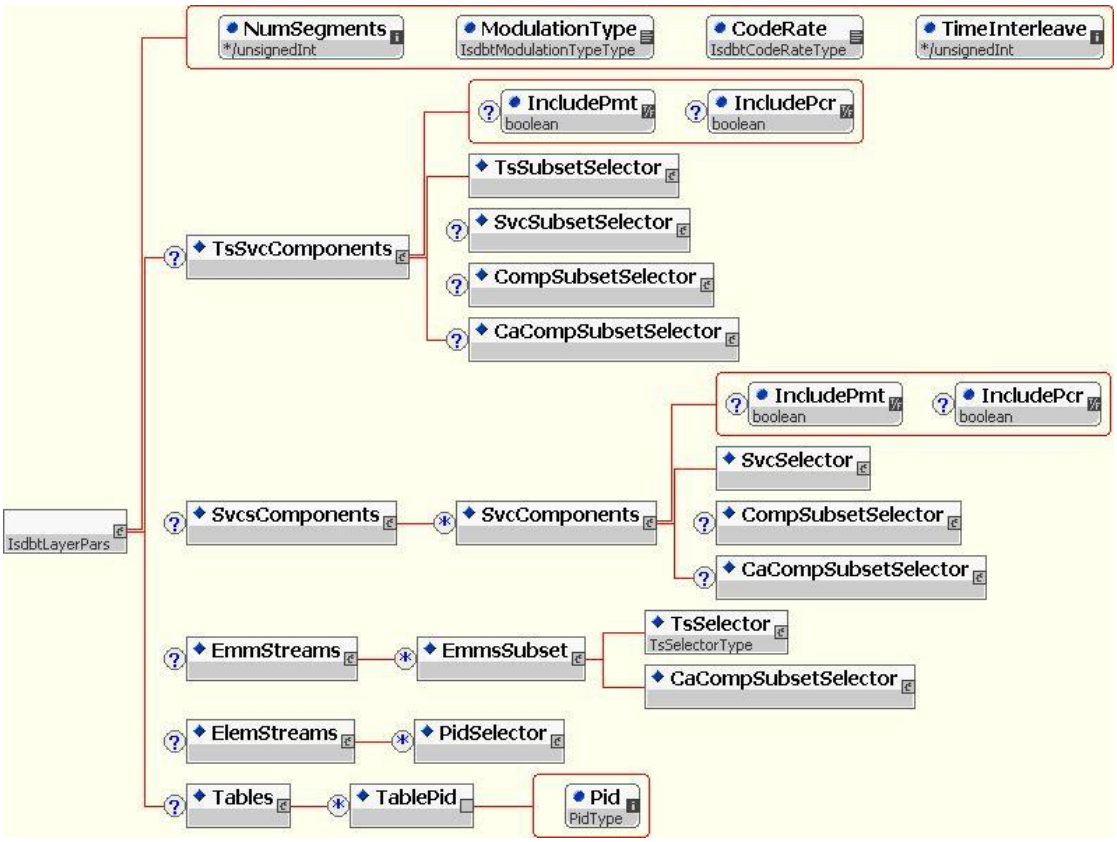


Figure 33. Graphical diagram of *IsdbtLayerPars*.

Figure 34 illustrates the selection of services, service components, EMM streams, elementary streams and tables from the composed outgoing transport stream that have to be mapped to this hierarchical layer.

The first row shows the selection of services and service components that originate from selected incoming transport streams that have to be mapped to this hierarchical layer. First a transport-stream subset selector is used to get all services from the composed outgoing transport streams that originate from the selected incoming transport streams. Thereafter a service subset selector is used to exclude services from being mapped to this layer. A component subset selector is used to select

component groups to exclude services from being mapped to this layer. Further a CA-component subset selector is used to select ECM-streams to be excluded.

The second row shows the selection of individual services and their components that have to be mapped to this hierarchical layer. First a service selector is used to get the service from the composed outgoing transport stream. Thereafter a component subset selector is used to select component groups from the selected service that have to be excluded from being mapped to this layer. Further a CA-component subset selector is used to select ECM-streams to be excluded.

The third and fourth rows show the selection of EMM-Streams and of Elementary Streams (unreferenced PIDs) that have to be mapped to this hierarchical layer.

Finally, the tables are selected that have to be mapped to this hierarchical layer.

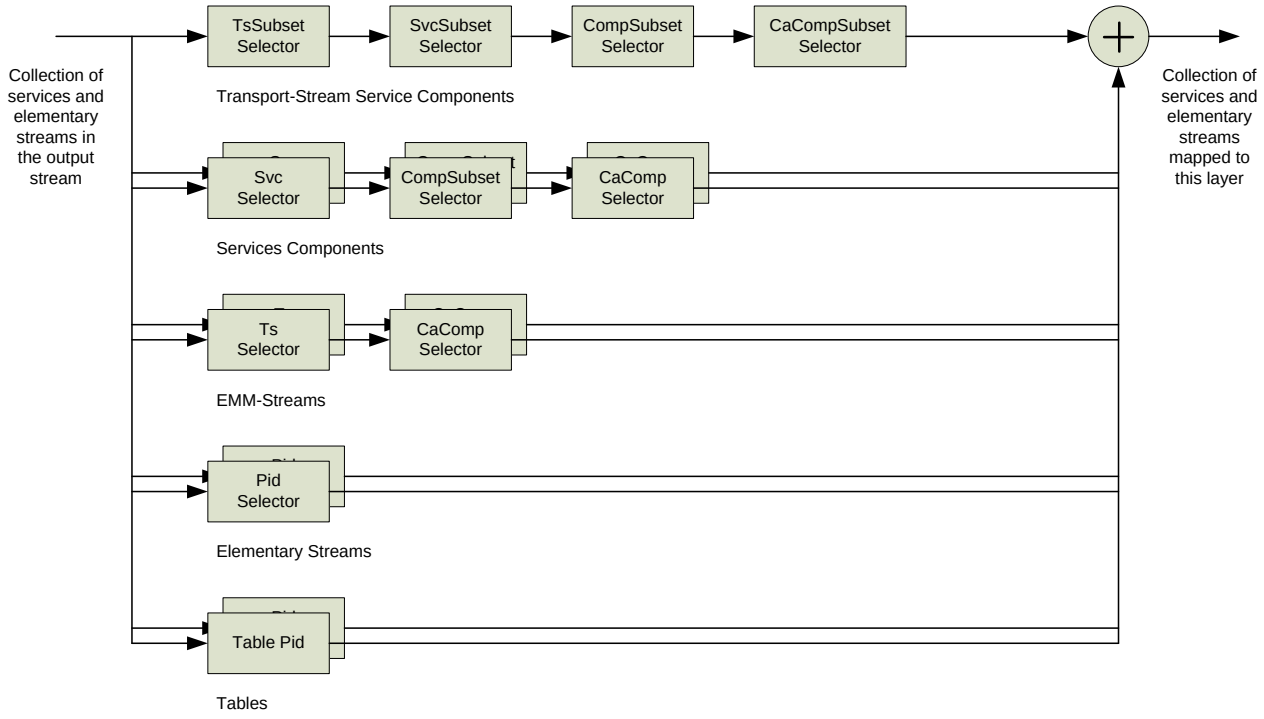


Figure 34 Illustration of selection of items from the transport stream for a hierarchical layer.

3.4.1.1.1 Transport Stream Service Components

TsSvcComponents

Optional element, specifying a collection of services, service components and the ECM-streams that are selected for this hierarchical layer. For this purpose it has four child elements: TsSubsetSelector, SvcSubsetSelector, CompSubsetSelector and CaCompSubsetSelector.

IncludePmt

Type: boolean, Use: optional (Default: "*true*")
Specifies the whether the PMT related to the selected service(s) has to be included in the selection. This can be set to "*true*" or "*false*".

IncludePcr

Type: boolean, Use: optional (Default: "*true*")
Specifies the whether the PCR stream related to the selected service(s) has to be included in the selection. This can be set to "*true*" or "*false*".

TsSvcComponents/TsSubsetSelector

Required element specifying a subset of transport streams. All services in the output transport stream that originate from the selected-transport-streams result in a collection of services. See section 3.6.

TsSvcComponents/TsSubsetSelector

Optional element, this element specifies the services to be excluded from the collection of services. The subset can be specified in positive or negative-selection mode (see section 3.6). When no subset is specified, no services are excluded from the collection.

TsSvcComponents/CompSubsetSelector

Optional element, this element specifies the service components to be excluded from the collection of services. The subset can be specified in positive or negative-selection mode (see section 3.6). When no subset is specified, no service components are excluded from the collection. The context is all services in the collection of services. Therefore the component-group selector has to be used in the component selectors of the component subset selector.

TsSvcComponents/CaCompSubsetSelector

Optional element, this element specifies the CA-components (ECM-streams) to be excluded from the collection of services (including their components). The subset can be specified in positive or negative-selection mode (see section 3.6). When no subset is specified, no CA-components are excluded from the collection.

3.4.1.1.2 Services Components

SvcsComponents

Optional element, specifying the services and their service components and their ECM-streams for this hierarchical layer by means of a sequence of SvcComponents elements.

SvcsComponents/SvcComponents

Optional element specifying a single service and its service components and its ECM-stream that are selected for this hierarchical layer. For this purpose it has three child elements: SvcSelector, CompSubsetSelector and CaCompSubsetSelector.

IncludePmt

Type: boolean, Use: optional (Default: "*true*")
Specifies the whether the PMT related to the selected service has to be included in the selection. This can be set to "*true*" or "*false*".

IncludePcr

Type: boolean, Use: optional (Default: "*true*")
Specifies the whether the PCR stream related to the selected service has to be included in the selection. This can be set to "*true*" or "*false*".

SvcsComponents/SvcComponents/SvcSelector

Required element, specifying selected service. See section 3.5.3.

SvcsComponents/SvcComponents/CompSubsetSelector

Optional element, this element specifies the service components to be excluded from the selected service. The subset can be specified in positive or negative-selection mode (see section 3.6). When no subset is specified, no service components are excluded from the service.

SvcsComponents/SvcComponents/CaCompSubsetSelector

Optional element, this element specifies the CA-components (ECM-streams) to be excluded from the selected service (including its components). The subset can be specified in positive or negative-selection mode (see section 3.6). When no subset is specified, no CA-components are excluded.

3.4.1.1.3 EMM Streams

EmmStreams

Optional elements, specifies the EMM-streams from the output transport stream that are selected for this hierarchical layer by a sequence of EmmsSubset elements.

EmmStreams/EmmsSubSet

Element, specifying EMM-streams by two child-elements: a TsSelector (see 3.5.2) specifying the originating transport-stream that contains the EMM-streams and an optional CaCompSubsetSelector element (see section 3.6) that specifies the CA-components (EMM-streams) to be selected from the transport-stream. When the CaCompSubsetSelector element is absent, all EMM-streams in the transport-stream are selected.

3.4.1.1.4 Elementary Streams

ElemStreams

Optional elements, specifying the elementary-streams from the output transport stream that are selected for this hierarchical layer by a sequence of PidSelectors (see section 3.5.5).

3.4.1.1.5 Tables

Tables

Optional elements, specifying the PSI/SI tables from the output transport stream that are selected for this hierarchical layer by a sequence of TablePid elements.

Tables/TablePid

Optional elements, specifies a single `PID` of one or more PSI/SI tables.

Pid

Type: PidType, Use: required
Specifies the `PID` of the table to be selected

3.4.2 TsTimeOffset

TsTimeOffset

Optional element. This element specifies the time offset that has to be used in case the time in the outgoing tables is not based on UTC-time. The time offset is positive in case the used time in the tables is in advance of the UTC-time. If this element is absent, no time offset is used.

TimeOffset

Type: TimeOffsetType, Use: required
Specifies the time difference between the stream's standard time and UTC-time. The valid range is -24 hours to +24 hours.

For example: when the stream's time is based on JST (Japanese Standard Time) a time offset of +9 hours has to be specified (= "+PT9H").

Below the graphical diagram of the TsTimeOffset element is shown.

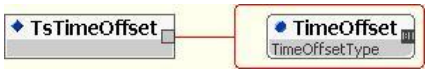


Figure 35. Graphical diagram of TsTimeOffset.

3.4.3 Transport Stream Composition

TsComposition

Optional element, however for a single output this element must be present exactly once in the collection of files that describe the RMC-Data. This element specifies the composition of the outgoing transport-streams and the composition of the services in the transport-stream output. Two levels of composition can be distinguished:

1. Transport-stream wide composition: transport-stream wide selection of services, components EMM-streams and elementary streams. This is specified in the child element TsWideComposition
2. Service wide composition: service wide refinement of the selection of components and elementary streams. This includes dropping of service components and the addition of new components or elementary streams to a service. This is specified in the child element SvcWideCompositions

Below the graphical diagram of the TsComposition element is shown.

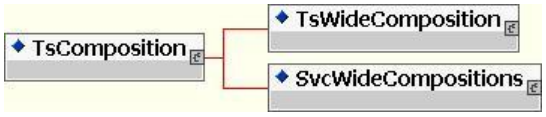


Figure 36. Graphical diagram of TsComposition.

Notes:

Some generic composition behaviour applies for addition of DVB objects (services, components, descriptors). This behaviour is listed below and will not be repeated for each individual DVB object.

If a DVB object is specified to be added to a collection of DVB objects, but the object is not actually present in one of the transport-stream inputs, then this is treated as a no-op and no event message is generated

If a DVB object is specified to be added to a collection of DVB objects, and the object is already present in the collection of DVB objects, then this is treated as a no-op and no event message is generated.

Example:

A transport-stream is selected with log_ts_id=2. A service is added to this transport-stream, using a service selector with selection attributes onw_id=7, ts_id=8 and svc_id=9. This service might already be present in the transport-stream with log_ts_id=2. If this indeed is the case, the MuxXpert ignores the added service.

3.4.3.1 Transport Stream Wide Composition

TsWideComposition

Element; specifying the transport-stream wide selection of services, components, EMM-streams and elementary streams for the outgoing transport-stream. The TsWideComposition has three optional child elements: TsSvcComponents, EmmStreams and ElemStreams.

Below the graphical diagram of the TsWideComposition element is shown.

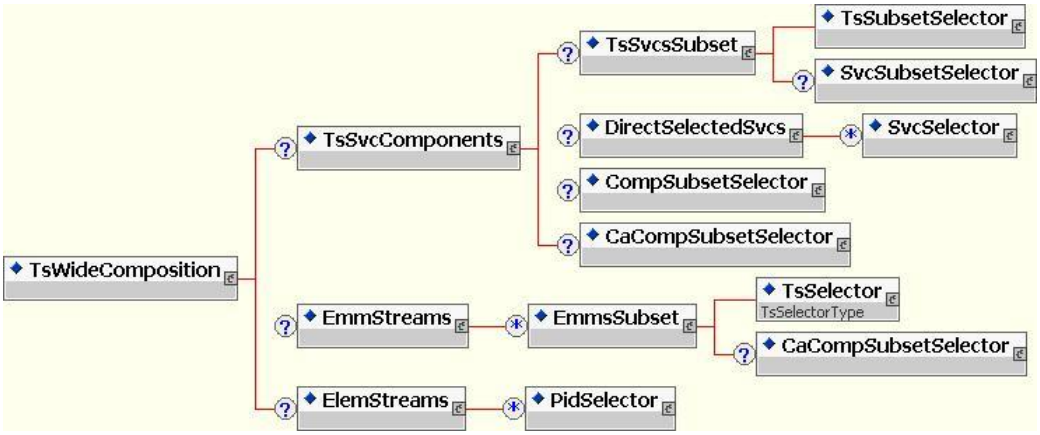


Figure 37. Graphical diagram of TsWideComposition.

Figure 38 illustrates the TS-wide composition. First a transport-stream subset selector is used to get all services from the selected incoming transport-streams, and then a service subset selector is used to drop selected services.

Then, additional services from the transport-stream inputs can be added to the collection by direct service selection. A component subset selector is used to select component groups to be dropped globally over all services in the collection of services. Further a CA-component subset selector is used to select ECM-streams to be dropped globally over all services in the collection of services.

Finally, there is addition of EMM-Streams and of Elementary Streams (unreferenced PIDs).

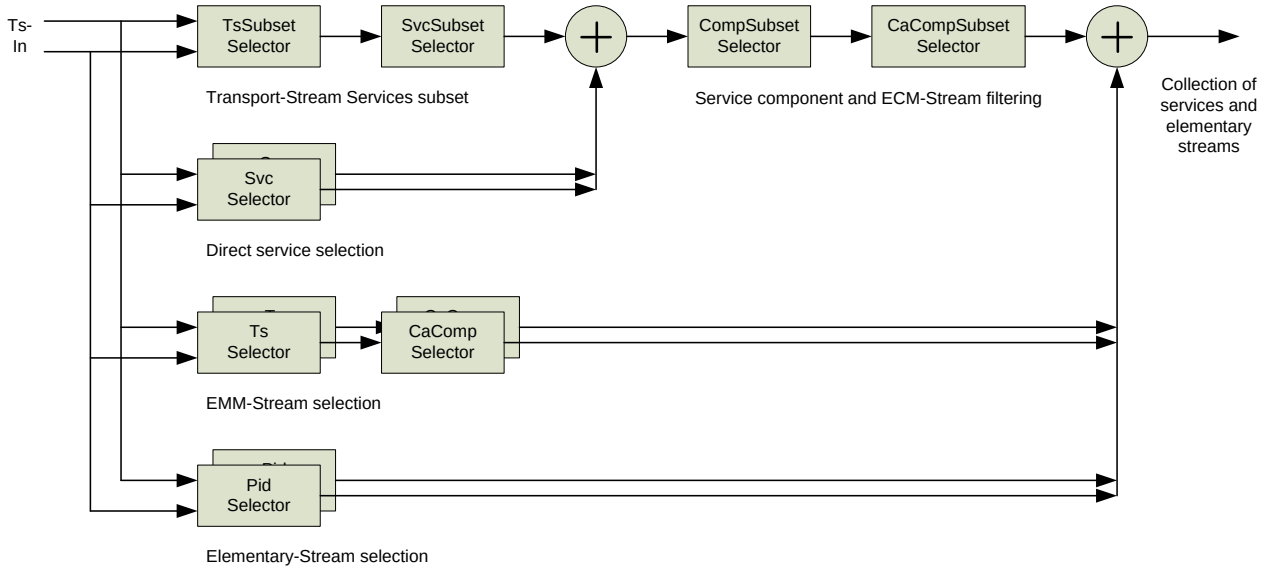


Figure 38 Illustration of transport-stream wide composition.

Notes:

If a transport-stream is composed from multiple incoming transport-streams, and the same service (same <onw_id, ts_id, svc_id>) appears in multiple selected transport-streams, then the service is remultiplexed only once.

When composing a transport-stream output, it is not possible to refer to a transport stream-output to use it as input.

3.4.3.1.1 Transport Stream Service Components

TsSvcComponents

Optional element, specifying the services, service components and the ECM-streams that are selected for the output stream. For this purpose it has four optional child elements: TsSvcsSubset, DirectSelectedSvcs, CompSubsetSelector and CaCompSubsetSelector.

TsSvcComponents/TsSvcsSubset

Optional element, specifying selected services by means of the TsSubsetSelector child element and the optional SvcSubsetSelector child element.
The TsSubsetSelector specifies a subset of transport-streams. All services from the selected-transport-streams result in a collection of services. The transport-streams selection cannot be composed of transport-stream outputs.
The collection of services is refined by the SvcSubsetSelector element. This element specifies the services to be dropped from the collection of services. The subset can be specified in positive or negative-selection mode (see section 3.6). When no service subset is specified, no services are dropped from the collection.

TsSvcComponents/DirectSelectedSvcs

Optional element, specifying selected services by means of a sequence of SvcSelector elements, see section 3.5.3. These services are added to the collection of services.

TsSvcComponents/CompSubsetSelector

Optional element, this element specifies the service components to be dropped from the collection of services. The subset can be specified in positive or negative-selection mode (see section 3.6). When no subset is specified, no service components are dropped from the collection. The operation is TS-wide; the context is all services in the collection of services, instead of a single service.
Therefore the component-single selector cannot be used in the component selectors of the component subset selector.

TsSvcComponents/CaCompSubsetSelector

Optional element, this element specifies the CA-components (ECM-streams) to be dropped from the collection of services (including their components). The subset can be specified in positive or negative-selection mode (see section 3.6). When no subset is specified, no CA-components are dropped from the collection. The operation is TS-wide.

3.4.3.1.2 EMM Streams

EmmStreams

Optional elements, specifies the EMM-streams that are selected for the output stream by a sequence of EmmsSubset elements.

EmmStreams/EmmsSubSet

Specifies EMM-streams by two child-elements: a TsSelector (see 3.5.2) specifying the transport-stream that contains the EMM-streams and an optional CaCompSubsetSelector element (see section 3.6) that specifies the CA-components (EMM-streams) to be selected from the transport-stream. When it is absent, all EMM-streams in the transport-stream are selected.

3.4.3.1.3 Elementary Streams

ElemStreams

Optional elements, specifies the elementary-streams that are selected for the output stream by a sequence of PidSelectors (see section 3.5.5).

3.4.3.2 Service Wide Compositions

SvcWideCompositions

Specifies the service wide compositions for the collection services in the outgoing transport-stream by means of a sequence of SvcWideComposition elements.

SvcWideCompositions/SvcWideComposition

Specifies a service wide refinement of the collection of components and elementary streams that make up one service. This includes the dropping of service components and the addition of new components or elementary streams to a service. The refinement is only allowed for (i) service that is selected already in the transport-stream wide composition and (ii) "new" services that are created from scratch. A new service is composed from "added" elementary streams only.

The collection of components within a service does not change in case no service wide composition is specified for that service. The SvcWideComposition has four child elements: SelectedSvc, ElemStreams, SvcComponents and CaCompSubsetSelector.

Below the graphical diagram of the SvcWideCompositions element is shown.

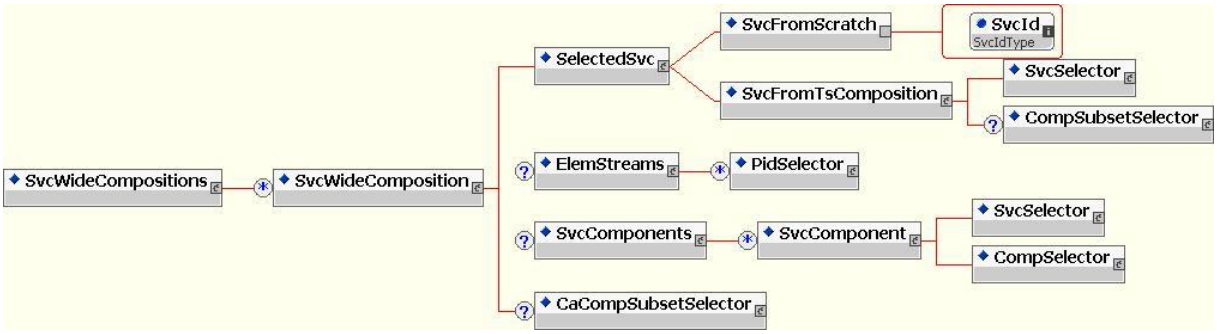


Figure 39. Graphical diagram of SvcWideComposition.

