

AS/NZS IEC 62676.2.31:2020
IEC 62676-2-31:2019



Australian/New Zealand Standard™

Video surveillance systems for use in security applications

Part 2.31: Live streaming and control based on web services



AS/NZS IEC 62676.2.31:2020

This Joint Australian/New Zealand Standard™ was prepared by Joint Technical Committee EL-031, Intruder Alarm Equipment and Installations. It was approved on behalf of the Council of Standards Australia on 22 September 2020 and by the New Zealand Standards Approval Board on 7 October 2020.

This Standard was published on 30 October 2020.

The following are represented on Committee EL-031:

- Australian Digital and Telecommunications Industry Association
- Australian Industry Group
- Australian Security Industry Association
- Engineers Australia
- Fire Protection Association Australia
- Insurance Council of Australia
- New Zealand Security Association
- NSW Police Force
- Security Providers Association of Australia
- South Australia Police
- Victorian Security Institute

This Standard was issued in draft form for comment as DR AS/NZS IEC 62676.2.31:2020.

Keeping Standards up-to-date

Ensure you have the latest versions of our publications and keep up-to-date about Amendments, Rulings, Withdrawals, and new projects by visiting:

www.standards.org.au

www.standards.govt.nz

ISBN 978 1 76113 016 8

Australian/New Zealand Standard™

Video surveillance systems for use in security applications

Part 2.31: Live streaming and control based on web services

First published as AS/NZS IEC 62676.2.31:2020.



© IEC 2020 — All rights reserved

© Standards Australia Limited/the Crown in right of New Zealand, administered by the New Zealand Standards Executive 2020

All rights are reserved. No part of this work may be reproduced or copied in any form or by any means, electronic or mechanical, including photocopying, without the written permission of the publisher, unless otherwise permitted under the Copyright Act 1968 (Cth) or the Copyright Act 1994 (New Zealand).

Preface

This Standard was prepared by the Standards Australia/Standards New Zealand Committee EL-031, Intruder Alarm Equipment and Installations.

The objective of this document is to define procedures for communication between network video clients and video transmitter devices. This new set of specifications makes it possible to build network video systems with devices and receivers from different manufacturers using common and well-defined interfaces. These interfaces cover functions such as media and imaging configuration, real-time streaming of audio and video, pan, tilt and zoom (PTZ) control as well as analytics.

The management and control interfaces defined in this document are described as web services. Annex F contains XML schema and Web Service Description Language (WSDL) definitions for the introduced network services.

This document is identical with, and has been reproduced from, IEC 62676-2-31:2019, *Video surveillance systems for use in security applications — Part 2-31: Live streaming and control based on web services*.

As this document has been reproduced from an International Standard, a full point substitutes for a comma when referring to a decimal marker.

Australian or Australian/New Zealand Standards that are identical adoptions of international normative references may be used interchangeably. Refer to the online catalogue for information on specific Standards.

The terms “normative” and “informative” are used in Standards to define the application of the appendices or annexes to which they apply. A “normative” appendix or annex is an integral part of a Standard, whereas an “informative” appendix or annex is only for information and guidance.