Figure 40 illustrates the service-wide composition of a single service. First (i) the service from the transport-stream wide composition to be refined is selected by a service selector followed by a component subset selector, or (ii) a new service from scratch is selected.

Thereafter, components from other services are added to the selected service. A CA-component subset selector selects the ECM-streams within the selected service to be dropped. Finally, elementary streams are added to the selected service.

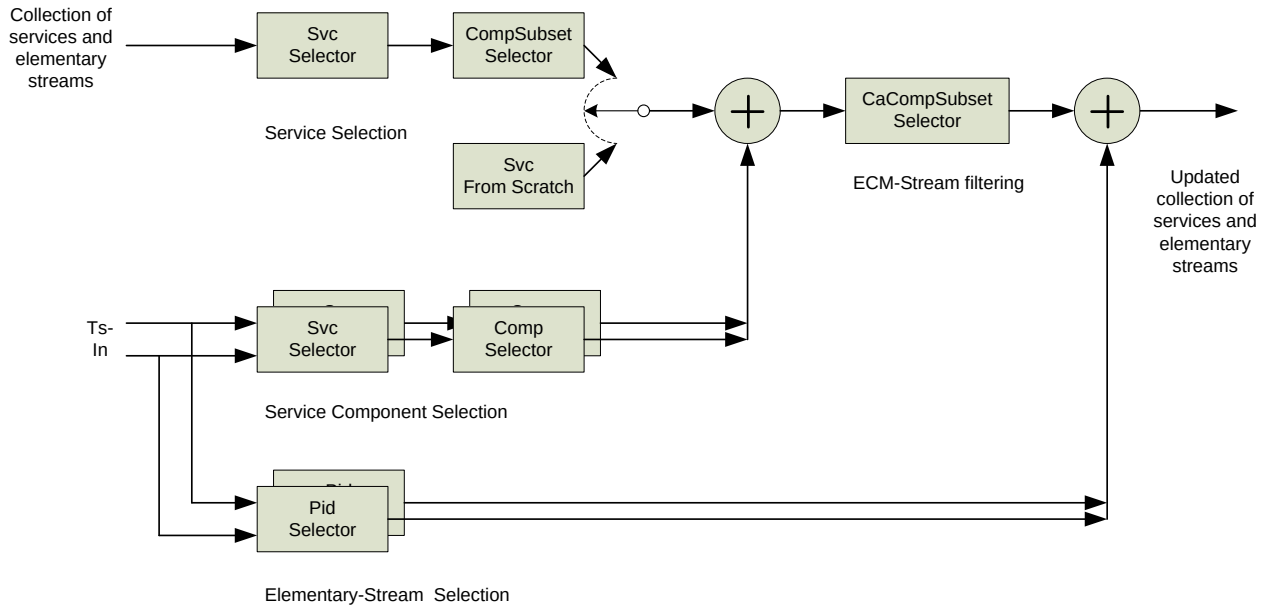


Figure 40 Illustration of service wide composition.

SelectedSvc

Element, specifies the selected service and service components this can be:

1. A new service without components; specified with the SvcFromScratch element.

   SvcId

   Type: SvcIdType, Use: required

   Specifies the `service_id` that identifies the new service.

Or

2. A service and a subset of its components from the collection of services (from the TS-wide composition), specified with the SvcFromTsCompositon element. This element has two child elements: SvcSelector (see section 3.5.3) specifying the selected service and CompSubsetSelector (see section 3.6), specifying the components to be dropped from the service. If no component subset selector is specified, all components are retained.

ElemStreams:

Element; specifies the elementary streams that need to be added to the service, by means of a sequence of PidSelector (each specifying a single PID; see section 3.5.5) elements.

SvcComponents

Element; specifies the service components (including their ECM-streams) to be added to the service, by means of a sequence of SvcComponent elements. The SvcComponent element has two child-elements: SvcSelector and CompSelector elements. This element pair identifies zero, one or more service components that are added to the selected service.

CaCompSubsetSelector

Element; specifies the CA-components (ECM-streams) to be dropped from the services (including their components). The subset can be specified in positive or negative-selection mode (see section 3.6). When no subset is specified, no CA-components are dropped from the collection. The operation is service-wide.

3.4.4 Transport Stream Adaptation

After a transport-stream has been composed, the contents of the outgoing transport-stream (services, components) can be *adapted*:

For *existing* services and components, adaptation can be specified to change fields and descriptors in the PSI and SI describing those services and components.

For *new* services and components, the PSI and SI fields to be inserted for those services and components may be specified.

TS-wide component adaptation can be specified: adaptation that operates on all components in the transport-stream.

And, *service wide component adaptation* can be specified: adaptation that operates on the components in a single service.

Further, *Service Adaptations* can be specified: operates on a single service. The output is the same service, but with adapted (P)SI.

Also the PCR-streams, elementary-streams, EMM- and ECM-streams that are part of the outgoing transport-stream can be *adapted*.

The figure below illustrates the output adaptations. It can be seen as a series of optional adaptors that selects certain objects from the output stream, adapts them and merges them with the output stream again. At the end it outputs an adapted collection of services, service components and elementary streams.

Note that adaptations modify the PSI/SI describing the service components. The contents of the component itself are not modified, with the exception of changing the PID.

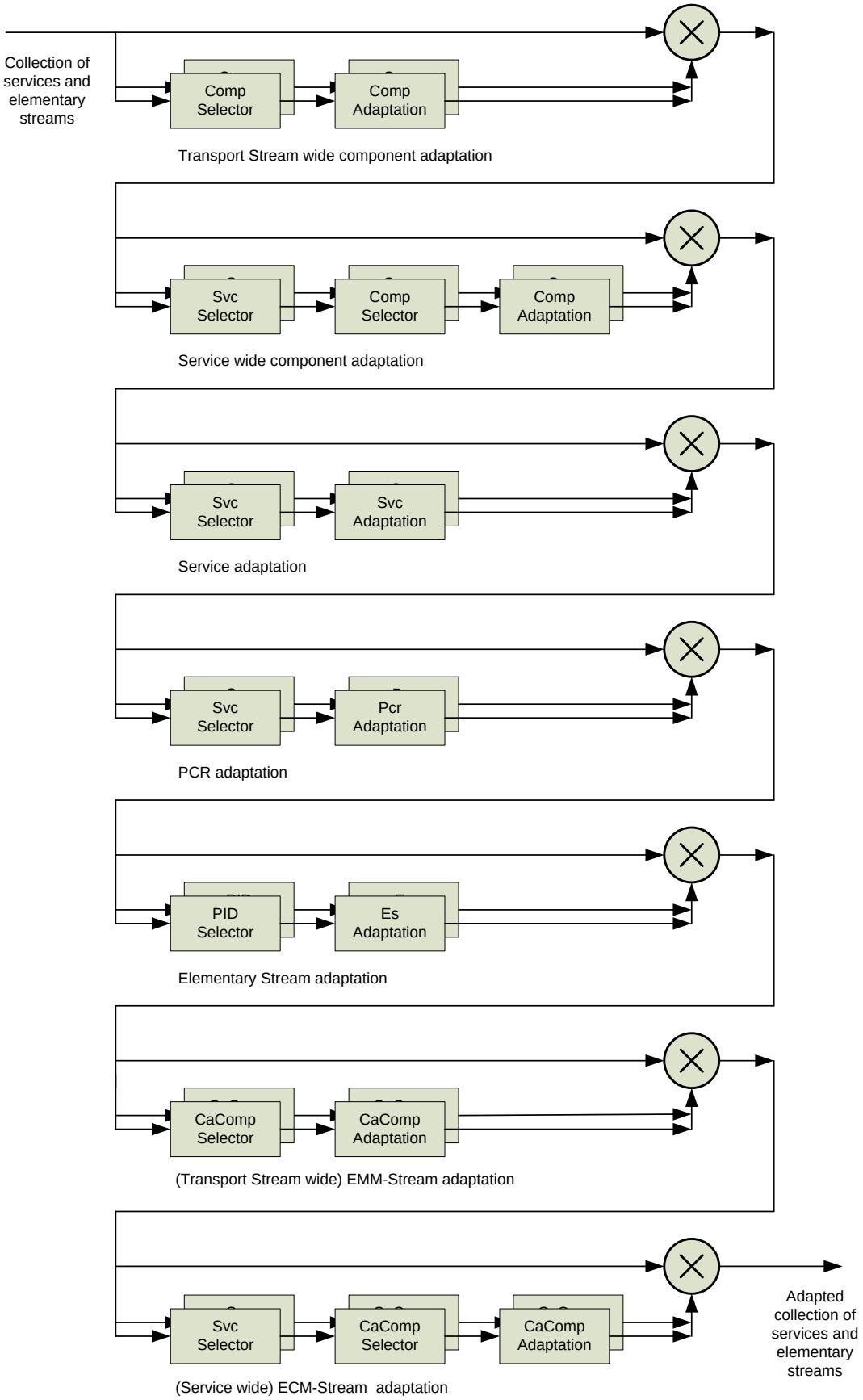


Figure 41. Illustration of adaptation of the output stream.

TsAdaptations

Element; specifying the transport-stream adaptations, it has six child elements: TsWideComponentAdaptations, SvcWideComponentAdaptations, SvcAdaptations, TsWideEsAdaptations, TsWideEmmAdaptations and SvcWideEcmAdaptations.

KeepOrigPid

Type: boolean, Use: optional (Default: "*false*")

Specifies whether the incoming PID values should be kept unchanged if possible. In case the KeepOrigPid is set to "*true*" then the outgoing PID values are equal to the original PID values if the PID values are unique and if the PID value is not overruled by PID-assignments.

Below the graphical diagram of the TsAdaptation element is shown.

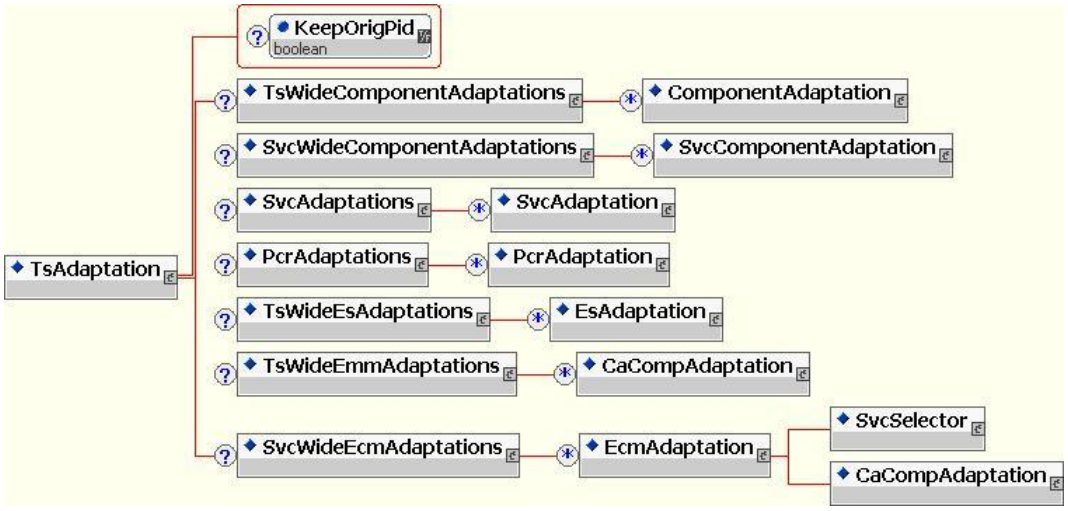


Figure 42. Graphical diagram of TsAdaptation.

TsAdaptations/TsWideComponentAdaptations

Specifies the TS-wide component adaptations by a sequence of ComponentAdaptation child element. See section 3.4.4.1.

TsAdaptations/SvcWideComponentAdaptations

Specifies the service-wide component adaptations by a sequence of SvcComponentAdaptation elements.

The SvcComponentAdaptation on its turn has two child elements:

1. SvcSelector (see section 3.5.3) element specifies the service from which the components will be adapted and
2. ComponentAdaptation that specifies the component adaptation. See section 3.4.4.1.

Note:

The service-wide component adaptation and the transport-stream-wide component adaptation are in principle the same, with the exception that the service-wide component adaptation includes a service selector that specifies the context.

TsAdaptations/SvcAdaptations

Specifies the service adaptations by means of a sequence of SvcAdaptation elements. The SvcAdaptation element specifies the service adaptations of a single service. See section 3.4.4.2.

TsAdaptations/TsWideEsAdaptations

Specifies the TS-wide elementary-stream adaptations by a sequence of EsAdaptation child element. See section **Error! Reference source not found.**.

TsAdaptations/TsWideEmmAdaptations

Specifies the TS-wide CA-component (EMM-stream) adaptations by a sequence of CaCompAdaptation child element. See section 3.4.4.5.

TsAdaptations/SvcWideEcmAdaptations

Specifies the service-wide ECM adaptations by a sequence of EcmAdaptation elements. The EcmAdaptation on its turn has two child elements:

1. SvcSelector (see section 3.5.3) element specifies the service from which the CA-components will be adapted and
2. CaComponentAdaptation that specifies the CA-component (ECM-stream) adaptation. See section 3.4.4.5.

3.4.4.1 Component Adaptation

ComponentAdaptation

Specifies the component adaptations; it operates on a single component or on a group of components. The result is the same component or collection of components, but with adapted PID and/or (P)SI.

If the adaptations operate on a group of components, then the adaptation and the descriptor composition are applied to each individual component in parallel.

The ComponentAdaptation element is used for specifying the component adaptation in the transport-stream-wide and in a service-wide context.

The ComponentAdaptation element has the following child elements: CompSelector, RemoveLevel, NewPid, NewStreamType, NewComponentTag and PmtEsInfoDescriptorComposition.

Below the graphical diagram of the ComponentAdaptation element is shown.

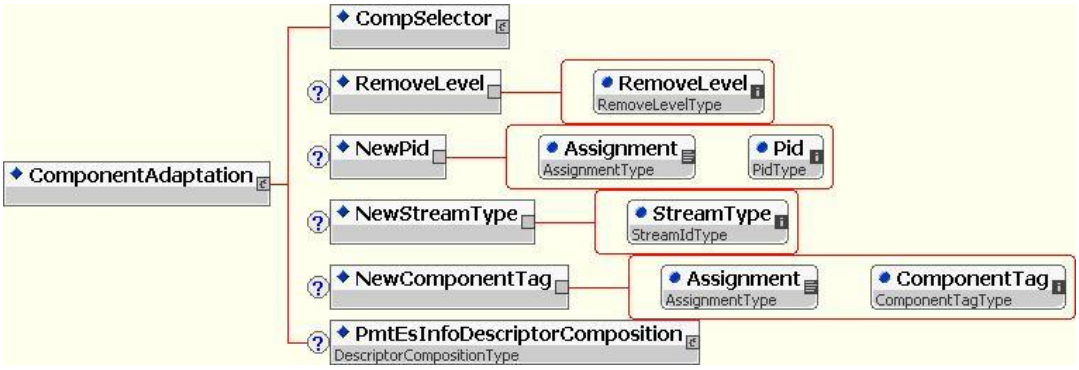


Figure 43. Graphical diagram of ComponentAdaptation.

CompSelector

Required element; specifying a single component or a group of components, selected for adaptation.

ComponentAdaptation/RemoveLevel

Optional element; specifying a (new) value for the `RemoveLevel`. The `RemoveLevel` controls the multiplex behaviour in case the sum of the incoming bit rates exceeds the outgoing bit rate for too long. The `RemoveLevel` can be used to assign a priority to a stream such that "unimportant" streams are removed first. Streams with `RemoveLevel=1` are removed first, streams with `RemoveLevel=4` last and streams whit `RemoveLevel=0` should not be removed at all.

RemoveLevel

Type: RemoveLevelType, Use: required
Specifies a (new) value for the `RemoveLevel`. Valid range 0..4.

ComponentAdaptation/NewPid

Optional element; specifying a (new) PID for the component.

Pid

Type: PidType, Use: required
Specifies the new PID value.

Assignment

Type: AssignmentType, Use: required
Specifies whether the new value is mandatory ("*mandatory*") or preferred ("*preferred*").

Note:

The NewPid element is only allowed in combination with a single component selection

ComponentAdaptation/NewStreamType

Optional element; specifying a (new) StreamType for the component

StreamType

Type: StreamTypeType, Use: required
Specifies a (new) `stream_type` for the selected component or a group of components.

ComponentAdaptation/NewComponentTag

Optional element; specifying a (new) `component_tag` for the component.

ComponentTag

Type: ComponentTagType, Use: required
Specifies the new `component_tag` value.

Assignment

Type: AssignmentType, Use: required
Specifies whether the new value is mandatory ("*mandatory*") or preferred ("*preferred*").

Note:

The NewComponentTag element is only allowed in combination with a single component selection

ComponentAdaptation /PmtEsInfoDescriptorComposition

Optional element; specifying the descriptor-composition operation on the `ES_info` descriptor loop of the component in the PMT. This element is of type *DescriptorCompositionType* see section 3.7.5.

3.4.4.2 Service Adaptation

SvcAdaptation

Element; specifying the service adaptations; it operates on a single service. The output is the same service, but with adapted (P)SI.

Service-adaptation can be used to specify the PCR PID for this service. Then, fields and descriptors in PMT, SDT-actual and EIT-actual can be adapted.

The SvcAdaptation element has the following child elements: SvcSelector, NewSvcId, NewSvcType, NewPcr, NewPmtPid, PmtFirstLoopDescriptorComposition and SdtActualAdaptation.

Below the graphical diagram of the SvcAdaptation element is shown.

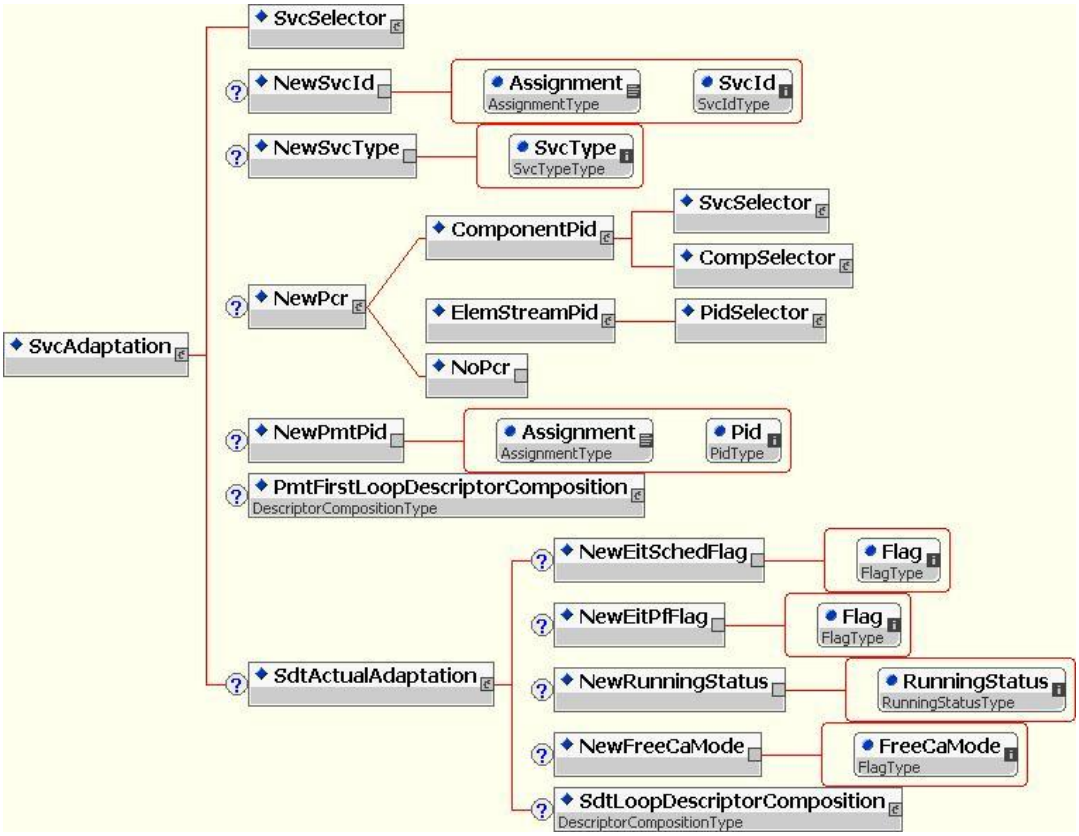


Figure 44. Graphical diagram of SvcAdaptation.

SvcSelector

Required element; (see section 3.5.3) specifying a single service, selected for adaptations.

SvcAdaptation/NewSvcId

Optional element; specifying a (new) `service_id` for the service.

SvcId

Type: SvcIdType, Use: required
Specifies the new the new `service_id` value

Assignment

Type: AssignmentType, Use: required
Specifies whether the new value is mandatory ("*mandatory*") or preferred ("*preferred*").

SvcAdaptation/NewSvcType

Optional element; specifying a (new) service type for the service.

SvcType

Type: SvcTypeType, Use: required
Specifies the new service type

Note:

If a NewSvcType element is specified for a remulitplexed service, then the new service type overrules the service type in the SDT. For construction of the service list descriptor in the NIT and BAT, the new value will be used.

SvcAdaptation/NewPcr

Optional element; however it must be present for new services; specifying a (new) value for the `PCR_PID` field in the PMT of the service. One of the following child elements can be used for specifying the PCR stream:

1. ComponentPid element; specifying a single service component, by means of a service selector (SvcSelector element, see section 3.5.3) followed by a single component selector (CompSelector element, see section 3.5.4).
2. ElemStreamPid element; specifying an elementary stream, by means of an elementary stream selector (PidSelector element to specifying a single PID, see section 3.5.5).
3. NoPcr element; specifying that no PCR stream is associated with the service.

SvcAdaptation/NewPmtPid

Optional element; specifying a (new) PID for the PMT of the service.

Pid

Type: PidType, Use: required
Specifies the new PMT-PID value value.

Assignment

Type: AssignmentType, Use: required
Specifies whether the new value is mandatory ("*mandatory*") or preferred ("*preferred*").

SvcAdaptation/PmtFirstLoopDescriptorComposition

Optional element; specifying the descriptor-composition operation on the first loop in the PMT. This element is of type *DescriptorCompositionType* see section 3.7.5.

SvcAdaptation/SdtActualAdaptation

The SDT-actual contains a number of "flags" that are encoded directly in the SDT-actual, instead of in descriptors. A number of elements are available to set (for new services) or overrule (for remultiplexed services) the value of these flags.
For new services, the Producer must specify these elements (otherwise there is lack of information.), unless the Producer turns off generation of the SDT-actual.
For remultiplexed services, SDT-actual adaptation elements are optional. If no new value has been specified for a certain flag, then the input is followed: the value of the flag is copied to the outgoing SDT-actual.
The SDT adaptations can be specified with the following optional elements:

1. NewEitSchedFlag; for specifying the EIT_schedule_flag in the SDT-actual for the service.

   Flag
   Type: FlagType, Use: required.
   Specifies the EIT_schedule_flag value.

2. NewEitPfFlag; for specifying the EIT_present_following_flag in the SDT-actual for the service.

   Flag
   Type: FlagType, Use: required.
   Specifies the EIT_present_following_flag value.

3. NewRunningStatus; specifying the running_status field in the SDT-actual for the service.

   RunningStatus
   Type: RunningStatusType, Use: required.
   Specifies the running_status field.

4. NewFreeCaMode; specifying the free_CA_mode field in the SDT-actual for the service.

   FreeCaMode
   Type: FlagType, Use: required.
   Specifies the free_CA_mode field.

5. SdtLoopDescriptorComposition; Specifies the descriptor-composition operations on the descriptor loop in the SDT-actual that describes a service. This element is of type *DescriptorCompositionType* see section 3.7.5.

3.4.4.3 PCR Stream Adaptation.

PcrAdaptation

Specifies the PCR stream adaptation; it operates on a single elementary stream. The PcrAdaptation element has the following child elements: SvcSelector, NewPid and RemoveLevel.

Below the graphical diagram of the EsAdaptation element is shown.

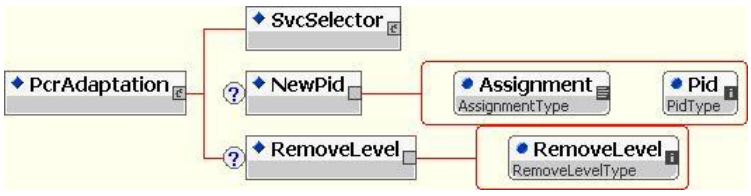


Figure 45. Graphical diagram of PcrAdaptation.

Note:
This adaptation changes the `PID` and/or the `RemoveLevel` value of a PCR stream. For selecting another PCR stream for a service the SvcAdaptation can be used see section 3.4.4.2.

SvcSelector
Required element, specifying selected service where the PCR is used. See section 3.5.3.

PcrAdaptation/NewPid
Optional element; specifying a (new) PID for the PCR stream.

Pid
Type: PidType, Use: required
Specifies the new PID value.

Assignment
Type: AssignmentType, Use: required
Specifies whether the new value is mandatory ("*mandatory*") or preferred ("*preferred*").

PcrAdaptation/RemoveLevel
Optional element; for specifying a (new) value for the `RemoveLevel`. The `RemoveLevel` controls the multiplex behaviour in case the sum of the incoming bit rates exceeds the outgoing bit rate for too long. The `RemoveLevel` can be used to assign a priority to a stream such that "unimportant" streams are removed first. Streams with `RemoveLevel=1` are removed first, streams with `RemoveLevel=4` last and streams whit `RemoveLevel=0` should not be removed at all.

RemoveLevel
Type: RemoveLevelType, Use: required
Specifies a (new) value for the `RemoveLevel`. Valid range 0..4.

3.4.4.4 Elementary Stream Adaptation.

EsAdaptation
Specifies the elementary stream adaptation; it operates on a single unreferenced elementary stream. The EsAdaptation element has the following child elements: PidSelector, NewPid and RemoveLevel.

Below the graphical diagram of the EsAdaptation element is shown.

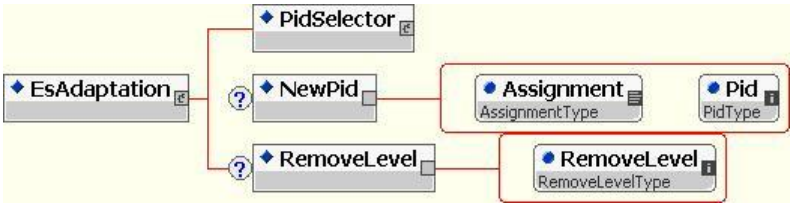


Figure 46. Graphical diagram of EsAdaptation.

PidSelector
Required element; (specifying a single PID; see section 3.5.5) specifying a single elementary-stream that is selected for adaptations.

EsAdaptation/NewPid

Optional element; specifying a (new) PID for the elementary stream.

Pid

Type: PidType, Use: required
Specifies the new PID value.

Assignment

Type: AssignmentType, Use: required
Specifies whether the new value is mandatory ("*mandatory*") or preferred ("*preferred*").

EsAdaptation/RemoveLevel

Optional element; for specifying a (new) value for the `RemoveLevel`. The `RemoveLevel` controls the multiplex behaviour in case the sum of the incoming bit rates exceeds the outgoing bit rate for too long. The `RemoveLevel` can be used to assign a priority to a stream such that "unimportant" streams are removed first. Streams with `RemoveLevel=1` are removed first, streams with `RemoveLevel=4` last and streams whit `RemoveLevel=0` should not be removed at all.

RemoveLevel

Type: RemoveLevelType, Use: required
Specifies a (new) value for the `RemoveLevel`. Valid range 0..4.

3.4.4.5 CA Component Adaptation.

CaCompAdaptation

Element; specifying the CA component adaptation; it operates on a single ECM or EMM stream. The CaCompAdaptation element has the following child elements: CaCompSelector, NewPid and RemoveLevel.

Below the graphical diagram of the CaCompAdaptation element is shown.

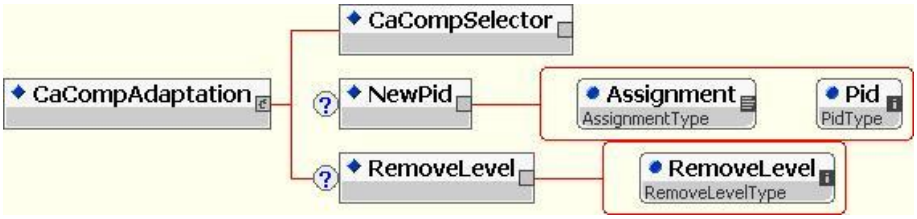


Figure 47. Graphical diagram of CaCompAdaptation.

CaCompSelector

Required element; (see section 3.5.6) specifying a single CA component (EMM or ECM stream), selected for adaptations.

CaCompAdaptation/NewPid

Optional element; specifying a (new) PID for the CA component stream.

Pid

Type: PidType, Use: required
Specifies the new PID value.

Assignment

Type: AssignmentType, Use: required

Specifies whether the new value is mandatory ("*mandatory*") or preferred ("*preferred*").

CaCompAdaptation/RemoveLevel

Optional element; for specifying a (new) value for the `RemoveLevel`. The `RemoveLevel` controls the multiplex behaviour in case the sum of the incoming bit rates exceeds the outgoing bit rate for too long. The `RemoveLevel` can be used to assign a priority to a stream such that "unimportant" streams are removed first. Streams with `RemoveLevel=1` are removed first, streams with `RemoveLevel=4` last and streams whit `RemoveLevel=0` should not be removed at all.

RemoveLevel

Type: RemoveLevelType, Use: required

Specifies a (new) value for the `RemoveLevel`. Valid range 0..4.

3.4.5 Table Assembly

Table *assembly* (= composition + adaptation) specifies: which (P)SI (sub)tables will be included in the output, how are these tables constructed and what adaptations are applied to the incoming tables.

TableAssembly

This element specifies the assembly of the tables for the output transport-stream, it has the following four optional child elements: TsTableParams, MpegTables, DvbSiTables and AtscPsipTables.

Below the graphical diagram of the ComponentAdaptation element is shown.

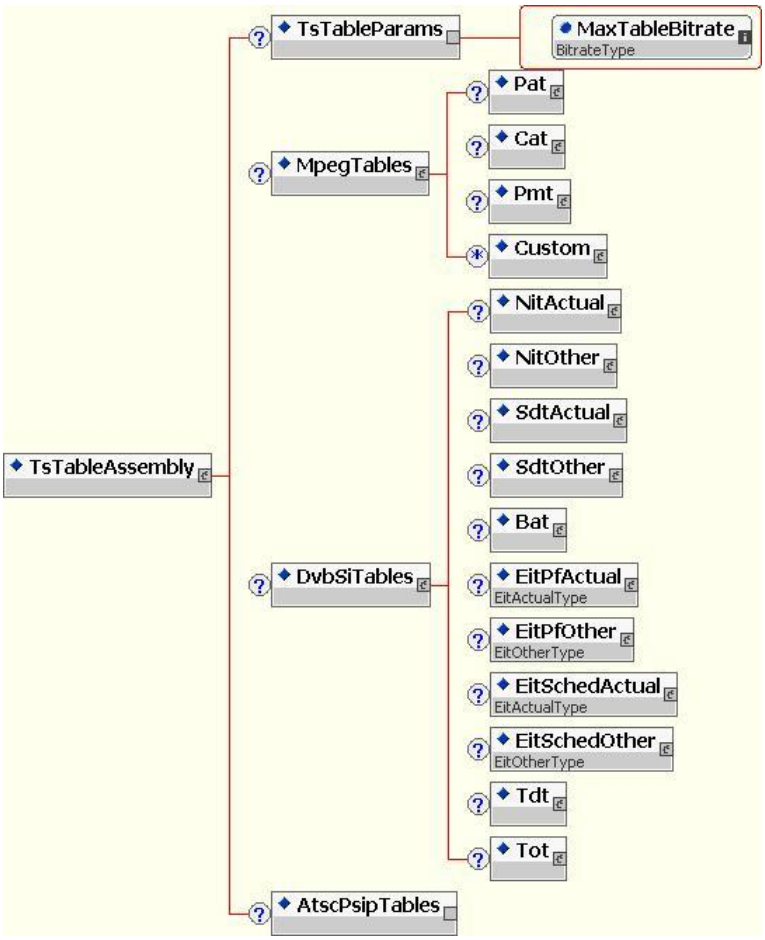


Figure 48. Graphical diagram of TsTableAssembly.

TableAssembly/TsTableParams

Optional element, however for a single output this element must be present exactly once in the collection of files that describe the RMC-Data.

MaxTableBitrate

Type: BitrateType, Use: required

Specifies the maximum total bitrate of the tables in the output transport-stream.

TableAssembly/MpegTables

Specifies the assembly of the MPEG-tables (PAT, CAT and PMT) and the Custom tables. It has the following four optional child elements: Pat, Cat, Pmt and Custom. See the sections below.

TableAssembly/DvbSiTables

Specifies the assembly of the DVB-SI-tables (NIT-actual, NIT-other, BAT, SDT-actual, SDT-other, EIT-p/f-actual, EIT-schedule-actual, EIT-p/f-other EIT-schedule-other, TDT and TOT). It has the following optional child elements: NitActual, NitOther, Bat, SdtActual, SdtOther, EitPfActual, EitSchedActual, EitPfOther, EitSchedOther, Tdt and Tot. See the sections below.

TableAssembly/AtscPsipTables

Now an empty element that can be used in later versions of this ICD for the specification of the ATSC-PSIP assembly.

3.4.5.1 PAT

The PAT (Program Association Table) provides the associatoin between a program_number (svc_id) and the PID of the PMT carrying the program definition [MPEG-2 SYS].

Pat

Optional element; specifying the assembly of the PAT-table.
One of the following three child elements can be used for specifying how the table is constructed: TableNotGenerated, ExternallySuppliedTable or FromTsComposition.
Further the Pat element has two optional child elements: TableCycleProperties and TableDescriptorSorting

Below the graphical diagram of the Pat element is shown.

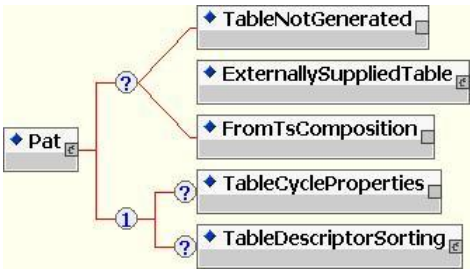


Figure 49. Graphical diagram of Pat.

TableNotGenerated

Empty element, that specifies that this table is not generated at all.

ExternallySuppliedTable

Specifying the table data to be inserted, see section 3.7.1.

FromTsComposition

Empty element, that specifies that this table has to be generated as result from the transport-stream composition and adaptations process.

TableCycleProperties
Element, specifying the table cycle properties (see section 3.7.2). If this element is absent, the default values are used.

TableDescriptorSorting
Element, specifying the order in which descriptors have to appear the table (see section 3.7.3). This element is ignored in combination with the option ExternallySuppliedTable.

3.4.5.2 CAT

The Conditional Access (CA) Table provides the association between one or more CA systems, their EMM streams and any special parameters associated with them [MPEG-2 SYS].

The EMMs inserted in the outgoing transport-stream indirectly specify which CA descriptors have to be inserted in the outgoing CAT. This way, each outgoing EMM stream will get an associated CA descriptor in the CAT.

Cat
Optional element, specifying the assembly of the CAT-table.
One of the following three child elements can be used for specifying how the table is constructed: TableNotGenerated, ExternallySuppliedTable or FromTsComposition.
Further the Cat element has two optional child elements: TableCycleProperties and TableDescriptorSorting

Below the graphical diagram of the Cat element is shown.

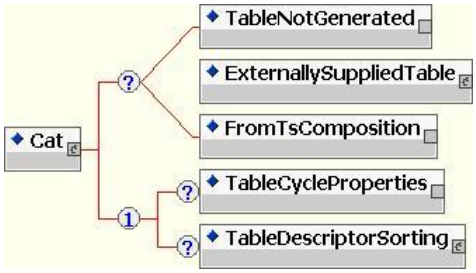


Figure 50. Graphical diagram of Cat.

TableNotGenerated
Empty element, that specifies that this table is not generated at all.

ExternallySuppliedTable
Specifying the table data to be inserted, see section 3.7.1.

FromTsComposition
Empty element, that specifies that this table has to be generated as result from the transport-stream composition and adaptations process.

TableCycleProperties
Element, specifying the table cycle properties (see section 3.7.2). If this element is absent, the default values are used.

TableDescriptorSorting
Element, specifying the order in which descriptors have to appear the table (see section 3.7.3). This element is ignored in combination with the option ExternallySuppliedTable.

3.4.5.3 PMT

The PMT (Program Map Table) provides the mapping between program number (svc_id) and the program elements (components) that comprise the service[2] [MPEG-2 SYS].

Pmt
Optional element, specifying the assembly of the PMT-table.
One of the following three child elements can be used for specifying how the table is constructed: TableNotGenerated, ExternallySuppliedTable or FromTsComposition.
Further the Pmt element has two optional child elements: TableCycleProperties and TableDescriptorSorting

Below the graphical diagram of the Pmt element is shown.

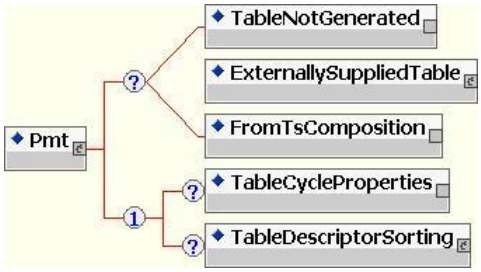


Figure 51. Graphical diagram of Pmt.

TableNotGenerated
Empty element, that specifies that this table is not generated at all.

ExternallySuppliedTable
Specifying the table data to be inserted, see section 3.7.1.

FromTsComposition
Empty element, that specifies that this table has to be generated as result from the transport-stream composition and adaptations process.

TableCycleProperties
Element, specifying the table cycle properties (see section 3.7.2). If this element is absent, the default values are used.

[2] In the MPEG-2 Systems specification, a service is called a program.

TableDescriptorSorting
Element, specifying the order in which descriptors have to appear the table (see section 3.7.3). This element is ignored in combination with the option ExternallySuppliedTable.

3.4.5.4 Custom

Custom tables are data structures that conform to the private-section format[3] [MPEG-2 SYS]. The structure of the section payload of a custom table is privately defined.

Custom
This element specifies the assembly of a custom-table. This element can be present zero, one or more times, specifying different custom tables, that have different table-ids.
One of the following three child elements can be used for specifying how the table is constructed: TableNotGenerated or ExternallySuppliedTable.
Further the Custom element has two optional child elements: TableCycleProperties and TableDescriptorSorting

TableId
Type: TableIdType, Use: required
Specifies the table_id of the custom table.

Below the graphical diagram of the Custom element is shown.

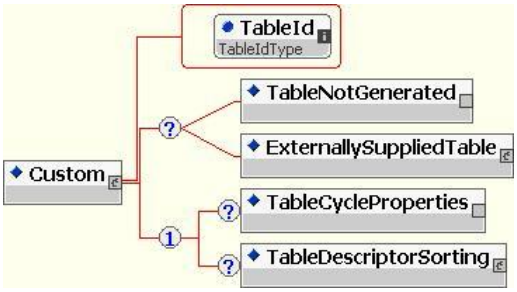


Figure 52. Graphical diagram of Custom.

TableNotGenerated
Empty element, that specifies that this table is not generated at all.

ExternallySuppliedTable
Specifying the table data to be inserted, see section 3.7.1.

TableCycleProperties
Element, specifying the table cycle properties (see section 3.7.2). If this element is absent, the default values are used.

TableDescriptorSorting
Element, specifying the order in which descriptors have to appear the table (see section 3.7.3). This element is ignored in combination with the option ExternallySuppliedTable.

[3] Standard (P)SI tables also conform to the private section format.

3.4.5.5 NIT-actual

The NIT-actual conveys information relating to the physical organization of the transport-streams carried via a given network, and the characteristics of the network itself [DVB-SI]. The NIT-actual is a mandatory DVB-SI table. For DVB-compliancy, the Producer should turn NIT-actual generation on.

NitActual

Optional element; specifying the assembly of the NIT-actual-table.
One of the following three child elements can be used for specifying how the table is constructed: TableNotGenerated, ExternallySuppliedTable or AssembledTable.
Further the NitActual element has two optional child elements: TableCycleProperties and TableDescriptorSorting

Below the graphical diagram of the NitActual element is shown.

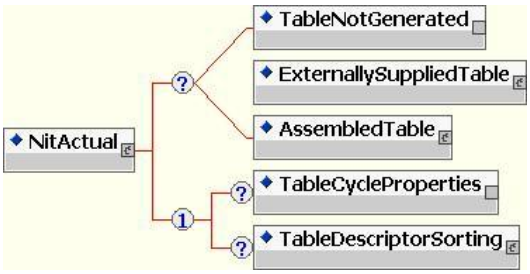


Figure 53. Graphical diagram of NitActual.

TableNotGenerated

Empty element, that specifies that this table is not generated at all.

ExternallySuppliedTable

Specifying the table data to be inserted, see section 3.7.1.

NitActual/AssembledTable

Element, specifying how the NIT-actual is assembled, see section 3.4.5.5.1.

TableCycleProperties

Element, specifying the table cycle properties (see section 3.7.2). If this element is absent, the default values are used.

TableDescriptorSorting

Element, specifying the order in which descriptors have to appear the table (see section 3.7.3). This element is ignored in combination with the option ExternallySuppliedTable.

3.4.5.5.1 NIT-actual Assembled Table

NitActual/AssembledTable

Element, specifying how the NIT-actual is assembled by one of the following two child elements: NitActualFromNitActual, NitActualFromNitOther or NitActualFromScratch.

Below the graphical diagram of the NitActual/AssembledTable element is shown.

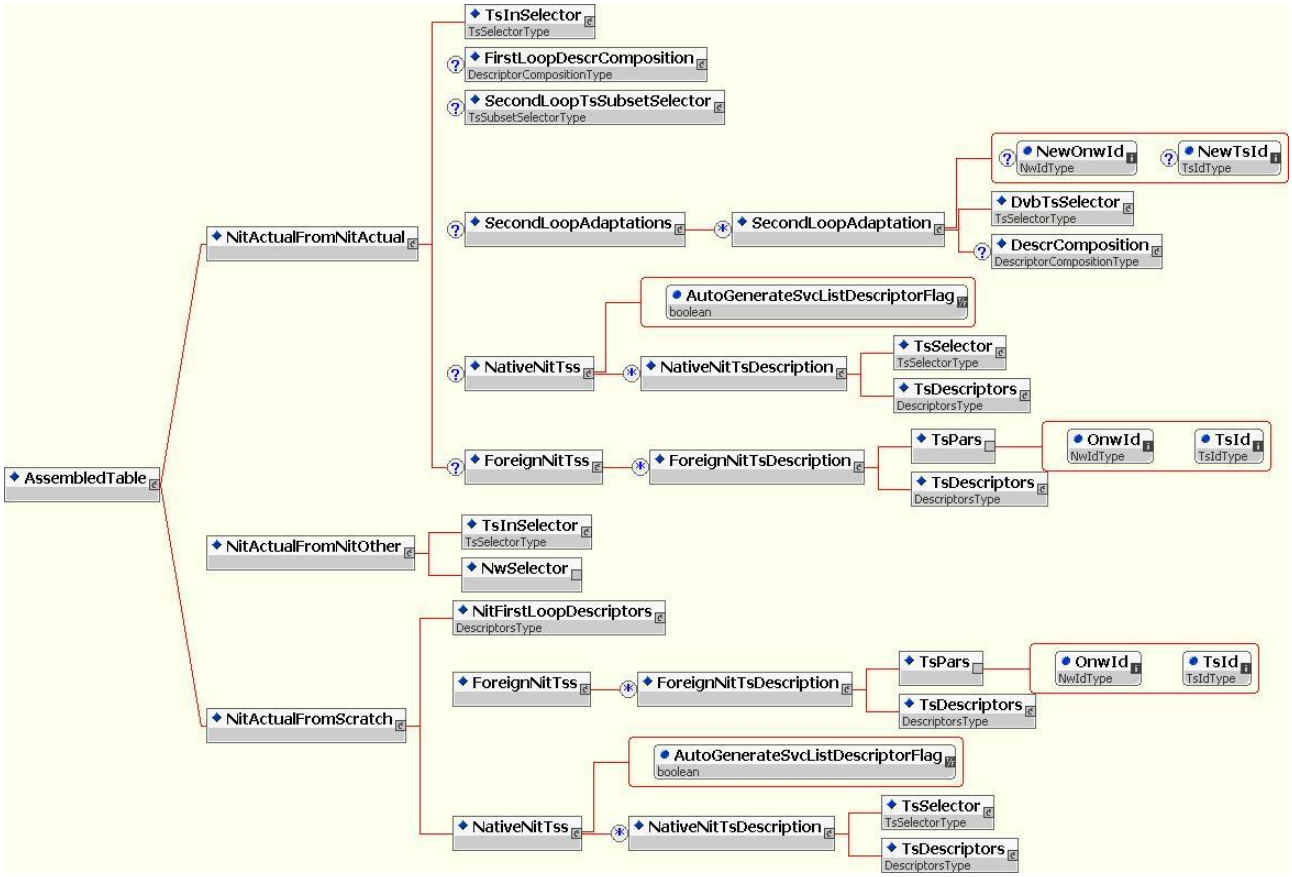


Figure 54. Graphical diagram of NitActual/AssembledTable.

NitActual/AssembledTable/NitActualFromNitActual

Element; specifying the NIT-actual fields, when the NIT-actual is created from an incoming NIT-actual. This is specified with six child elements.

1. The incoming NIT-actual table is selected by a transport-stream selector child element (TsInSelector of type *TsSelectorType* see section 3.5.2), which selects the input from which to take the NIT-actual.
2. The optional element FirstLoopDescrComposition (of type *DescriptorCompositionType* see section 3.7.5), specifies the descriptor-composition operation on NIT 1st loop in the NIT-actual. NIT 1st loop descriptors can be added, removed or replaced.

The optional elements SecondLoopTsSubsetSelector, SecondLoopAdaptations, ForeignNitTss and NativeNitTss specify the NIT 2nd loop descriptors adaptations and new NIT 2nd loops.

3. The optional element SecondLoopTsSubsetSelector (of type *TsSubsetSelectorType* see section 3.6), specifies a subset of transport streams from the incoming NIT-actual that will be included in the second loop in the NIT-actual. Default all transport streams that were in the incoming NIT-actual will be included.
4. The optional element SecondLoopAdaptations contains a sequence of SecondLoopAdaptation elements. The SecondLoopAdaptation element specify the adaptations in one NIT 2nd loop.

NewOnwId
Type: NwIdType, Use: optional
Specifies a new `original_network_id` of the selected transport streams in NIT 2nd loop.

NewTsId
Type: TsIdType, Use: optional
Specifies a new `transport_stream_id` of the selected transport streams in NIT 2nd loop.

The SecondLoopAdaptation element has two child elements:

a. DvbTsSelector required element; a transport-stream selector in mode "DVB Qualified" (type *TsSelectorType* see section 3.5.2) which selects a NIT 2nd loop.

b. DescrComposition optional element (of type *DescriptorCompositionType* see section 3.7.5); specifying the descriptor-composition operation on NIT 2nd loop descriptors in the NIT-actual. NIT 2nd loop descriptors can be added, removed or replaced.

Two child elements are used to specify new transport streams in NIT 2nd loop: ForeignNitTss for the descriptors of the "elsewhere" constructed transport-streams and NativeNitTss for the "here" constructed transport-streams.

5. The optional element ForeignNitTss specifies the NIT 2nd loops for the transport-streams that are constructed "elsewhere". For these transport-streams no a priori information is present. This element contains a sequence of ForeignNitTsDescriptions elements. That specifies the 2nd loop of a single transport-stream. The ForeignNitTsDescription has two child elements:

   a. TsPars element specifies the "elsewhere" constructed transport-stream.

      OnwId
      Type: NwIdType, Use: required
      Specifies the `original_network_id` of the "elsewhere" constructed transport-stream.

      TsId
      Type: TsIdType, Use: required
      Specifies the `transport_stream_id` of the "elsewhere" constructed transport-stream.

   b. TsDescriptors element specifying the NIT 2nd loop descriptors for this transport-stream. This element is of type *DescriptorsType* see section 3.7.6.

6. The optional element NativeNitTss specifies the NIT 2nd loops for the "here" constructed transport-streams (specified via composition and adaptation), the following is known:
   - ts_id and onw_id and
   - the services included in the transport-stream.
   Therefore, a service-list descriptor can be generated automatically.

   AutoGenerateSvcListDescriptorFlag
   Type: boolean, Use: required
   Specifies whether a service-list descriptor must be generated automatically. If "*true*" then, a service-list descriptor for each transport-stream TS-Y that has a network id identical to the network id of outgoing transport-stream must be generated.

   Additional descriptors to be inserted in the NIT 2nd are specified by a sequence of NativeNitTsDescription elements.

The NativeNitTsDescription has two child elements:

a. TsSelector element specifies the "here" constructed transport-stream (TsSelector see section 3.5.3).

b. TsDescriptors element specifying the NIT 2nd loop descriptors for this transport-stream. This element is of type *DescriptorsType* see section 3.7.6.

NitActual/AssembledTable/NitActualFromNitOther

Element; specifying a NIT-other table, and within this table one sub-table. This NIT-other sub-table is converted to a NIT-actual by changing the table-id. The NIT-other sub-table is selected by a transport-stream selector child element (TsInSelector of type *TsSelectorType* see section 3.5.2), which selects the input from which to take the NIT-other, and network selector child element (NwSelector see section 3.5.1) which selects the sub-table to be converted.

NitActual/AssembledTable/NitActualFromScratch

Element; specifying the NIT-actual fields, when the NIT-actual is created from scratch (this is: not from an incoming NIT-other).
The descriptors to be inserted in the NIT 1st loop are specified by the child element:
NitFirstLoopDescriptors element. This element is of type *DescriptorsType* see section 3.7.6.

The Nit 2nd loop describes the transport-streams in the network. Two child elements are used to specify transport streams in NIT 2nd loop: ForeignNitTss for the descriptors of the "elsewhere" constructed transport-streams and NativeNitTss for the "here" constructed transport-streams.

1. The optional element ForeignNitTss specifies the NIT 2nd loops for the transport-streams that are constructed "elsewhere". For these transport-streams no a priori information is present. This element contains a sequence of ForeignNitTsDescriptions elements. That specifies the 2nd loop of a single transport-stream. The ForeignNitTsDescription has two child elements:

   a. TsPars element specifies the "elsewhere" constructed transport-stream.

      OnwId
      Type: NwIdType, Use: required
      Specifies the `original_network_id` of the "elsewhere" constructed transport-stream.

      TsId
      Type: TsIdType, Use: required
      Specifies the `transport_stream_id` of the "elsewhere" constructed transport-stream.

   b. TsDescriptors element specifying the NIT 2nd loop descriptors for this transport-stream. This element is of type *DescriptorsType* see section 3.7.6.

2. The optional element NativeNitTss specifies the NIT 2nd loops for the "here" constructed transport-streams (specified via composition and adaptation), the following is known:
   - ts_id and onw_id and
   - the services included in the transport-stream.
   Therefore, a service-list descriptor can be generated automatically.

   AutoGenerateSvcListDescriptorFlag
   Type: boolean, Use: required

Specifies whether a service-list descriptor must be generated automatically. If "*true*" then, a service-list descriptor for each transport-stream TS-Y that has a network id identical to the network id of outgoing transport-stream must be generated.

Additional descriptors to be inserted in the NIT 2nd are specified by a sequence of NativeNitTsDescription elements.

The NativeNitTsDescription has two child elements:

a. TsSelector element specifies the "here" constructed transport-stream (TsSelector see section 3.5.3).

b. TsDescriptors element specifying the NIT 2nd loop descriptors for this transport-stream. This element is of type *DescriptorsType* see section 3.7.6.

3.4.5.6 NIT-other

The NIT-other is an optional DVB-SI table that describes *other* networks than the network in which the NIT-other is generated.

NitOther

Optional element, specifying the assembly of the NIT-other-table.
One of the following three child elements can be used for specifying how the table is constructed: TableNotGenerated, ExternallySuppliedTable or AssembledTable.
Further the NitOther element has two optional child elements: TableCycleProperties and TableDescriptorSorting

Below the graphical diagram of the NitOther element is shown.

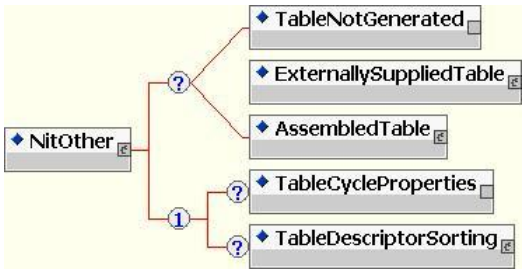


Figure 55. Graphical diagram of NitOther.

TableNotGenerated

Empty element, that specifies that this table is not generated at all.

ExternallySuppliedTable

Specifying the table data to be inserted, see section 3.7.1.

NitOther/AssembledTable

Element; specifying how the NIT-other is assembled, see section 3.4.5.6.1.

TableCycleProperties

Element; specifying the table cycle properties (see section 3.7.2). If this element is absent, the default values are used.

TableDescriptorSorting

Element; specifying the order in which descriptors have to appear the table (see section 3.7.3). This element is ignored in combination with the option ExternallySuppliedTable.

3.4.5.6.1 NIT-other Assembled Table

NitOther/AssembledTable

Element; specifying how the NIT-other is assembled per sub-table. It contains a sequence of NitOtherSubTable elements.

Below the graphical diagram of the NitOther/AssembledTable element is shown.

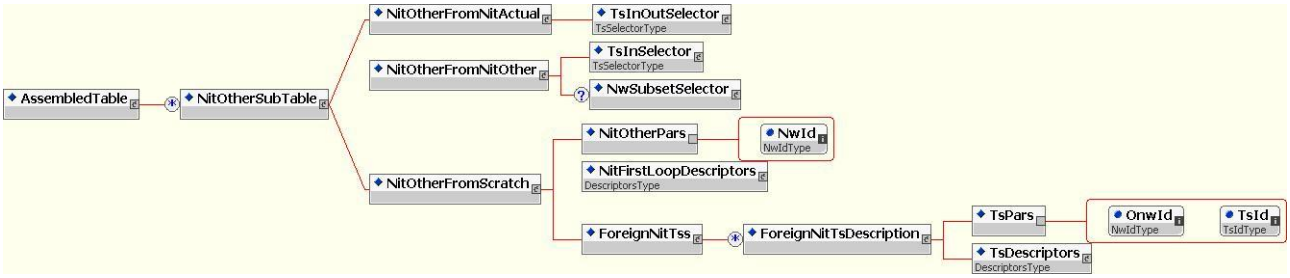


Figure 56. Graphical diagram of NitOther/AssembledTable.

NitOther/AssembledTable/NitOtherSubTable

Element; specifying how NIT-other sub-tables are assembled by one of the following three child elements: NitOtherFromNitActual, NitOtherFromNitOther or NitOtherFromScratch.

NitOther/AssembledTable/NitOtherSubTable/NitOtherFromNitActual

Element specifying a NIT-actual table that is taken from an input of output stream. This NIT-actual table is converted to one NIT-other sub-table by changing the table-id. The NIT-actual table to be converted is selected by a transport-stream selector child element (TsInOutSelector of type *TsSelectorType* see section 3.5.2)

NitOther/AssembledTable/NitOtherSubTable/NitOtherFromNitOther

Element specifying zero, one or more NIT-other sub-tables to be included in the output transport-stream. The NIT-other sub-tables are selected by a transport-stream selector child element (TsInSelector of type TsSelectorType see section 3.5.2), which selects the input from which to take the NIT-other table, and a network subset selector child element (NwSubsetSelector see section 3.6) which selects the NIT-other sub-tables to be included.

NitOther/AssembledTable/NitOtherSubTable/NitOtherFromScratch

Element, specifying the NIT-other fields, when one NIT-other sub-table is created from scratch. This is specified with three child elements:

1. NitOtherPars; specifying the `network_id` of the NIT-other sub-table.

   NwId

   Type: NwIdType, Use: required
   Specifies the `network_id` of of the NIT-other sub-table.

2. NitFirstLoopDescriptors; specifying the descriptors to be inserted in the NIT 1st loop. This el-

ement is of type *DescriptorsType* see section 3.7.6.

3. ForeignNitTss; specifying the NIT 2nd loop descriptors by a sequence of ForeignNitTsDescription elements. The ForeignTsDescription has two child elements:
   a. TsPars element specifies the "elsewhere" constructed transport-stream.

      OnwId
      Type: NwIdType, Use: required
      Specifies the `original_network_id` of the "elsewhere" constructed transport-stream.

      TsId
      Type: TsIdType, Use: required
      Specifies the `transport_stream_id` of the "elsewhere" constructed transport-stream.

   b. TsDescriptors element specifying the NIT 2nd loop descriptors for this transport-stream. This element is of type *DescriptorsType* see section 3.7.6.

3.4.5.7 SDT-actual

The SDT-actual is a mandatory DVB-SI table that describes the services contained within a particular[4] transport-stream [DVB-SI]. For DVB-compliancy, the Producer should turn SDT-actual generation on.

SdtActual
Optional element, specifying the assembly of the SDT-actual-table.
One of the following four child elements can be used for specifying how the table is constructed: TableNotGenerated, ExternallySuppliedTable , FromTsComposition or AssembledTable.
Further the SdtActual element has two optional child elements: TableCycleProperties and TableDescriptorSorting

Below the graphical diagram of the SdtActual element is shown.

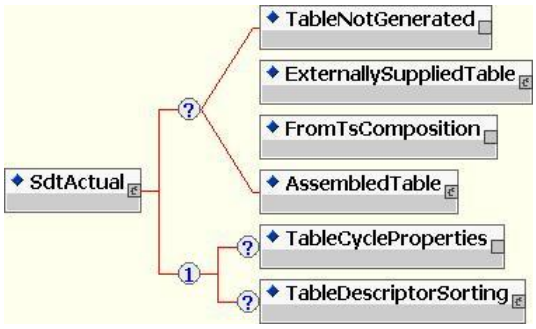


Figure 57. Graphical diagram of SdtActual.

TableNotGenerated
Empty element, that specifies that this table is not generated at all.

ExternallySuppliedTable
Specifying the table data to be inserted, see section 3.7.1.

[4] This definition is taken unmodified from DVB. In fact, *particular* refers to the transport stream in which the SDT-actual is contained.

FromTsComposition
Empty element, that specifies that this table has to be generated as result from the transport-stream composition and adaptations process.

SdtActual/AssembledTable
Element, specifying how the SDT-actual is assembled, see section 3.4.5.7.1.

TableCycleProperties
Element, specifying the table cycle properties (see section 3.7.2). If this element is absent, the default values are used.

TableDescriptorSorting
Element, specifying the order in which descriptors have to appear the table (see section 3.7.3). This element is ignored in combination with the option ExternallySuppliedTable.

3.4.5.7.1 SDT-actual Assembled Table

SdtActual/AssembledTable
Element, specifying how the SDT-actual is assembled it contains the following child element: SdtActualFromSdtOther.

Below the graphical diagram of the SdtActual/AssembledTable element is shown.

Figure 58. Graphical diagram of SdtActual/AssembledTable.

SdtActual/AssembledTable/SdtActualFromSdtOther
Element specifying a SDT-other table, and within this table one sub-table. This SDT-other sub-table is converted to a SDT-actual by changing the table-id. The SDT-other sub-table is selected by a transport-stream selector child element (TsInSelector of type *TsSelectorType* see section 3.5.2), which selects the input from which to take the SDT-other, and another transport-stream selector child element in mode "DVB Qualified" (DvbTsSelector of type *TsSelectorType* see section 3.5.2) which selects the sub-table to be converted.

3.4.5.8 SDT-other

The SDT-other is an optional DVB-SI tables that describes the services contained in other transport-streams than the transport in which the SDT-other is generated [DVB-SI].

SdtOther
Optional element, specifying the assembly of the SDT-other-table.
One of the following three child elements can be used for specifying how the table is constructed: TableNotGenerated, ExternallySuppliedTable or AssembledTable.
Further the SdtOther element has two optional child elements: TableCycleProperties and TableDescriptorSorting.

Below the graphical diagram of the SdtOther element is shown.

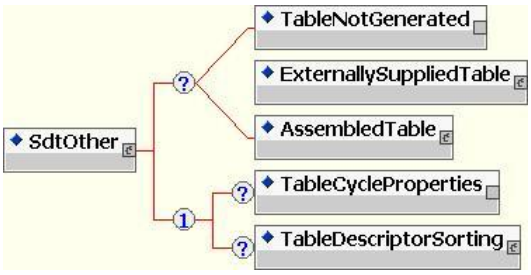


Figure 59. Graphical diagram of SdtOther

TableNotGenerated
Empty element, that specifies that this table is not generated at all.

ExternallySuppliedTable
Specifying the table data to be inserted, see section 3.7.1.

SdtOther/AssembledTable
Element, specifying how the SDT-other is assembled, see section 3.4.5.8.1.

TableCycleProperties
Element, specifying the table cycle properties (see section 3.7.2). If this element is absent, the default values are used.

TableDescriptorSorting
Element, specifying the order in which descriptors have to appear the table (see section 3.7.3). This element is ignored in combination with the option ExternallySuppliedTable.

3.4.5.8.1 SDT-other Assembled Table

SdtOther/AssembledTable
Element, specifying how the SDT-other sub-tables are assembled. It contains a sequence of SdtOtherSubTable elements.

Below the graphical diagram of the SdtOther/AssembledTable element is shown.

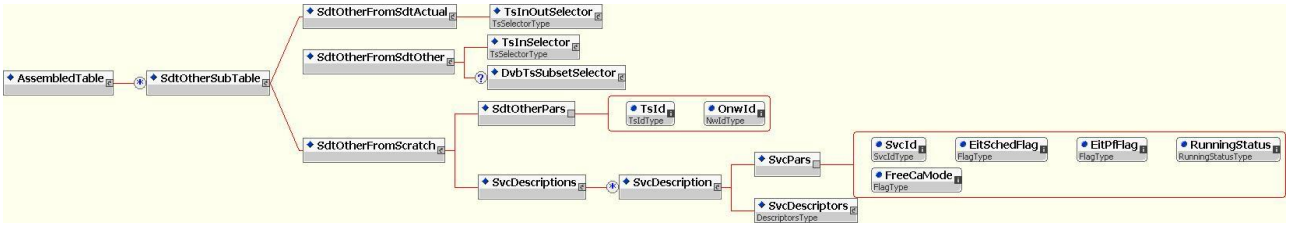


Figure 60. Graphical diagram of SdtOther/AssembledTable

SdtOther/AssembledTable/SdtOtherSubTable
Element, specifying how SDT-other sub-tables are assembled by one of the following three child elements: SdtOtherFromSdtActual, SdtOtherFromSdtOther or SdtOtherFromScratch.

SdtOther/AssembledTable/SdtOtherSubTable/SdtOtherFromSdtActual

Element specifying a SDT-actual table taken from an input or output stream. This SDT-actual table is converted to one SDT-other sub-table by changing the table-id. The SDT-actual table to be converted is selected by a transport-stream selector child element (TsInOutSelector of TsSelector type see section 3.5.2).

SdtOther/AssembledTable/SdtOtherSubTable/SdtOtherFromSdtOther

Element specifying zero, one or more SDT-other sub-tables to be included in the output transport-stream. The SDT-other sub-tables are selected by a transport-stream selector child element (TsInSelector of type *TsSelectorType* see section 3.5.2), which selects the input from which to take the SDT-other table, and transport-stream subset selector child element in mode "DVB Qualified" (DvbTsSubsetSelector see section 3.6) which selects the SDT-other sub-tables to be included.

SdtOther/AssembledTable/SdtOtherSubTable/SdtOtherFromScratch

Element, specifying the SDT-other fields, when one SDT-other sub-table is created from scratch. This specified with two child elements:

1. SdtOtherPars; specifying the `transport_stream_id` and the `original_network_id` of the SDT-other sub-table.

   TsId
   Type: TsIdType, Use: required
   Specifies the `transport_stream_id` of of the SDT-other sub-table.

   OnwId
   Type: NwIdType, Use: required
   Specifies the `original_network_id` of of the SDT-other sub-table.

2. SvcDescriptions; this element specifies the SDT loop descriptors for the services in a transport-stream. It contains a sequence of SvcDescription elements. The SvcDescription element specifies the loop descriptors of a single service. The SvcDescription has two child elements:

   a. SvcPars element, that specifies the service and SDT "flags" related to this service in the SDT loop.

      SvcId
      Type: SvcIdType, Use: required
      Specifies the `service_id`

      EitSchedFlag
      Type: FlagType, Use: required
      Specifies the EIT_schedule_flag value.

      EitPfFlag
      Type: FlagType, Use: required.
      Specifies the EIT_present_following_flag value.

      RunningStatus
      Type: RunningStatusType, Use: required.
      Specifies the running_status field.

      FreeCaMode
      Type: FlagType, Use: required.
      Specifies the EIT_schedule_flag value.

b. SvcDescriptors element specifying the SDT loop descriptors for this service. This element is of type *DescriptorsType* see section 3.7.6.

3.4.5.9 BAT

The BAT is an optional DVB-SI table that provides information regarding bouquets. A bouquet is a collection of services, which may traverse the boundary of a network [DVB-SI].

For each outgoing transport-stream, the Producer specifies the bouquets to be inserted by specifying the bouquet attributes and the services in the bouquet. Specifying remultiplexing of the BAT is not possible.

Bat

Optional element, specifying the assembly of the BAT-table.
One of the following three child elements can be used for specifying how the table is constructed: TableNotGenerated, ExternallySuppliedTable or AssembledTable.
Further the Bat element has two optional child elements: TableCycleProperties and TableDescriptorSorting.

Below the graphical diagram of the Bat element is shown.

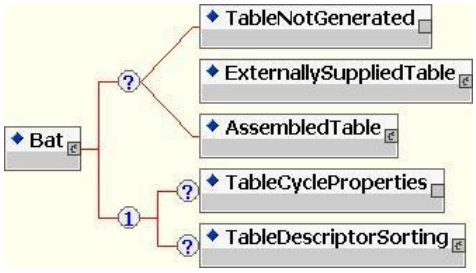


Figure 61. Graphical diagram of Bat.

TableNotGenerated

Empty element, that specifies that this table is not generated at all.

ExternallySuppliedTable

Specifying the table data to be inserted, see section 3.7.1.

Bat/AssembledTable

Element, specifying how the BAT is assembled, see section 3.4.5.9.1.

TableCycleProperties

Element, specifying the table cycle properties (see section 3.7.2). If this element is absent, the default values are used.

TableDescriptorSorting

Element, specifying the order in which descriptors have to appear the table (see section 3.7.3). This element is ignored in combination with the option ExternallySuppliedTable.

3.4.5.9.1 BAT Assembled Table

Bat/AssembledTable

Element, specifying how the BAT sub-tables are assembled. It contains a sequence of BatSubTable elements.

Below the graphical diagram of the Bat/AssembledTable element is shown.

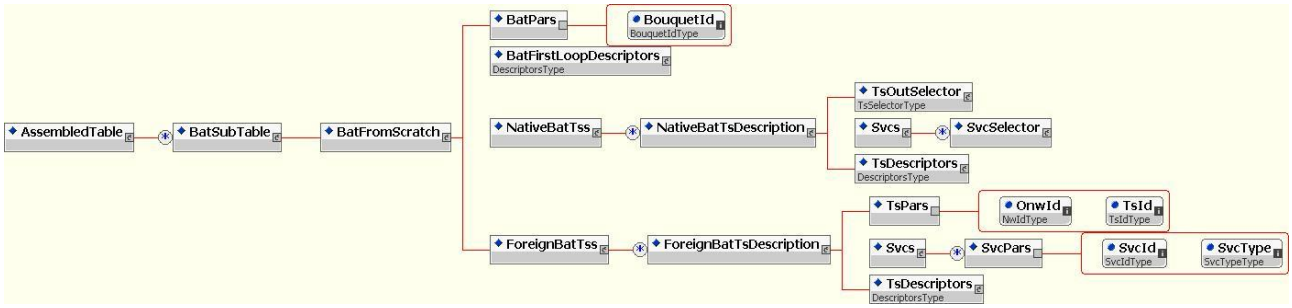


Figure 62. Graphical diagram of Bat/AssembledTable.

Bat/AssembledTable/BatSubTable

Element, specifying how one BAT sub-table is assembled. Only BAT-sub-table creation from scratch is supported, as specified by the following child element: BatFromScratch.

Bat/AssembledTable/BatSubTable/BatFromScratch

Element, specifying the BAT fields, when the BAT sub-table is created from scratch. This is specified with the following child elements: BatPars, BatFirstLoopDescriptors and for specifying the BAT 2nd loop descriptors: NativeBatTss and ForeignBatTss.

Bat/AssembledTable/BatSubTable/BatFromScratch/BatPars

Element, specifying the `bouquet_id` of the BAT sub-table.

BouquetId

Type: BouqetIdType, Use: required

Specifies the `bouquet_id` of the BAT sub-table.

Bat/AssembledTable/BatSubTable/BatFromScratch/BatFirstLoopDescriptors

Element, specifying the descriptors to be inserted in the BAT 1st loop. This element is of type *DescriptorsType* see section 3.7.6.

Bat/AssembledTable/BatSubTable/BatFromScratch/NativeBatTss

This element specifies the BAT 2nd loop descriptors for the services in transport-streams that are constructed "here". This element contains a sequence of NativeBatTsDescription elements. That specifies the 2nd loop descriptors of a single transport-stream.

The NativeBatTsDescripton element has the following child elements:

1. TsOutSelector element specifying the "here" constructed transport-stream, see section 3.5.2.

2. Svcs that specifies services in this bouquet from the "here" constructed transport-stream by a sequence of service selectors (SvcSelector elements, see section 3.5.3).
3. TsDescriptors specifying the additional BAT 2nd loop descriptors for this transport-stream. This element is of type *DescriptorsType* see section 3.7.6.

Bat/AssembledTable/BatSubTable/BatFromScratch/ForeignBatTss

This element specifies the BAT 2nd loop descriptors for the services in transport-streams that are constructed "elsewhere". For these services no a priori information is present. This element contains a sequence of ForeignBatTsDescription elements. That specifies the 2nd loop descriptors of a single transport-stream.
The ForeignBatTsDescriptionelement has the following child elements:

1. TsPars element specifying the "elsewhere" constructed transport-stream.

   OnwId
   Type: NwIdType, Use: required
   Specifies the `original_network_id` of of the BAT sub-table.

   TsId
   Type: TsIdType, Use: required
   Specifies the `transport_stream_id` of of the BAT sub-table.

2. Svcs that specifies the services in this bouquet from the "elsewhere" constructed transport-stream by a sequence of SvcPars elements containing the SvcId and SvcType attributes. The corresponding service-list descriptors have to be generated automatically.

   SvcId
   Type: SvcIdType, Use: required
   Specifies one `service_id` for the service-list descriptor in the BAT sub-table.

   SvcType
   Type: SvcTypeType, Use: required
   Specifies the `service_type` of the related service for the service-list descriptor in the BAT sub-table.

3. TsDescriptors element, specifying the additional BAT 2nd loop descriptors for this transport-stream. This element is of type *DescriptorsType* see section 3.7.6.

3.4.5.10 EIT-P/F-actual

The EIT-p/f-actual describes the present and the following event. The EIT-p/f-actual is a mandatory DVB-SI table. For DVB-compliancy, the Producer should turn EIT-p/f-actual generation on.

EitpfActual

Optional element, specifying the assembly of the EIT-p/f-actuaTable. This element is of the type *EitActualType* that defines the EIT-p/f-actual and the EIT-schedule-actual tables.

3.4.5.11 EIT-Schedule-actual

The EIT-schedule-actual describes the future events. The EIT-schedule-actual is optional.

EitSchedActual

Optional element, specifying the assembly of the EIT-schedule-actual. This element is of the type *EitActualType* that defines the EIT-p/f-actual and the EIT-schedule-actual tables.

3.4.5.12 EIT-actual Type

The Eit-actual type is used for the specification of the EIT-p/f-actual and the EIT-schedule-actual table assembly.

EitActualType

An element of this type specifies the assembly of the EIT-p/f-actual or the EIT-schedule-actual. One of the following three child elements can be used for specifying how the table is constructed: TableNotGenerated, ExternallySuppliedTable or AssembledTable.
Further it has two optional child elements: TableCycleProperties and TableDescriptorSorting.

Below the graphical diagram of the *EitActualType* is shown.

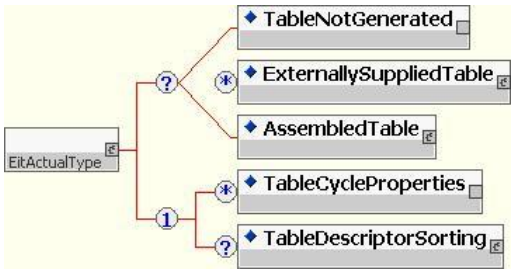


Figure 63. Graphical diagram of *EitActualType*.

TableNotGenerated

Empty element, that specifies that this table is not generated at all.

ExternallySuppliedTable

Specifying the table data to be inserted, see section 3.7.1. In behalf of the EIT- schedule-actual tables this can be a sequence of ExternallySuppliedTable elements, to allow multiple EIT sub-tables with different table-ids.

EitActualType/AssembledTable

Element, specifying how the EIT-actual is assembled, see section 3.4.5.12.1.

TableCycleProperties

Element, specifying the table cycle properties (see section 3.7.2). If this element is absent, the default values are used. In behalf of the EIT-schedule-actual tables this can be a sequence of TableCycleProperties elements, to allow different cycle properties for EIT sub-tables with different table-ids.

TableDescriptorSorting

Element, specifying the order in which descriptors have to appear the table (see section 3.7.3). This element is ignored in combination with the option ExternallySuppliedTable.

3.4.5.12.1 EIT-actual Type Assembled Table

EitActualType/AssembledTable

Element, specifying how the EIT-actual sub-tables are assembled. It contains a sequence of EitActualSubTable elements and it contains the DefaultEitActualSubTable element and optionally the EitTimeFilter element.

Below the graphical diagram of the *EitActualType/AssembledTable* is shown.

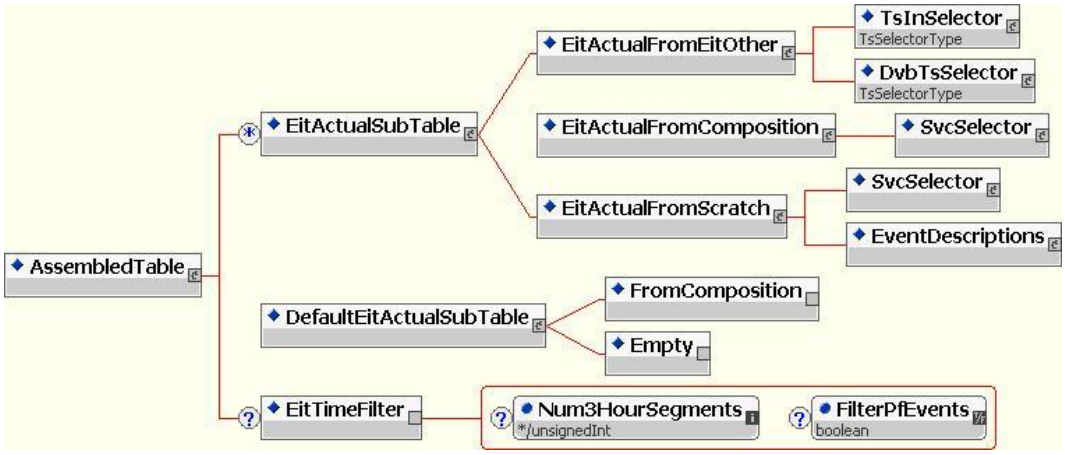


Figure 64. Graphical diagram of *EitActualType/AssembledTable*.

EitActualType/AssembledTable/EitActualSubTable

Element, specifying how one or more EIT-actual sub-tables are assembled by one of the following three child elements:

1. EitActualFromEitOther: Element specifying an EIT-other table, and within this table one or more sub-tables. These EIT-other sub-tables are converted to an EIT-actual by changing the table-id. The EIT-other sub-tables are selected by a transport-stream selector child element (TsInSelector of type *TsSelectorType* see section 3.5.2), which selects the input from which to take the EIT-other, and another transport-stream selector child element in mode "DVB Qualified" element (DvbTsSelector of type *TsSelectorType* see section 3.5.2) which selects the sub-tables to be converted.
2. EitActualFromComposition: Element specifying one EIT-sub-table that has to be generated as result from the transport-stream composition and adaptations process. The service selector (SvcSelector element, see section 3.5.3) selects the service for which the EIT-sub-table is generated from composition.
3. EitActualFromScratch: Element, specifying the EIT-actual fields, when one EIT-actual sub-table is created from scratch. This is specified with two child elements:
   a. SvcSelector (see section 3.5.3); specifying the service for which the EIT-sub-table is created.
   b. EventDescriptions; specifying the EIT loop descriptors by a sequence of Event elements (see section 3.7.4).

EitActualType/AssembledTable/DefaultEitActualSubTable

Element, specifying the default EIT-actual sub-table, it contains one of the following child elements:

1. FromComposition; specifying that default the EIT-sub-tables are generated as result from the transport-stream composition and adaptations unless it is specified differently.
2. Empty; specifying that default an empty or no EIT-sub-table is generated unless it is specified differently.

EitActualType/AssembledTable/EitTimeFilter

This optional element specifies the EIT-event-filtering.

Num3HourSegments

Type: unsignedInt, Use: optional

Specifies the number of 3-hours-periods, specifying a time window to retain in the EIT-Schedule-table (for the EIT-Present-Following table it has no effect). All events outside the window are deleted.

FilterPfEvents

Type: boolean, Use: optional, Default: true

Specifies whether the events in the EIT-Present-Following table have to be filtered (for the EIT-Schedule table it has no effect). If true, the MuxXpert determines which event becomes the present and following event based on their start time and duration. Past events are deleted.

If false, the present and the following event are assigned independent on their start time and duration.

3.4.5.13 Eit-P/F-other

The EIT-p/f-other describes the present and the following event of other transport-streams. The EIT-p/f-other is an optional DVB-SI table.

EitpfOther

Optional element, specifying the assembly of the EIT-p/f-other. This element is of the type *EitOtherType* that defines the EIT-p/f-other and the EIT-schedule-other tables.

3.4.5.14 EIT-Schedule-other

The EIT-schedule-other describes the future events of other transport-streams. The EIT-schedule-other is optional.

EitSchedOther

Optional element, specifying the assembly of the EIT-schedule-other. This element is of the type *EitOtherlType* that defines the EIT-p/f-other and the EIT-schedule-other tables.

3.4.5.15 EIT-other Type

The EIT-other type is used for the specification of the EIT-p/f-other and the EIT-schedule-other table assembly.

EitOtherType

An element of this type specifies the assembly of the EIT-p/f-other or the EIT-schedule-other. One of the following three child elements can be used for specifying how the table is constructed: TableNotGenerated, ExternallySuppliedTable or AssembledTable.

Further it has two optional child elements: TableCycleProperties and TableDescriptorSorting.

Below the graphical diagram of the *EitOtherType* is shown.

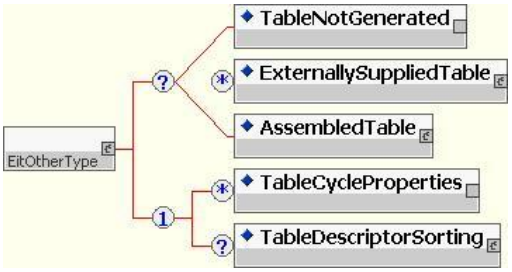


Figure 65. Graphical diagram of *EitOtherType*.

TableNotGenerated

Empty element, that specifies that this table is not generated at all.

ExternallySuppliedTable

Specifying the table data to be inserted, see section 3.7.1. In behalf of the EIT- schedule-other tables this can be a sequence of ExternallySuppliedTable elements, to allow multiple EIT sub-tables with different table-ids.

EitOtherType/AssembledTable

Element, specifying how the EIT-other is assembled, see section 3.4.5.15.1.

TableCycleProperties

Element, specifying the table cycle properties (see section 3.7.2). If this element is absent, the default values are used. In behalf of the EIT- schedule-other tables this can be a sequence of TableCycleProperties elements, to allow different cycle properties for EIT sub-tables with different table-ids.

TableDescriptorSorting

Element, specifying the order in which descriptors have to appear the table (see section 3.7.3). This element is ignored in combination with the option ExternallySuppliedTable.

3.4.5.15.1 EIT-other Type Assembled Table

EitOtherType/AssembledTable

Element, specifying how the EIT-other sub-tables are assembled. It contains a sequence of EitOtherSubTable elements.

Below the graphical diagram of the EitOtherType/AssembledTable is shown.

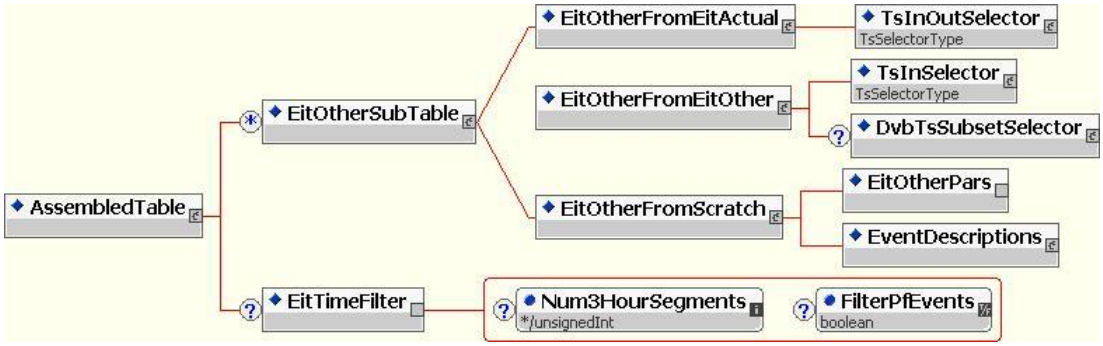


Figure 66. Graphical diagram of EitOtherType/AssembledTable.

EitOtherType/AssembledTable/EitOtherSubTable

Element, specifying how one or more EIT-other sub-table are assembled by one of the following three child elements: EitOtherFromEitActual, SdtOtherFromSdtOther or EitOtherFromScratch.

EitOtherType/AssembledTable/EitOtherSubTable/EitOtherFromEitActual

Element; specifying an EIT-actual table taken from an input or output stream. This EIT-actual table is converted to EIT-other sub-tables by changing the table-id. The EIT-actual table to be converted is selected by a transport-stream selector child element (TsInOutSelector of *TsSelector* type see section 3.5.2).

EitOtherType/AssembledTable/EitOtherSubTable/EitOtherFromEitOther

Element specifying EIT-other sub-tables to be included in the output transport-stream. The EIT-other sub-tables are selected by a transport-stream selector child element (TsInSelector of type *TsSelectorType* see section 3.5.2), which selects the input from which to take the EIT-other table, and transport-stream subset selector child element in mode "DVB Qualified" (DvbTsSubsetSelector of type *TsSelectorType* see section 3.5.2) which selects the EIT-other sub-tables to be included.

EitOtherType/AssembledTable/EitOtherSubTable/EitOtherFromScratch

Element, specifying the EIT-other fields, when one EIT-other sub-table is created from scratch. This is specified with two child elements:

1. EitOtherPars; specifying the `service_id`, `transport_stream_id` and the `original_network_id` in the EIT-other sub-table.

   SvcId

   Type: SvcIdType, Use: required
   Specifies the `service_id` in the EIT-other sub-table.

TsId
Type: TsdType, Use: required
Specifies the `transport_stream_id` in the EIT-other sub-table.

OnwId
Type: NwIdType, Use: required
Specifies the `original_network_id` of the EIT-other sub-table.

2. EventDescriptions; specifying the EIT loop descriptors by a sequence of Event elements (see section 3.7.4).

EitOtherType/AssembledTable/EitTimeFilter
This optional element specifies the EIT-event-filtering.

Num3HourSegments
Type: unsignedInt, Use: optional
Specifies the number of 3-hours-periods, specifying a time window to retain in the EIT-Schedule-table (for the EIT-Present-Following table it has no effect). All events outside the window are deleted.

FilterPfEvents
Type: boolean, Use: optional, Default: true
Specifies whether the events in the EIT-Present-Following table have to be filtered (for the EIT-Schedule table it has no effect). If true, the MuxXpert determines which event becomes the present and following event based on their start time and duration. Past events are deleted.
If false, the present and the following event are assigned independent on their start time and duration.

3.4.5.16 TDT

The TDT carries UTC-time and date information [DVB-SI].

Tdt
Optional element, specifying the assembly of the TDT-table.
One of the following three child elements can be used for specifying how the table is constructed: TableNotGenerated, ExternallySuppliedTable or AssembledTable.
Further the Tdt element has two optional child elements: TableCycleProperties and TableDescriptorSorting.

Below the graphical diagram of the Tdt element is shown.

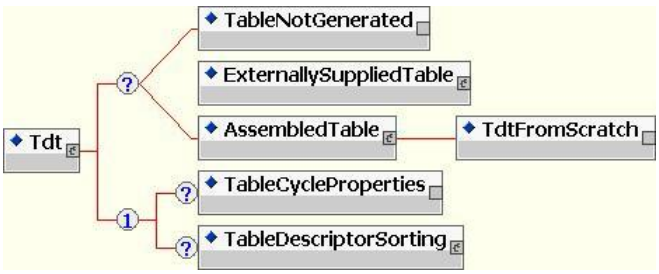


Figure 67. Graphical diagram of Tdt.

TableNotGenerated
Empty element, that specifies that this table is not generated at all.

ExternallySuppliedTable
Specifying the table data to be inserted, see section 3.7.1.

Tdt/AssembledTable
Element, specifying how the TDT is assembled it contains the following child element:
TdtFromScratch: an empty element, that specifies the generation of the TDT and the substitution by the table-cycler of UTC_time field in the TDT with the actual UTC time at the moment of transmission of the TDT section.

TableCycleProperties
Element, specifying the table cycle properties (see section 3.7.2). If this element is absent, the default values are used.

TableDescriptorSorting
Element, specifying the order in which descriptors have to appear the table (see section 3.7.3). This element is ignored in combination with the option ExternallySuppliedTable.

3.4.5.17 TOT

The TOT carries UTC-time, date information and local time offset [DVB-SI]. The TOT contains a descriptor loop, which makes TOT assembly more involved than generation of the TDT. The TOT descriptor loop will usually contain a single descriptor, the local_time_offset_descriptor.

Tot
Optional element, specifying the assembly of the TOT-table.
One of the following three child elements can be used for specifying how the table is constructed: TableNotGenerated, ExternallySuppliedTable or AssembledTable.
Further the Tot element has two optional child elements: TableCycleProperties and TableDescriptorSorting.

Similar to the TDT-generation, the generation of the TOT table includes the substitution of UTC-time by the table-cycler.

Below the graphical diagram of the Tot element is shown.

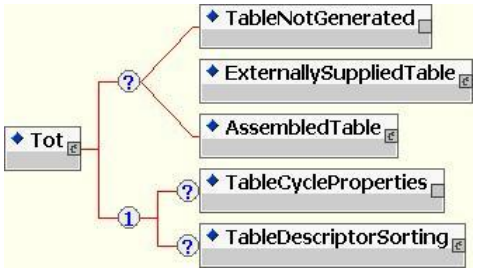


Figure 68. Graphical diagram of Tot.

TableNotGenerated
Empty element, that specifies that this table is not generated at all.

ExternallySuppliedTable
Specifying the table data to be inserted, see section 3.7.1.

Tot/AssembledTable
Element, specifying how the TOT is assembled, see section 3.4.5.17.1.

TableCycleProperties
Element, specifying the table cycle properties (see section 3.7.2). If this element is absent, the default values are used.

TableDescriptorSorting
Element, specifying the order in which descriptors have to appear the table (see section 3.7.3). This element is ignored in combination with the option ExternallySuppliedTable.

3.4.5.17.1 TOT Assembled Table

Tot/AssembledTable
Element, specifying how the TOT is assembled by one of the two following child elements: TotFromTot or TotFromScratch.

Below the graphical diagram of the Tot/AssembledTable element is shown.

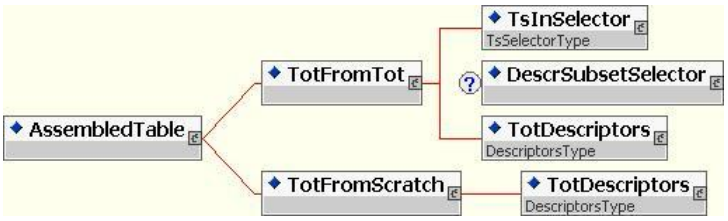


Figure 69. Graphical diagram of Tot/AssembledTable.

Tot/AssembledTable/TotFromTot
Element specifying a TOT table that is used as the basis for the descriptor loop in the outgoing TOT. The TOT taken from a transport-stream input, is selected by a transport-stream selector child element (TsInSelector of type *TsSelectorType* see section 3.5.2) which descriptors have to be copied from the selected TOT to the outgoing TOT is specified with the descriptor subset selector element (DescrSubsetSelector see section 3.6).
Further it contains it has a child element TotDescriptors, that specifies the additional TOT loop descriptors for output transport-stream. This element is of type *DescriptorsType* see section 3.7.6.

Tot/AssembledTable/TotFromScratch
Element specifying generation of the TOT table it has a child element TotDescriptors, that specifies the additional TOT loop descriptors for output transport-stream. This element is of type *DescriptorsType* see section 3.7.6.

3.4.6 Reserved Pids

To prevent the MuxXpert using certain PIDs it is possible to specify user reserved PIDs. The reserved PIDs will not be used by the MuxXpert for automatic PID allocation. However, reservation is overruled:

If the user specifies a mandatory PID assignment that assigns a reserved PID.

If the reserved PID is a mandatory DVB or MPEG PID.

If the PID is specified in the Externally Supplied Table. See section 3.7.1.

Note:
The user doesn't have to specify the DVB and MPEG reserved PIDs.

ReservedPids
Optional element, used for specifying the user reserved PIDs.
The ReservedPids element contains a sequence of ReservedPid elements.

ReservedPids/ ReservedPid
Specifies one reserved PID.

Pid
Type: PidType, Use: required
Specifies one reserved PID.

Below the graphical diagram of the ReservedPids element is shown.

Figure 70. Graphical diagram of ReservedPids.

3.5 Selectors

A *selector* is a functional entity that selects a subset from a collection of DVB objects (services, components, etc.). Each selector implements a *selection criterion*, which can be parameterised by *selection attributes*. Several *selection methods* are available to specify the selection criterion. The selection methods and their attributes are specified in the subsections below.

3.5.1 Network Selector

NwSelector
Element; specifying a network selection. The network selector is used to select a "network" in the context of NIT processing. The output of a network selector is a NIT sub-table that describes the selected network.

NwId
Type: NwIdType, Use: required
Specifies the `network_id` of the network to be selected.

Below the graphical diagram of the NwSelector element is shown.

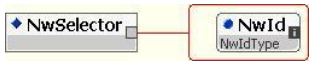


Figure 71. Graphical diagram of NwSelector.

3.5.2 Transport Stream Selector

A transport-stream selector is of type *TsSelectorType*. In this way it is possible to make distinction (in naming) between transport-stream selectors that operate on inputs, outputs or transport-stream selectors that use DVB-qualifiers only.

The precise input and output of a transport-stream selector depends on the context in which the selector is used. In a composition description, a transport-stream selector operates either:

- on a collection of logical transport-stream inputs and/or outputs, or
- on a collection of objects that are related to a specific transport-stream, e.g. a collection of SDT-other sub-tables.
- If the context is service selection, all services contained in the designated transport-stream are selected.
- If the context is SDT-other, all SDT-other sub-tables in the designated transport-stream are selected.
- Etc.

TsSelectorType

An element of this type, specifyies a transport stream selection. The transport-stream selector selects all "objects" of a certain type in a transport-stream.
One of the following two child elements can be used for specifying a specific method to select a transport-stream: DvbQualified or ByLogicalTsId.

Below the graphical diagram of the *TsSelectorType* is shown.

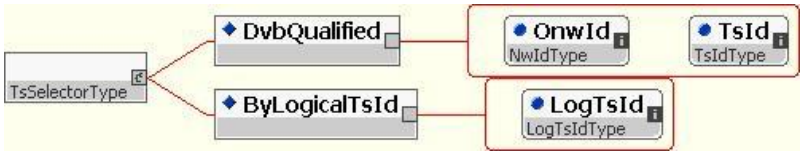


Figure 72. Graphical diagram of *TsSelectorType*.

TsSelectorType/DvbQualified

Defines the selection method "DVB Qualified" of a Transport-Stream Selector. The transport-stream is selected by its DVB attributes `original_network_id` and `transport_stream_id`.

OnwId

Type: NwIdType, Use: required
Specifies `original_network_id` of the transport stream to be selected

TsId

Type: TsIdType, Use: required
Specifies the `transport_stream_id` of the transport stream to be selected

TsSelectorType/ByLogicalTsId

Defines the selection method "Logical TS" of a Transport-Stream Selector. The selector identifies a transport-stream input.

LogTsId

Type: LogTsIdType, Use: required
Specifies the logical transport-stream input or output to be selected.

3.5.2.1 Transport Stream Selector – Selection

Transport-stream selection based on logical TS input/output identifier is self-evident and unambiguous.

Transport-stream selection based on DVB-qualification has to consider one complication: a selection attribute (`network_id` or `ts_id`) may not be available.

A transport-stream selector selects precisely one transport-stream. When selection method "DVB Qualified" is used, the selector may match multiple transport-streams. But only one is selected.

3.5.3 Service Selector

SvcSelector

Element; specifying a service selection. The service selector selects zero or one service from a collection of services. Service selection is used to:

- Refine the collection of services obtained through a transport-stream selector (*add/drop* service);
- Select a service for service adaptation.

The result of a service selector is zero or one service; a service selector cannot select multiple services.

One of the following three child elements can be used for specifying a specific method to select a service: FullyQualified, Short or DvbQualified.

Below the graphical diagram of the SvcSelector element is shown.

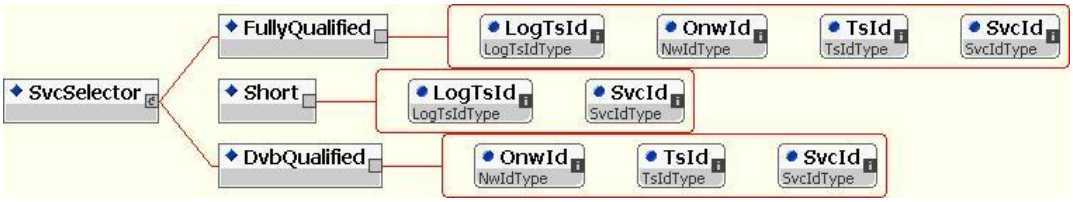


Figure 73. Graphical diagram of SvcSelector.

SvcSelector/FullyQualified

Defines the selection method "Fully Qualified" of a Service Selector. This selection method identifies a service by both the logical input at which the service enters the system and the DVB attributes of the service `original_network_id`, `transport_stream_id` and `service_id`.

LogTsId

Type: LogTsIdType, Use: required
Specifies the logical transport-stream input or output of the service to be selected.

OnwId

Type: NwIdType, Use: required
Specifies `original_network_id` of the service to be selected

TsId

Type: TsIdType, Use: required
Specifies the `transport_stream_id` of the service to be selected

SvcId

Type: SvcIdType, Use: required
Specifies the `service_id` of the service to be selected

SvcSelector/Short

Defines selection method "Fully Qualified Shorthand" of a Service Selector. This selection method is similar to FullyQualified, but without `original_network_id` and `transport_stream_id`. This is an option for "convenience" (two attributes instead of four). Note that a logical input and `service_id` still uniquely identify a service, because it is an MPEG-2 requirement that a program number (`service_id`) is unique within a transport-stream.

LogTsId

Type: LogTsIdType, Use: required
Specifies the logical transport-stream input or output of the service to be selected.

SvcId

Type: SvcIdType, Use: required
Specifies the `service_id` of the service to be selected

SvcSelector/DvbQualified

Defines the selection method "DVB Qualified" of a Service Selector. This selection method identifies a service by its DVB attributes only (`original_network_id`, `transport_stream_id` and `service_id`). The service is selected irrespective of the logical input at which the service appears ("location transparency").

OnwId

Type: NwIdType, Use: required
Specifies `original_network_id` of the service to be selected

TsId

Type: TsIdType, Use: required
Specifies the `transport_stream_id` of the service to be selected

SvcId

Type: SvcIdType, Use: required
Specifies the `service_id` of the service to be selected

3.5.3.1 Service Selector – Selection

A service selector matches an incoming service if all selection attributes specified in the service selector match the actual value in the incoming transport-streams.

One specific remark applies to the network identifier, which is called `network_id` in the NIT, while it is called `onw_id` in the service selector (which is the DVB convention).

A complication arises if the service selector matches multiple services. This may happen if, intentionally or unintentionally, the same service is contributed in multiple incoming transport-streams. The behaviour depends on the application of the service selector:

If the service selector is used for service *dropping*, then all copies of the selected service are dropped.

If the service selector is used for service *adding*, then just a single copy of the selected service is added.

3.5.4 Component Selector

CompSelector

Element; for specifying a component selection. A Component Selector operates on a service or on a collection of services. The output of a component selector is zero, one or multiple components. Component selection is always based on PSI (unreferenced PIDs are selected with PID selectors.)

Component selectors can be used to:

- Refine the collection of components in a service or in a transport-stream (*add / drop* components);
- Select a component or a set of components for component *adaptation*.

SingleGroupFlag

Type: SingleGroupType, Use: required

Defines the subclass:

- "*single*": The selector either selects nothing or selects a single component.
- "*group*": the selector can select zero, one or more components (a collection of components, sometimes referred to as a *group* of components).

The component-**single** selector is meant for operations that are meaningful only when applied to a single component, e.g. assignment of `component_tag`.

Component Selection methods are based on "PID" selection or a "Combined" selection based on "Stream Type" and/or "Descriptor Field".

One of the following two child elements can be used for specifying a specific method to select a single component or a group of components: ByComponentPid or CombinedSelection.

Below the graphical diagram of the CompSelector element is shown.

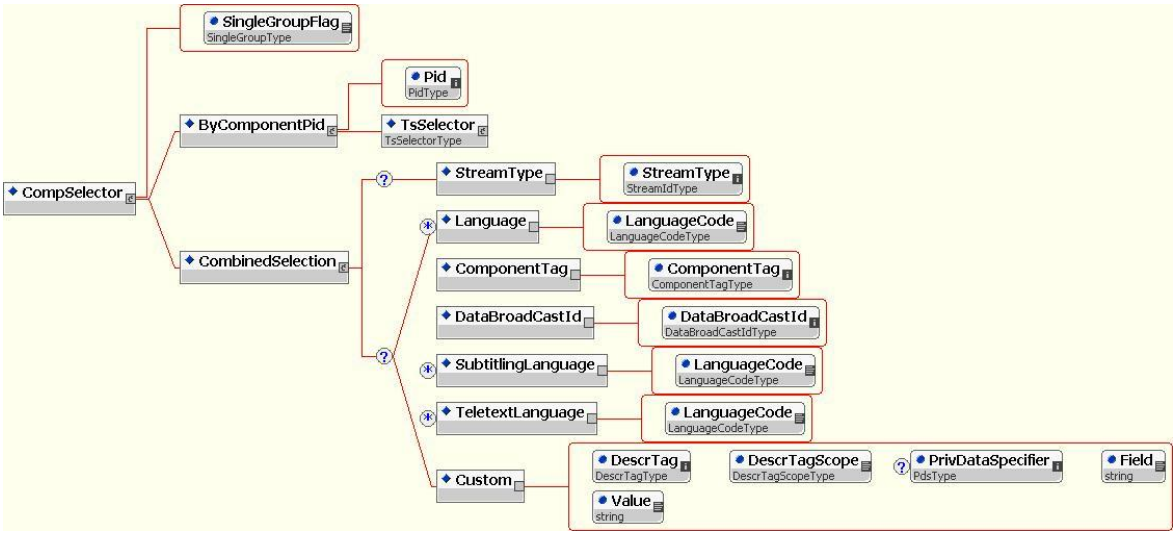


Figure 74. Graphical diagram of CompSelector.

CompSelector/ByComponentPid

Defines the selection method "Component PID" of a Component Selector. This selection method is based on the component's PID. The Pid attribute together the TsSelector element uniquely identifies a component. Selection method "ByComponentPid" is subtly different from a PidSelector that selects an "Unreferenced PID". The "Component PID" method selects a component only if it is part of the service (or collection of services), as selected by the selector that sets the context for the component selector. A PidSelector does not consult PSI at all.

Note: the TsSelector is needed to distinguish components within a service that have identical PIDs which might be the case after the service-composition and before the PID-(re)allocation.

Pid

Type: PidType, Use: required
Specifies the PID of the component to be selected

CompSelector/CombinedSelection

Within CombinedSelection three different methods are allowed for selection:

1. by Stream Type
2. by a Descriptor Field
3. by Stream Type and by a Descriptor Field

For this purpose it can contain the following child elements: StreamType, Language, ComponentTag, DataBroadCastId, SubtitlingLanguage, TeletextLanguage and Custom.

CompSelector/CombinedSelection/StreamType

Defines the selection method "Stream Type" of a Component Selector. This selection method is based on the component's stream type as listed in the component-description loop of the PMT. Stream Type is compared against the value of the StreamType attribute.
This method can be used in combination with one of the "Descriptor Field" selections.

StreamType

Type: StreamTypeType, Use: required
Specifies the stream_type of the component to be selected

Below the different selection methods based on a "Descriptor Field" are listed. Only one of the listed methods can be used.

CompSelector/CombinedSelection/Language

Element, which may occur multiple times for specifying a sequence of ISO_639_language_code values. Component selection is based on comparing the fields ISO_639_language_code in the descriptor ISO_639_language_descriptor with the attribute LanguageCode.

LanguageCode

Type: LanguageCodeType, Use: required
Specifies the ISO_639_language_code in the descriptor ISO_639_language_descriptor of the component to be selected

---

CompSelector/CombinedSelection/ComponentTag

Component selection based on comparing field component_tag in descriptor stream_identifier_descriptor with attribute ComponentTag.

ComponentTag

Type: ComponentTagType, Use: required
Specifies the component_tag in descriptor stream_identifier_descriptor of the component to be selected

CompSelector/CombinedSelection/DataBroadCastId

Component selection based on comparing field data_broadcast_id in descriptor data_broadcast_id_descriptor with the sequence of attributes DataBroadCastId.

DataBroadCastId

Type: DataBroadCastIdType, Use: required
Specifies data_broadcast_id in descriptor data_broadcast_id_descriptor of the component to be selected

CompSelector/CombinedSelection/SubtitlingLanguage

Element, which may occur multiple times for specifying a sequence of ISO_639_language_code values. Component selection is based on comparing the fields ISO_639_language_code in the descriptor subtitling_descriptor with the attribute LanguageCode.

LanguageCode

Type: LanguageCodeType, Use: required
Specifies the ISO_639_language_code in descriptor subtitling_descriptor of the component to be selected

CompSelector/CombinedSelection/TeletextLanguage

Element, which may occur multiple times for specifying a sequence of ISO_639_language_code values. Component selection is based on comparing the fields ISO_639_language_code in the descriptor teletext_descriptor with the attribute LanguageCode.

LanguageCode

Type: LanguageCodeType, Use: required
Specifies ISO_639_language_code in descriptor teletext_descriptor of the component to be selected

CompSelector/CombinedSelection/Custom

The Custom component-selection method can be used to select components based on:
an arbitrary field (the *selection field*), specified with the Field attribute within an arbitrary descriptor-tag scope and the value for the private_data_specifier field in the PDSD in case the "*private*" scope is selected in an arbitrary descriptor in the ES-info descriptor loop of the PMT.
The value of the selected field is compared with attribute Value attribute. If the values matches then the component(s) is/are selected.

Field

Type: string, Use: required
Specifies the selection field within a descriptor

DescrTag

Type: DescrTagType, Use: required
Specifies the descriptor tag of the descriptor

DescrTagScope

Type: DescrTagScopeType, Use: required
Specifies the descriptor tag scope of the descriptor
The following values are allowed for the DescrTagScope attribute:
- "*public*": Descriptors not preceded by a PDSD are selected.
- "*private*": Descriptors preceded by a PDSD where the private_data_specifier matches

the PrivDataSpecifier attribute, up to the end of the loop or the occurrence of another PDSD are selected.
- "*global*": All descriptors, irrespective of PDSDs are selected.

PrivDataSpecifier
Type: PdsType, Use: optional
Specifies the `private_data_specifier` of the descriptor in case the selection scope is "*private*".

Value
Type: string, Use: required
Specifies the value that is compared to the value of the selected field.

Note:
The Custom component-selection is not (yet) supported by the MuxXpert.

3.5.4.1 Component Selector – Selection

A component selector selects all components for which the selection attributes specified in the component selector match the actual value of those attributes in the incoming service(s).

Whether a component selector based on "Language", "SubtitlingLanguage" or "TeletextLanguage" matches depends on the context of the component selector:
When the component selector is used for positive selection then the component selector selects the component if at least one of the actual languages attributes of the component matches with at least one of the languages in the component selector.
When the component selector is used for negative selection then the component selector drops the component if all actual languages attributes of the component matches with the languages in the component selector.

If for a certain component the selection attribute is not present (e.g. method "Language", but no `ISO_639_language_descriptor` is present), then that component is not selected. Furthermore, no error event message is generated.

Notes:
If a component selector is based on a certain descriptor field and the descriptor containing the field is not present, then the component is not selected.

Component-single selectors select zero or one component. A problem occurs if *multiple* components match the selection criterion. This is probably against the intention of the Producer, who expects a single match. To handle this situation robustly, the MuxXpert generates an event message.

If a component selector matches multiple components and the single / group flag is set to *single*, then the output of the selector is one of the matching components.

If a component selector matches multiple components and the single / group flag is set to *group*, then the output of the selector is all matching components.

3.5.5 PID Selector

PidSelector
Element; specifying a PID selection. The PID selector operates on a collection of transport-streams. The result of a PID selector is zero, one or more elementary stream ("PID").
A PID selector can be used to:

1. Select elementary streams in transport-streams that do not contain PSI (e.g. raw output of a

video encoder);

2. Identify elementary streams that are not listed in the PSI, e.g. special-purpose streams on a pre-defined PID.

One of the following four child elements can be used:

1. DvbQualified or ByLogicalTsId for specifying one specific PID
2. DvbQualifiedRange or ByLogicalTsIdRange for specifying a PID range.

Below the graphical diagram of the PidSelector element is shown.

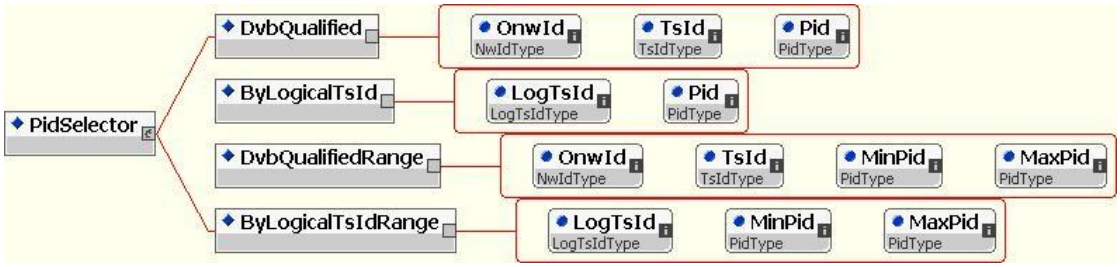


Figure 75. Graphical diagram of PidSelector.

PidSelector/DvbQualified

Defines the selection method "DVB Qualified" of a PID Selector. The selector identifies a PID by using DVB attributes (`original_network_id`, `transport_stream_id` and `PID`).

OnwId

Type: NwIdType, Use: required
Specifies `original_network_id` of the elementary stream to be selected

TsId

Type: TsIdType, Use: required
Specifies the `transport_stream_id` of the elementary stream to be selected

Pid

Type: PidType, Use: required
Specifies the `PID` of the elementary stream to be selected

PidSelector/ByLogicalTsId

Defines the selection method "Logical TS" of a PID Selector. The selector identifies a PID by its transport-stream input at which the PID enters the system and the PID value.

LogTsId

Type: LogTsIdType, Use: required
Specifies the transport-stream input of the elementary stream to be selected

Pid

Type: PidType, Use: required
Specifies the `PID` of the elementary stream to be selected

PidSelector/DvbQualifiedRange

Defines the selection method "DVB Qualified Range" of a PID Selector. The selector identifies a series of PIDs by using DVB attributes (`original_network_id`, `transport_stream_id`) and by specifying a range of PIDs by a minimum and maximum PID value.

OnwId

Type: NwIdType, Use: required
Specifies `original_network_id` of the elementary stream to be selected

TsId

Type: TsIdType, Use: required
Specifies the `transport_stream_id` of the elementary stream to be selected

PidMin

Type: PidType, Use: required
Specifies the first `PID` value of the elementary stream to be selected

PidMax

Type: PidType, Use: required
Specifies the last `PID` value of the elementary stream to be selected

PidSelector/ByLogicalTsIdRange

Defines the selection method "Logical TS Range" of a PID Selector. The selector identifies a series of PIDs by its transport-stream input at which the PID enters the system and a range of PIDs by a minimum and maximum PID value.

LogTsId

Type: LogTsIdType, Use: required
Specifies the transport-stream input of the elementary stream to be selected

PidMin

Type: PidType, Use: required
Specifies the first `PID` value of the elementary stream to be selected

PidMax

Type: PidType, Use: required
Specifies the last `PID` value of the elementary stream to be selected

3.5.6 CA-Component Selector

CaCompSelector

Element; specifying a conditional access component selection. The CA-Component selector operates on a transport-stream, collection of services or components. The output of the CA-Component selector is zero or one EMM or ECM stream[5]. Whether an EMM-stream or ECM-stream is selected depends on the context. If the context is a transport-stream then EMM streams are selected, if the context is a service or component then ECM streams are selected.
The CA-Component selector can be used to select the EMM and ECM-streams, for a particular output transport-stream and for a particular conditional-access system.

[5] EMM streams are used to carry encrypted "Entitlement" messages, which enable receivers – usually with the help of a smart card – to de-scramble certain designated programs or channels. The ECM streams carry the key that are needed for de-scrambling.

CasId

Type: CasIdType, Use: required
Specifies the `CA_system_ID` ([MPEG-2 SYS]) of the ECM or EMM stream to be selected.

Below the graphical diagram of the CaCompSelector element is shown.

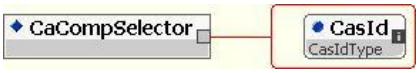


Figure 76. Graphical diagram of CaCompSelector.

Note:
If the CA-Component selector acts as an EMM selector, the MuxXpert uses the CAT in the transport-stream. In this CAT, the CA-descriptor with the selected CA_system_ID is located. Field CA_PID in this descriptor lists the EMM PID.

If the CA-Component selector acts as an ECM selector, the MuxXpert uses the PMT related to the service. In this PMT, the CA-descriptor with the selected CA_system_ID is located. Field CA_PID in this descriptor lists the ECM PID.

3.5.7 Table Selector

TableSelector

Element; specifying a table selection. The Table Selector operates on a transport-stream. The output is either an empty set, or a single (P)SI- or custom- table. Selection based on *sub* tables is not supported.
Table selectors are used to select a particular table for further processing.
One of the following two child elements can be used for specifying a specific method to select a table: DvbQualified or ByLogicalTsId.

Below the graphical diagram of the TableSelector element is shown.

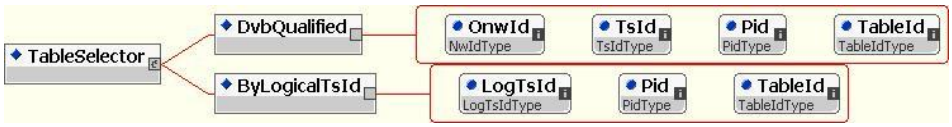


Figure 77. Graphical diagram of TableSelector.

TableSelector/DvbQualified

Defines the selection method "DVB Qualified" of a Table Selector. The selector identifies a table by using DVB attributes (`original_network_id`, `transport_stream_id`, `PID` and `table_id`).

OnwId

Type: NwIdType, Use: required
Specifies `original_network_id` of the table to be selected

TsId

Type: TsIdType, Use: required
Specifies the `transport_stream_id` of the table to be selected

Pid

Type: PidType, Use: required
Specifies the `PID` of the table to be selected

TableId

Type: TableIdType, Use: required

Specifies the table_id of the table to be selected

TableSelector/ByLogicalTsId

Defines the selection method "Logical TS" of a Table Selector. The selector identifies a table by its transport-stream input at which the table enters the system and the PID and table_id value.

LogTsId

Type: LogTsIdType, Use: required
Specifies the transport-stream input of the table to be selected

Pid

Type: PidType, Use: required
Specifies the PID of the table to be selected

TableId

Type: TableIdType, Use: required
Specifies the table_id of the table to be selected

3.5.8 Descriptor Selector

DescrSelector

Element; specifying a descriptor selection. The Descriptor Selector selects zero, one, or a collection of descriptors. A descriptor selector operates on a descriptor loop in an incoming (P)SI table. Descriptor selectors are used to:

- Identify descriptors for addition / deletion;
- Identify descriptors for adaptation.

Selection is complicated by the descriptor-tag scope set by the PDSD (private_data_specifier_descriptor). Descriptor Selection based on descriptor tag is based on descriptor tag and a descriptor tag scope.

DescrTag

Type: DescrTagType, Use: required
Specifies the descriptor tag of the descriptor to be selected

DescrTagScope

Type: DescrTagScopeType, Use: required
Specifies the a descriptor tag scope of the descriptor to be selected
The following values are allowed for the DescrTagScope attribute:

- "*public*": Descriptors not preceded by a PDSD are selected.
- "*private*": Descriptors preceded by a PDSD where the private_data_specifier matches the PrivDataSpecifier attribute, up to the end of the loop or the occurrence of another PDSD are selected.
- "*global*": All descriptors, irrespective of PDSDs are selected.

PrivDataSpecifier

Type: PdsType, Use: optional
Specifies the private_data_specifier of the descriptor to be selected in case the selection scope is "*private*".

Below the graphical diagram of the DescrSelector element is shown.

Figure 78. Graphical diagram of DescrSelector.

Note:

The descriptor selector does not support a selection method to select a specific descriptor in a sequence of descriptors with the same tag and within the same descriptor-tag scope. The PDSD itself can be selected by specifying the PDSD's descriptor tag and setting the descriptor-tag scope to "global."

A descriptor selector may match multiple descriptors (which becomes more likely if DescrTagScope is "global"). When the descriptor selector is used for descriptor deletion, all matching descriptors will be deleted. When the descriptor is used for component selection, only the first occurrence of the descriptor will be selected.

3.6 Subset Selectors

A *Subset selector* selects a subset from a collection of "DVB objects" (transport-streams, services, components or descriptors). The subset selectors consist of a sequence of "DVB selector" elements. Further, it has a SelectionMethod attribute that can specify on of the two modes.

SelectionMethod

Type: SelectionMethodType, Use: required
Specifies the selection method this can be:

- "*positive*": The selectors specified in the subset selector selects objects to be included in the result. Other objects are dropped.
- "*negative*": The selectors specified in the subset selector have to be excluded (dropped) from the result. All objects that are not explicitly selected for exclusion are copied to the result.

The selection mode is either "*positive*" or "*negative*". A combination of positive and negative selection is not possible.

Notes:

The subset selector may select an *empty set* (positive-selection mode, nothing included in the output). This can be used to construct DVB objects "from scratch".

The subset selector may select all objects in the collection (negative-selection mode, nothing excluded in the output).

As example the graphical diagram of the SvcSubsetSelector element is shown.

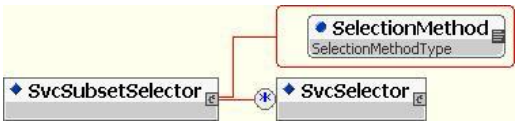


Figure 79. Graphical diagram of SvcSubsetSelector.

The following Subset Selectors can be used:

NwSubsetSelector

Subset Selector specifically for networks. It has the SelectionMethod attribute and the child elements are a sequence of NwSelector elements.

TsSubsetSelector

Subset Selector specifically for transport-streams. It has the SelectionMethod attribute and the child elements are a sequence of TsSelector elements.

DvbTsSubsetSelector

Subset Selector specifically for transport-streams. It has the SelectionMethod attribute and the child elements are a sequence of DvbTsSelector elements.

SvcSubsetSelector

Subset Selector specifically for services. It has the SelectionMethod attribute and the child elements are a sequence of SvcSelector elements.

CompSubsetSelector

Subset Selector specifically for components. It has the SelectionMethod attribute and the child elements are a sequence of CompSelector elements.

CaCompSubsetSelector

Subset Selector specifically for CA components. It has the SelectionMethod attribute and the child elements are a sequence of CaCompSelector elements.

DescrSubsetSelector

Subset Selector specifically for descriptors. It has the SelectionMethod attribute and the child elements are a sequence of DescrSelector elements.

Notes:

If a DVB object is selected to be dropped in negative-selection mode, but the object is not present in the collection of DVB objects, then the MuxXpert treats this selector as a no-op and the MuxXpert does not generate an event message.

If a DVB object is selected to be dropped, and the object is detected multiple times in the collection of DVB objects, then the MuxXpert drops all occurrences of that object.

3.7 Table Related Elements and Types

3.7.1 Externally Supplied Table

ExternallySuppliedTable

Element; specifying table data. The externally supplied (P)SI table, which can either be a standard DVB/MPEG-2 table (or sub-table), or a custom table in private-section format.
The ExternallySuppliedTable element has two child elements. One child element defines the table properties: the ExternalTableProperties element. The contents of the table are specified with the child element SubTables.

Below the graphical diagram of the ExternallySuppliedTable element is shown.

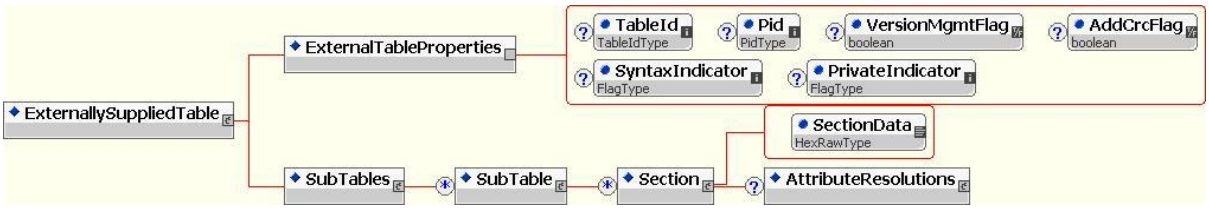


Figure 80. Graphical diagram of ExternallySuppliedTable.

ExternallySuppliedTable/ExternalTableProperties

Element; specifying the table properties.

TableId

Type: TableIdType, Use: optional, however it is required if a table can have different table-ids (e.g. EIT-schedule-other).
Specifies the `table_id` of the table.

Pid

Type: PidType, Use: optional
Specifies the `PID` of the table
If no PID is specified, the MuxXpert will use the default PID for the table. In case a PID is specified this PID must be marked as user reserved PID to prevent the MuxXpert from using this PID for other purposes. See section 3.4.6.

VersionMgmtFlag

Type: boolean, Use: optional (default '*false*')
Specifies whether automatic table-version management is requested. If the value is set to '*true*' then the MuxXpert manages the `version_number` field of the table, otherwise the MuxXpert assumes that the Producer manages the `version_number` field.

AddCrcFlag

Type: boolean, Use: optional (default '*false*')
Specifies whether automatic CRC generation is requested. If the value is set to "*true*" for a particular table, and the table has generic section syntax (SyntaxIndicator is '1'), then the MuxXpert computes the CRC_32 for each section within the table and append it to the section data.
No automatic CRC generation can only work if the Producer does not request table-version management, because the table version is included in the CRC computation. No CRC is generated for tables with private syntax.

SyntaxIndicator

Type: FlagType, Use: optional (default '1')
Specifies whether this is a table in *generic* section syntax (SyntaxIndicator is '1'), or a table with *private* syntax (SyntaxIndicator is '0'.) For each section in the table, the MuxXpert will set field `section_syntax_indicator` in the private-section header to the value of SyntaxIndicator.

PrivateIndicator

Type: FlagType, Use: optional (default '0')
Specifies the value of the `private_indicator` field in the private-section header.

Notes:

A Producer that uses the current/next mechanism has to perform its own table-version management. It cannot rely on automatic table-version management as requested by VersionMgmtFlag.

For tables with entirely private syntax (attribute SyntaxIndicator is '0'), special semantics apply:

1. Table-version management is not supported. The Producer may not specify the VersionMgmtFlag attribute.
2. The MuxXpert does not support generation of CRC_32. The Producer needs not to specify the AddCrcFlag attribute.
3. Only one single SubTable is allowed.

ExternallySuppliedTable/Subtables

Specifies the sub-tables of the table. This element may have zero, one or more SubTable elements.

ExternallySuppliedTable/Subtables/SubTable

Specifies the contents of one sub-table. This element may have zero, one or more Section elements.

ExternallySuppliedTable/ Subtables/SubTable/Section

Specifies the contents of one Section. This element has one child element: AttributeResolutions element (optional) specifies the attribute resolutions see section 3.7.7.

SectionData

Type: HexRawType, Use: required

Specifies the contents of a single section. The section data starts at field table_id_extension (refer to Table 2-30 in [MPEG-2 SYS]).

The section data *excludes*:

- The 3-byte header of a private section (table_id, ..., private-section_length).
- Field CRC_32 (unless element AddCrcFlag is *false*).

These fields are computed by the MuxXpert.

Notes:

The Producer of RMC-Data must specify all sub-tables and all sections of a sub-table within an ExternallySuppliedTable element.

The Producer of RMC-Data must specify the sections of a table ordered by section_number.

If the MuxXpert receives a Table with one or more sections that have a length greater or equal than 4096 bytes, then the MuxXpert discards the table in its entirety.

3.7.2 Table Cycle Properties

TableCyclingProperties

Element; specifying the repetition rate parameters at which the table should be "cycled". The cycling parameters are encoded in six optional attributes.

TableId

Type: TableIdType, Use: optional
Specifies the `table_id` of the table.
In case a table can have different table-ids (e.g. EIT-schedule-other) this field can be used to specify for which table-id these properties are valid. If no table-id is specified, the properties are valid for all table-ids of the table

MinimumCycleRate

Type: CycleRateType, Use: optional
Specifies the minimum table repetition rate in number of repetitions per second (Hz). If it is not specified, the default value will be used.

TargetCycleRate

Type: CycleRateType, Use: optional
Specifies the target repetition rate. This rate is used if sufficient bit rate is available for playing out (P)SI. If it is not specified, the default value will be used.

Priority

Type: TablePriorityType, Use: optional (default "*normal*")
Specifies the priority of the (P)SI table *relative* to the other tables. The priority is used to re-assign repetition rates when the total (P)SI bit rate is not sufficient to play out all tables at their requested repetition rate. The `Priority` values can be: "*high*", "*normal*" or "*low*".

MinimumSectionDistanceMs

Type: SectionDistanceMsType, Use: optional (default 25)
Specifies the minimum spacing between sections from the same sub-table (default a minimum spacing of 25 ms is used, as required by DVB).

VersionNr

Type: VersionNrType, Use: optional
Specifies the `version_number` field of the table and of its sub-tables. If not specified, the table version is managed by the MuxXpert.
The version number does not apply to externally supplied tables.

MinimumVersionTime

Type: NumSecondsType, Use: optional (default 5)
Specifies the minimum time period a table is transmitted before the version is incremented, with a default of 5 seconds. This can only specified for the tables for which the MuxXpert performs the table version management.

Below the graphical diagram of the TableCycleProperties element is shown.

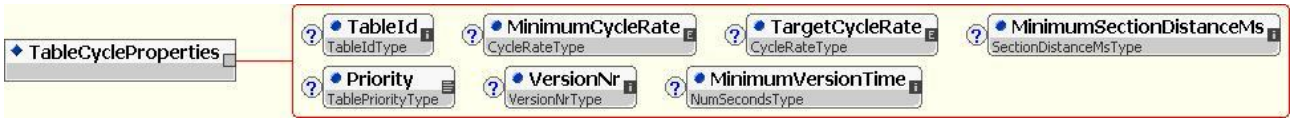


Figure 81. Graphical diagram of TableCycleProperties.

3.7.3 Table Descriptor Sorting

Descriptor sorting can be considered the last step in the (P)SI assembly process. Sorting is not applied to externally supplied tables.

The sort order is specified by means of an ordered list of descriptor tags. A separate list can be specified for public descriptors and for each private descriptor-tag scope. The sort is applied independently for each descriptor-tag scope. Descriptor sorting may not change the descriptor-tag scope in which the descriptor is inserted. If multiple descriptors (within the same scope) have the same de-

scriptor tag, or if the descriptor tag is not listed, then the sequential ordering of these descriptors may not change either.

Descriptors with a descriptor tag that occurs first in the list are output first, followed by the descriptor tag that is second in the list, etc. Descriptors with descriptor tags that do not occur in the sort list are put at the end of the loop (but still within the same scope). An example is provided in Figure 82 below.

Sort Order: Public: 3, 2, 1
PDSD-x: 2, 3

In: 1A, 2A, 1B, 2B, 3, PDSD-x, 1A, 2A, 1B, 2B, 3

Out: 3, 2A, 2B, 1A, 1B, PDSD-x, 2A, 2B, 3, 1A, 1B

Figure 82. Descriptor sorting – example.

The RMC-Data does not support the following options:

Specification of descriptor order per sub-table.

Sorting of PDSD sections.

TableDescriptorSorting

Element; specifying the descriptor ordering within a table by means of a sequence of ScopeDescrSorting elements that specifies the sorting within a certain descriptor-tag scope.

Below the graphical diagram of the TableDescriptorSorting element is shown.

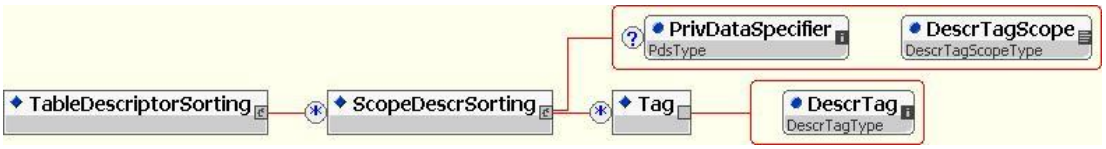


Figure 83. Graphical diagram of TableDescriptorSorting.

TableDescriptorSorting/ScopeDescrSorting

Specifies the descriptor sorting within a descriptor-tag scope by means of a sequence of Tag elements.
The descriptor-tag scope is set with the DescrTagScope and PrivDataSpecifier attributes.

DescrTagScope

Type: DescrTagScopeType, Use: required
Specifies the a descriptor tag scope of the descriptor to be sorted
The following values are allowed for the DescrTagScope attribute:

- "*public*": Descriptors not preceded by a PDSD.
- "*private*": Descriptors preceded by a PDSD where the `private_data_specifier` matches the PrivDataSpecifier attribute, up to the end of the loop or the occurrence of another PDSD.

PrivDataSpecifier

Type: PdsType, Use: optional
Specifies the `private_data_specifier` of the descriptor to be sorted in case the sorting scope is "*private*".

TableDescriptorSorting/ScopeDescrSorting/Tag

Specifies one descriptor tag.

DescrTag

Type: DescrTagType, Use: required

Specifies the descriptor tag.

3.7.4 Event

Event

Element; specifying the contents of one single event within the event loop of an Event information section.

Refer to Figure 84 below, which originates from Table 7 in [DVB-SI]. The event starts at "`event_id`", and ends at the end of the descriptor loop.

The child element EventPars specifies the event parameters and the child element EventDescriptors specify the descriptors in the event's descriptor loop.

Syntax	# Bits	Identifier
event_information_section(){		
table_id	8	uimsbf
section_syntax_indicator	1	bslbf
: : :		
segment_last_section_number	8	uimsbf
last_table_id	8	uimsbf
for(i=0;i<N;i++){		
event_id	16	uimsbf
start_time	40	bslbf
duration	24	uimsbf
running_status	3	uimsbf
free_CA_mode	1	bslbf
descriptors_loop_length	12	uimsbf
for(i=0;i<N;i++){		
descriptor()		
}		
}		
CRC_32	32	rpchof
}		

A single event as encoded with the Event element

Figure 84. Part of the EIT that is encoded with the Event element.

Below the graphical diagram of the Event element is shown.

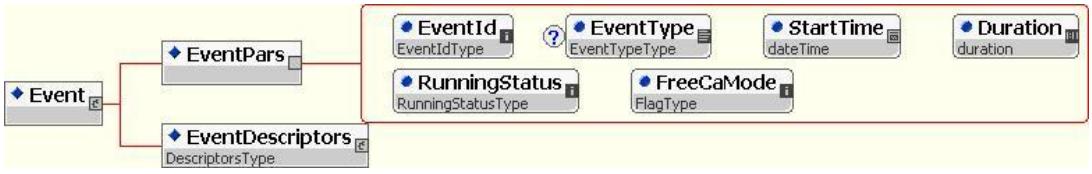


Figure 85. Graphical diagram of Event.

Event/EventPars

The event parameters are encoded in five required attributes of the EventPars element.

EventId

Type: EventIdType, Use: required

Specifies the `event_id` of the event.

EventType
Type: EventTypeType, Use: optional (default: "*automatic*")
Specifies the event type. This can be set to "*automatic*", "*present*", "*following*" or "*schedule*".
The intended use for the `EventType` attribute is to specify whether an event is a present or a following event, regardless its start time and duration. In that case the `FilterPfEvents` attribute in EitTimeFilter element should be set to "*false*".

StartTime
Type: dateTime, Use: required
Specifies the `start_time` of the event (e.g. "2007-03-20T13:35:00").

Duration
Type: duration, Use: required
Specifies the `duration` of the event (e.g. "PT1H30M").

RunningStatus
Type: RunningStatusType, Use: required
Specifies the `running_status` of the event.

FreeCaMode
Type: FlagType, Use: required
Specifies the `free_CA_mode` of the event.

Event/EventDescriptors
Element; specifying the descriptors in the event's descriptor loop. This element is of type *DescriptorsType* see section 3.7.6.

Note:
If the MuxXpert receives an Event that would lead to an EIT section with a length greater or equal than 4096 bytes, then the MuxXpert discards the Event.

3.7.5 Descriptor Composition Type

DescriptorCompositionType
The type is a used for the composition of descriptors. This is the process of dropping and/or adding descriptors from/to a (collection of descriptor) loop(s). This is used in the specification of the service adaptations and the component adaptations.
An element of this type, first specifies which descriptors from the incoming descriptor loop have to be copied to the outgoing descriptor loop. This subset is specified with a descriptor subset selector element (DescrSubsetSelector, see section 3.6). To create a descriptor loop from scratch an empty subset can be specified. Thereafter the descriptors to be added to the descriptor loop are specified, by the NewDescriptors child element. This element is of type *DescriptorsType* see section 3.7.6.

Below the graphical diagram of the *DescriptorCompositionType* is shown.

Figure 86. Graphical diagram of *DescrCompositionType*.

Notes:

New descriptors that are added, are subject to attribute *resolution*.

Descriptors that are remultiplexed from incoming transport-streams are subject to attribute *relocation*.

Descriptors in the outgoing descriptor loop are also subject to *descriptor sorting*.

If descriptor composition is applied on multiple descriptor loops in parallel, then the MuxXpert applies the descriptor composition on each individual descriptor loop.

Example:
A component selector selects all Spanish audio components. To these components, a certain descriptor is added with descriptor composition. The MuxXpert will add the descriptor to *each* component's ES-info section in the PMT.

3.7.6 Descriptors Type

DescriptorsType

The type is a used for specifying a set of descriptors that is added to a descriptor loop.
An element of *DescriptorsType* contains a sequence of Descriptor elements.

Below the graphical diagram of the *DescriptorsType* is shown.

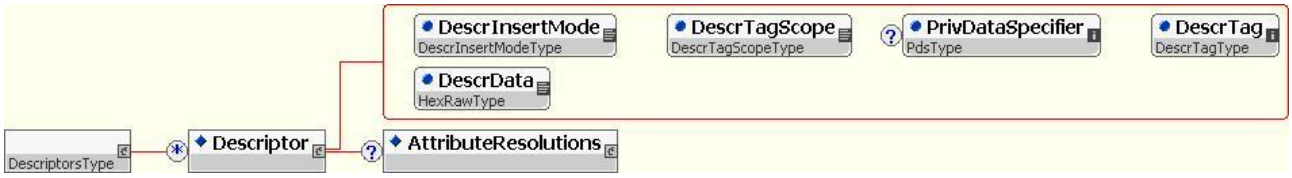


Figure 87. Graphical diagram of *DescriptorsType*.

DescriptorsType/Descriptor

Element; specifying the content of a single descriptor. This element has one optional child element (AttributeResolutions) and five required attributes.

DescrInsertMode

Type: DescrInsertModeType, Use: required
Specifies the descriptor insert mode, this can be:

1. "*add*": the descriptor is added to the descriptor loop, irrespective whether the descriptor is already present.
2. "*replace*": the descriptor is added to the descriptor loop. In case the descriptor is already present then all old occurrences within the relevant descriptor-tag scope are deleted.

DescrTag

Type: DescrTagType, Use: required
Specifies the descriptor tag of the descriptor to be inserted

DescrTagScope

Type: DescrTagScopeType, Use: required
Specifies the he descriptor-tag scope, this can be "*public*" or "*private*".

PrivDataSpecifier

Type: PdsType, Use: optional

Specifies the value for the private_data_specifier field in the PDSD in case the "*private*" scope is selected

PrivDataSpecifier
Type: HexRawType, Use: optional
Specifies the data of a single descriptor (without descriptor_tag and descriptor_length).

AttributeResolutions
Element; specifying the values and fields in the descriptor data that need to be substituted, see section 3.7.7.

Notes:

If the MuxXpert receives a Descriptor that has a length greater or equal than 256 bytes, then the MuxXpert discards the descriptor.

If a descriptor is to be added, and the descriptor is already present in the descriptor loop and DescrInsertMode "*replace*" is specified, then the MuxXpert *replaces* the existing descriptor with the new descriptor. If the descriptor occurs multiple times, all old occurrences within the relevant descriptor-tag scope are deleted.

If a descriptor is to be added, and the descriptor is already present in the descriptor loop and DescrInsertMode "*add*" is specified, then the MuxXpert *adds* the new descriptor to the existing descriptor(s).

If a descriptor is to be added, and the descriptor is to be inserted in a certain private descriptor-tag scope, but the corresponding PDSD is not present in the descriptor loop, then the MuxXpert adds a PDSD (together with the new descriptor, of course).

If one or more descriptors are dropped such that the collection of descriptors within the descriptor-tag scope of a certain PDSD becomes empty, then the MuxXpert drops that PDSD.

The MuxXpert adds descriptors at the end of the descriptors in the specified descriptor-tag scope. This is: just before the next PDSD, or at the end of the descriptor loop for the "last scope".

3.7.7 Attribute Resolution

Attribute Resolution is used in the case that a "data item", which can be a descriptor or an entire (sub-)table, contain DVB attributes that need to be substituted by a "real" attribute value, using *selectors*.

When no selector is specified then the context of the current object is used. For example: when assembling a NIT-actual for a certain transport-stream and no transport-stream selector is specified then attributes from that transport-stream are used.

In case a referenced object is not found, the original attributes are not substituted.

AttributeResolutions
Element; specifying the DVB-attributes that need to be substituted. The child elements are a sequence of the "DVB Attribute" resolution elements: SvcAttributes, TsAttributes, ComponentAttributes and ElemStreamAttributes.

Below the graphical diagram of the AttributeResolutions element is shown.

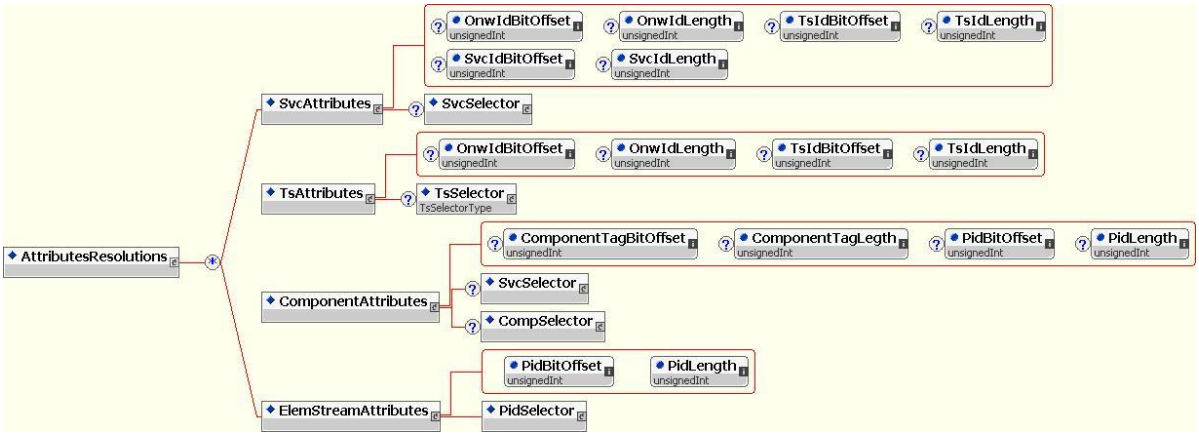


Figure 88. Graphical diagram of AttributeResolutions.

AttributeResolutions/SvcAttributes

Element; specifying which service-identifying-attributes need to be substituted in the data item.
An optional child element is the SvcSelector to indicate which service is referenced.
The attributes that specify the fields and the location within the data item:

OnwIdBitOffset

Type: unsignedInt, Use: optional
Specifies the location of the `original_network_id` field. If absent then the `original_network_id` will not be substituted.

OnwIdBitLength

Type: unsignedInt, Use: optional
Specifies the length in number of bits of the `original_network_id` field. If absent then the `original_network_id` will not be substituted.

TsIdBitOffset

Type: unsignedInt, Use: optional
Specifies the location of the `transport_stream_id` field. If absent then the `transport_stream_id` will not be substituted.

TsIdBitLength

Type: unsignedInt, Use: optional
Specifies the length in number of bits of the `transport_stream_id` field. If absent then the `transport_stream_id` will not be substituted.

SvcIdBitOffset

Type: unsignedInt, Use: optional
Specifies the location of the `service_id` field. If absent then the `service_id` will not be substituted.

SvcIdBitLength

Type: unsignedInt, Use: optional
Specifies the length in number of bits of the `service_id` field. If absent then the `service_id` will not be substituted.

AttributeResolutions/TsAttributes

Defines which transport-stream-identifying-attributes need to be substituted in the data item. An optional child element is the TsSelector to indicate which transport-stream is referenced.
The attributes that specify the fields and the location within the data item:

OnwIdBitOffset

Type: unsignedInt, Use: optional
Specifies the location of the `original_network_id` field. If absent then the `original_network_id` will not be substituted.

OnwIdBitLength

Type: unsignedInt, Use: optional
Specifies the length in number of bits of the `original_network_id` field. If absent then the `original_network_id` will not be substituted.

TsIdBitOffset

Type: unsignedInt, Use: optional
Specifies the location of the `transport_stream_id` field. If absent then the `transport_stream_id` will not be substituted.

TsIdBitLength

Type: unsignedInt, Use: optional
Specifies the length in number of bits of the `transport_stream_id` field. If absent then the `transport_stream_id` will not be substituted.

AttributeResolutions/ComponentAttributes

Defines which component-identifying-attributes need to be substituted in the data item. The optional child elements CompSelector and SvcSelector indicate which single component is referenced.
The attributes that specify the fields and the location within the data-element:

ComponentTagBitOffset

Type: unsignedInt, Use: optional
Specifies the location of the `component_tag` field. If absent then the `component_tag` will not be substituted.

ComponentTagBitLength

Type: unsignedInt, Use: optional
Specifies the length in number of bits of the `component_tag` field. If absent then the `component_tag` will not be substituted.

PidBitOffset

Type: unsignedInt, Use: optional
Specifies the location of the `PID` field. If absent then the `PID` will not be substituted.

PidBitLength

Type: unsignedInt, Use: optional
Specifies the length in number of bits of the `PID` field. If absent then the `PID` will not be substituted.

AttributeResolutions/ElemStreamAttributes
Defines which elementary-stream-identifying-attributes need to be substituted in the data item. An optional child element is the PidSelector (specifying a single PID) to indicate which PID is referenced.

PidBitOffset
Type: unsignedInt, Use: optional
Specifies the location of the `PID` field. If absent then the `PID` will not be substituted.

PidBitLength
Type: unsignedInt, Use: optional
Specifies the length in number of bits of the `PID` field. If absent then the `PID` will not be substituted.

3.8 User Defined Data Types

This section describes all user defined XML data types used in the XSD schema.

AssignmentType
Base type: string, Options: "*preferred*" and "*mandatory*".
Type used for indicating whether an assignment is mandatory ("*mandatory*") or preferred ("*preferred*").

AtscConstellationType
Base type: string, Options: "*8-vsb*" and "*16-vsb*".
Type used for specifying an ATSC constellation value.

BitrateType
Base type: unsignedLong
Type used for specifying a bit rate.

BouquetIdType
Base type: unsignedShort
Type used for specifying a `bouqet_id` value.

CasIdType
Base type: unsignedShort
Type used for specifying a `CA_system_id` value.

ComponentTagType
Base type: unsignedByte
Type used for specifying a `component_tag` value

CycleRateType
Base type: double, Options: minInclusive='1e-6' maxInclusive= '40'
Type used for specifying a table repetition rate in number of repetitions per second (Hz). The minimum frequency is 1e-6 Hz and the maximum frequency is 40 Hz.

DataBroadCastIdType
Base type: unsignedShort
Type used for specifying a `data_broadcast_id` value.

dBmType
Base type: double
Type used for specifying the level of a signal.

DescrInsertModeType
Base type: string, Options: "*add*" and "*replace*".
Type used for specifying the descriptor insert mode, this can be:
- "*add*": the descriptor is added to the descriptor loop, irrespective whether the descriptor is already present.
- "*replace*": the descriptor is added to the descriptor loop. In case the descriptor is already present then all old occurrences within the relevant descriptor-tag scope are deleted

DescrTagScopeType
Base type: string, Options: "*public*", "*private*" and "*global*".
Type used for specifying a descriptor tag scope of the descriptor, this can be:
- "*public*": Descriptors not preceded by a PDSD.
- "*private*": Descriptors preceded by a PDSD.
- "*global*": Descriptors, irrespective of PDSDs.

DescrTagType
Base type: unsignedByte
Type used for specifying a `descriptor_tag` value.

DtmbBandwidthType
Base type: string, Options: "*5MHz*", "*6MHz*", "*6MHz*" and "*8MHz*".
Type used for specifying the ADTB-T and DMB-T/H bandwidth value.

DtmbCodeRateType
Base type: string, Options: "*0.4*", "*0.6*" and "*0.8*".
Type used for specifying the ADTB-T and DMB-T/H FEC code rate value.

DtmbConstellationType
Base type: string, Options: "*qam4-nr*", "*qam4*", "*qam16*", "*qam32*" and "*qam64*".
Type used for specifying the ADTB-T and DMB-T/H constellation value.

DtmbFrameHeaderType
Base type: string, Options: "*pn420*", "*pn595*" and "*pn945*".
Type used for specifying an ADTB-T and DMB-T/H frame header mode.

DtmbInterleaverModeType
Base type: string, Options: "*1*" and "*2*".
Type used for specifying an ADTB-T and DMB-T/H interleaver mode.
Interleave mode 1: B=52, M=240
Interleave mode 2: B=52, M=720

Dvbc2BandwidthType
Base type: string, Options: "*6MHz*" or "*8MHz*".
Type used for specifying the bandwidth of channel raster of the network.

Dvbc2DSliceFecHdrType
Base type: string, Options: "*hem*" or "*robust*".

Type used for specifying the FEC frame header type.

Dvbc2DSliceTiDepthType
Base type: string, Options: "*none*", "*4_symbols*", "*8_symbols*" or "*16_symbols*".
Type used for specifying the time interleaving depth within the data slice.

Dvbc2DSliceType
Base type: string, Options: "*type1*" or "*type2*".
Type used for specifying the data slice type.

Dvbc2GuardIntervalType
Base type: string, Options: "*1_128*", "*1_32*", "*1_16*", "*1_8*", "*1_4*", "*19_128*" or "*19_256*".
Type used for specifying the guard interval used for the modulated signal.

Dvbc2L1TiModeType
Base type: string, Options: "*none*", "*best_fit*", "*4_symbols*" or "*8_symbols*".
Type used for specifying the L1 time interleaving mode

Dvbc2PlpCodeRateType
Base type: string, Options: "*2_3*", "*3_4*", "*4_5*", "*5_6*", "*8_9*" or "*9_10*".
Type used for specifying the code rate used by a PLP.

Dvbc2PlpFecType
Base type: string, Options: "*16k*" or "*64k*".
Type used for specifying the FEC type used by a PLP: 16K LDPC or 64K LDPC

Dvbc2PlpModulationType
Base type: string, Options: "*qam16*", "*qam64*", "*qam256*", "*qam1024*" or "*qam4096*".
Type used for specifying the modulation type used by a PLP.

DvbhOfdmTransmissionModeType
Base type: string, Options: "*2k*", "*4k*" and "*8k*".
Type used for specifying the DVB-H OFDM Transmission Mode.

DvbS2CodeRateType
Base type: string, Options: "*1_4*", "*1_3*", "*2_5*", "*1_2*", "*3_5*", "*2_3*", "*3_4*", "*4_5*", "*5_6*", "*8_9*", and "*9_10*".
Type used for specifying a DVB-S.2 code rate value.

DvbS2ModulationTypeType
Base type: string, Options: "*qpsk*" and "*8-psk*".
Type used for specifying a DVB-S.2 modulation type value.

DvbsCodeRateType
Base type: string, Options: "*1_2*", "*2_3*", "*3_4*", "*5_6*" and "*7_8*".
Type used for specifying a DVB-S code rate value.

DvbsModulationTypeType
Base type: string, Options: "*qpsk*".
Type used for specifying a DVB-S modulation type value.

Dvbt2BandwidthType

Base type: string, Options: "*1_7MHz*", "*5MHz*", "*6MHz*", "*7MHz*", "*8MHz*" or "*10MHz*".
Type used for specifying the bandwidth used for the modulated signal.

Dvbt2FefSignalType

Base type: string, Options: "*zero*" or "*prbs*".
Type used for specifying the generated signal during the FEF period.

Dvbt2FftModeType

Base type: string, Options: "*1k*", "*2k*", "*4k*", "*8k*", "*16k*" or "*32k*".
Type used for specifying the FFT mode (or size) in the channel.

Dvbt2GuardIntervalType

Base type: string, Options: "*1_128*", "*1_32*", "*1_16*", "*1_8*", "*1_4*", "*19_128*" or "*19_256*".
Type used for specifying the guard interval used for the modulated signal.

Dvbt2L1ModulationType

Base type: string, Options: "*bpsk*", "*qpsk*", "*qam16*" or "*qam64*".
Type used for specifying the constellation of the L1-post signalling block.

Dvbt2MisoType

Base type: string, Options: "*off*", "*tx1*", "*tx2*" or "*sum*".
Type used for specifying the Multiple Input Single Output mode: Off (=SISO), Tx1 only, Tx2 only or the sum of Tx1 and Tx2.

Dvbt2PaprType

Base type: string, Options: "*none*", "*ace*", "*tr*" or "*ace_tr*".
Type used for specifying the PAPR reduction used: None, ACE only, TR only or both ACE and TR constellation.

Dvbt2PlpCodeRateType

Base type: string, Options: "*1_2*", "*2_3*", "*3_4*", "*3_5*", "*4_5*" or "*5_6*".
Type used for specifying the code rate used by a PLP.

Dvbt2PlpFecType

Base type: string, Options: "*16k*" or "*64k*".
Type used for specifying the FEC type used by a PLP: 16K LDPC or 64K LDPC

Dvbt2PlpModulationType

Base type: string, Options: "*qpsk*", "*qam16*", "*qam64*" or "*qam256*".
Type used for specifying the modulation type used by a PLP.

DvbtOfdmTransmissionModeType

Base type: string, Options: "*2k*" and "*8k*".
Type used for specifying the DVB-T OFDM Transmission Mode.

EventIdType

Base type: unsignedShort
Type used for specifying an `event_id` value.

EventTypeType

Base type: string, Options: "*automatic*", "*present*", "*following*" or "*schedule*".
Type used for specifying the type of an event.

FecModeType

Base type: string, Options: "*disable*" and "*2D*".
Type used for specifying the error-correction mode, this can be:

- "*disable*" : No error correction is applied.
- "*2D*" : 2D FEC error correction.

FlagType

Base type: unsignedShort, Options: minInclusive='0' maxInclusive= '1'
Type used for specifying the value of a flag, this can be '0' or '1'.

HexRawType

Base type: string
Type used for specifying hex data bytes. The data bytes may be separated by a space. Characters or a character string can be specified by placing the characters between quotes.

Examples:
DescrData = "FF 29 9A A2 80"
DescrData = ""Nederland 1""

IpAddressType

Base type: string
Type used for specifying an IP address. The format of the IP address must consist of 4 IP-address parts separated by a dot (e.g. "192.168.39.2").

IpInProtocolType

Base type: string, Options: "*auto*", "*udp*" and "*rtp*".
Type used for specifying the protocol expected for encapsulation of incoming Transport Packets.

IpOutProtocolType

Base type: string, Options: "*udp*" and "*rtp*".
Type used for specifying the protocol for encapsulation of outgoing Transport Packets.

ModCod

Type: IsdbsModCodType, Use: required
Specifies the modulation type and code rate used for the layer. This can be set to "*bpsk_1_2*", "*qpsk_1_2*", "*qpsk_2_3*", "*qpsk_3_4*", "*qpsk_5_6*", "*qpsk_7_8*", "*8psk_2_3*" or "*not_alloc*".

IsdbsModCodType

Base type: string, Options: "*bpsk_1_2*", "*qpsk_1_2*", "*qpsk_2_3*", "*qpsk_3_4*", "*qpsk_5_6*", "*qpsk_7_8*", "*8psk_2_3*" and "*not_alloc*".
Type used for specifying the ISDB-S modulation type and code rate for a ISDB-S layer, this can be:

- "*bpsk_1_2*" : modulation: BPSK; code rate: 1/2
- "*qpsk_1_2*" : modulation: QPSK; code rate: 1/2
- "*qpsk_2_3*" : modulation: QPSK; code rate: 2/3
- "*qpsk_3_4*" : modulation: QPSK; code rate: 3/4
- "*qpsk_5_6*" : modulation: QPSK; code rate: 5/6
- "*qpsk_7_8*" : modulation: QPSK; code rate: 7/8
- "*8psk_2_3*" : modulation: 8PSK; code rate: 2/3
- "*not_alloc*" : layer not allocated

IsdbtBroadcastTypeType

Base type: string, Options: "*tv*", "*rad1*" and "*rad3*".
Type used for specifying the ISDB-T broadcast type, this can be:

- "*tv*" : 1- or 13-segment TV broadcast
- "*rad1*" : 1-segment radio broadcast
- "*rad3*" : 3-segment radio broadcast

IsdbtCodeRateType

Base type: string, Options: "*1_2*", "*2_3*", "*3_4*", "*5_6*" and "*7_8*".
Type used for specifying the ISDB-T code rate in a layer.

IsdbtGuardIntervalType

Base type: string, Options: "*1_32*", "*1_16*", "*1_8*" and "*1_4*".
Type used for specifying the ISDB-T guard-interval length.

IsdbtLayerType

Base type: string, Options: "*a*", "*b*", "*c*" and "*none*".
Type used for specifying a hierarchical layer.

IsdbtModulationTypeType

Base type: string, Options: "*dpsk*", "*qpsk*", "*qam16*" and "*qam64*".
Type used for specifying the ISDB-T modulation type in a layer.

IsdbtTmModeType

Base type: string, Options: "*1*", "*2*" and "*3*".
Type used for specifying the ISDB-T transmission mode, this can be:

- "*1*" : Mode 1 : 2k
- "*2*" : Mode 2 : 4k
- "*3*" : Mode 3 : 8k

LanguageCodeType

Base type: string, Options: string length = '3'
Type used for specifying an `ISO_639_language_code` value (e.g. "fre").

LogTsIdType

Base type: unsignedShort
Type used for specifying a unique transport-stream identification.

NumSecondsType
Base type: unsignedShort
Type used for specifying a time period in number of seconds.

NwIdType
Base type: unsignedShort
Type used for specifying a `network_id` or a `original_network_id` value.

OfdmBandwidthType
Base type: string, Options: "*5MHz*", "*6MHz*", "*6MHz*" and "*8MHz*".
Type used for specifying an OFDM bandwidth value.

OfdmCodeRateType
Base type: string, Options: "*1_2*", "*2_3*", "*3_4*", "*5_6*" and "*7_8*".
Type used for specifying an OFDM code rate value.

OfdmConstellationType
Base type: string, Options: "*qpsk*", "*qam16*" and "*qam64*".
Type used for specifying an OFDM constellation value.

OfdmGuardIntervalType
Base type: string, Options: "*1_32*", "*1_16*", "*1_8*" and "*1_4*".
Type used for specifying an OFDM guard interval value.

OfdmInterleavingType
Base type: string, Options: "*indepth*" and "*native*".
Type used for specifying an OFDM interleaving value.

OnOffType
Base type: string, Options: "*on*" and "*off*".
Type used for specifying whether certain options are on or off.

PdsType
Base type: unsignedShort
Type used for specifying a `private_data_specifier` value.

PidType
Base type: unsignedShort
Type used for specifying a `PID` value.

Polarisation
Base type: string, Options: "*horz*" and "*vert*".
Type used for specifying the LNB polarisation.

QambInterleavingType

Base type: string, Options: "*i128_j1d*", "*i64_j2*", "*i32_j4*", "*i16_j8*", "*i8_j16*", "*i128_j1*", "*i128_j2*", "*i128_j3*", "*i128_j4*", "*i128_j5*", "*i128_j6*" and "*i128_j7*".
Type used for specifying the QAM-B interleaving as in the table below.

Value	CW (Code Word)	I (# of taps)	J (increment)
"*i128_j1d*"	0001	128	1
"*i64_j2*"	0011	64	2
"*i32_j4*"	0101	32	4
"*i16_j8*"	0111	16	8
"*i8_j16*"	1001	8	16
"*i128_j1*"	0000	128	1
"*i128_j2*"	0010	128	2
"*i128_j3*"	0100	128	3
"*i128_j4*"	0110	128	4
"*i128_j5*"	1000	128	5
"*i128_j6*"	1010	128	6
"*i128_j7*"	1100	128	7
"*i128_j8*"	1110	128	8

QambModulationTypeType

Base type: string, Options: "*qam64*" and "*qam256*".
Type used for specifying a QAM-B modulation type value.

QamModulationTypeType

Base type: string, Options: "*qam16*", "*qam32*", "*qam64*", "*qam128*" and "*qam256*".
Type used for specifying a QAM modulation type value.

RemoveLevelType

Base type: unsignedByte, Options: minInclusive='0' maxInclusive= '4'
Type used for specifying a `RemoveLevel` value. The `RemoveLevel` controls the multiplex behaviour in case the sum of the incoming bit rates exceeds the outgoing bit rate for too long. The `RemoveLevel` can be used to assign a priority to a stream such that "unimportant" streams are removed first. Streams with `RemoveLevel=1` are removed first, streams with `RemoveLevel=4` last and streams whit `RemoveLevel=0` should not be removed at all.

RfFrequencyType

Base type: integer
Type used for specifying the carrier frequency for the RF up-converter in Hz.

RunningStatusType

Base type: unsignedByte
Type used for specifying a `running_status` value.

SectionDistanceMsType

Base type: unsignedByte, Options: minInclusive='0' maxInclusive= '100'
Type used for specifying spacing between sections from the same sub-table in a number of milli-seconds. The minimum distance is 0 ms and the maximum distance is 100ms.

SelectionMethodType

Base type: string, Options: "*positive*" and "*negative*".
Type used for specifying a selection method, this can be:
- "*positive*": The selectors specified in the subset selector selects objects to be included in the result. Other objects are dropped.
- "*negative*" : The selectors specified in the subset selector have to be excluded (dropped)

SingleGroupType

Base type: string, Options: "*group*" and "*single*".
Type used for specifying a component selector subclass, this can be:
- "*single*" : The selector either selects nothing or selects a single component.
- "*group*" : The selector can select zero, one or more components.

StreamIdType

Base type: unsignedByte
Type used for specifying a `stream_id` value.

SvcIdType

Base type: unsignedShort
Type used for specifying a `service_id` value.

SvcTypeType

Base type: unsignedByte
Type used for specifying a `service_type` value.

SymbolRateType

Base type: unsignedLong
Type used for specifying a symbol rate.

TimeOffsetType

Base type: duration, Options: minInclusive='-PT24H00M00S' maxInclusive='+PT24H00M00S'
Type used for specifying a time offset.

TableIdType

Base type: unsignedByte
Type used for specifying a `table_id` value.

TablePriorityType

Base type: string, Options: "*high*", "*normal*" and "*low*".
Type used for specifying the priority of the (P)SI table *relative* to the other tables. The priority is used to re-assign repetition rates when the total (P)SI bit rate is not sufficient to play out all tables at their requested repetition rate. The values can be: "*high*", "*normal*" or "*low*".

TransmitModeType

Base type: string, Options: "*188*" and "*add16*".
Type used for specifying the channel's transmit mode, this can be set to 188-byte transmit mode ("*188*") or 204-byte transmit mode by adding 16 bytes ("*add16*").

TsIdType

Base type: unsignedShort
Type used for specifying a `transport_stream_id` value.

UsedType

Base type: string, Options: "*used*" and "*unused*".
Type used for specifying whether certain options are used or not. The values can be "*used*" or "*unused*".

VersionNrType

Base type: unsignedShort, Options: minInclusive='0' maxInclusive= '31'
Type used for specifying the `version_number` field of the table.