NOTES

CONTENTS

FOREWORD.....	8
INTRODUCTION.....	10
1 Scope.....	11
2 Normative references	11
3 Terms and definitions	12
4 Overview	13
4.1 General.....	13
4.2 Device IO.....	13
4.3 Imaging configuration.....	13
4.4 Media configuration	13
4.4.1 Media profiles	13
4.4.2 Video source mode	16
4.5 Real-time streaming.....	16
4.6 PTZ Control	17
4.7 Analytics	18
4.8 Interfaces.....	20
5 Device IO service	20
5.1 General.....	20
5.2 VideoOutputs	20
5.2.1 General	20
5.2.2 GetVideoOutputs	20
5.3 VideoOutputConfiguration	21
5.3.1 GetVideoOutputConfiguration	21
5.3.2 SetVideoOutputConfiguration	21
5.3.3 GetVideoOutputConfigurationOptions	22
5.4 VideoSources	22
5.4.1 General	22
5.4.2 GetVideoSources.....	22
5.5 AudioOutputs	22
5.5.1 General	22
5.5.2 GetAudioOutputs	23
5.6 AudioSources	23
5.6.1 General	23
5.6.2 GetAudioSources.....	23
5.7 Capabilities.....	23
6 Media service	24
6.1 General.....	24
6.2 Media profile methods.....	25
6.2.1 Create media profile	25
6.2.2 Get media profiles	25
6.2.3 Add one or more configurations to a profile.....	26
6.2.4 Remove one or more configurations from a profile	27
6.2.5 Delete media profile.....	27
6.3 Media configurations.....	28
6.3.1 General	28
6.3.2 Video source configuration	28

6.3.3	Video encoder configuration	29
6.3.4	Audio source configuration	29
6.3.5	Audio encoder configuration	29
6.3.6	PTZ Configuration	29
6.3.7	Analytics configuration.....	29
6.3.8	Metadata configuration	30
6.3.9	Audio output configuration	30
6.3.10	Audio decoder configuration	31
6.4	Media Configuration Methods.....	31
6.4.1	General	31
6.4.2	Get configurations	31
6.4.3	Modify a configuration	32
6.4.4	Get configuration options.....	32
6.4.5	GetVideoEncoderInstances.....	33
6.5	GetStreamUri.....	34
6.6	GetSnapshotUri	35
6.7	Multicast	35
6.7.1	General	35
6.7.2	Start multicast streaming	36
6.7.3	Stop multicast streaming	36
6.8	SetSynchronizationPoint	37
6.9	Video source mode	37
6.9.1	General	37
6.9.2	GetVideoSourceModes	37
6.9.3	SetVideoSourceMode	38
6.10	OSD (on-screen display).....	38
6.10.1	General	38
6.10.2	CreateOSD	39
6.10.3	DeleteOSD	40
6.10.4	GetOSDs	40
6.10.5	SetOSD	41
6.10.6	GetOSDOptions.....	41
6.11	Privacy masks.....	42
6.11.1	General	42
6.11.2	CreateMask	43
6.11.3	DeleteMask	43
6.11.4	GetMasks	44
6.11.5	SetMask	44
6.11.6	GetMaskOptions	45
6.12	Capabilities.....	45
6.13	Events	46
6.13.1	ProfileChange.....	46
6.13.2	ConfigurationChange	46
6.13.3	ActiveConnections	47
6.14	Deviations of media service version 1	47
6.14.1	General	47
6.14.2	Profile management.....	47
6.14.3	Configuration listing.....	48
6.14.4	Privacy masks	48

7	Imaging service	48
7.1	General.....	48
7.2	Imaging settings.....	48
7.2.1	Parameters	48
7.2.2	GetImagingSettings	50
7.2.3	SetImagingSettings.....	51
7.2.4	GetOptions	51
7.3	Imaging Presets	52
7.3.1	General	52
7.3.2	GetPresets	52
7.3.3	GetCurrentPreset	52
7.3.4	SetCurrentPreset.....	53
7.4	Focus operations	54
7.4.1	Move	54
7.4.2	GetMoveOptions.....	54
7.4.3	Stop.....	55
7.4.4	GetImagingStatus	55
7.5	Capabilities	56
8	PTZ service	56
8.1	General.....	56
8.2	PTZ node	57
8.2.1	General	57
8.2.2	GetNodes	57
8.2.3	GetNode	57
8.3	PTZ configuration	58
8.3.1	General	58
8.3.2	GetConfigurations.....	59
8.3.3	GetConfiguration	59
8.3.4	GetConfigurationOptions	60
8.3.5	SetConfiguration.....	60
8.3.6	GetCompatibleConfigurations	61
8.4	Move operations	61
8.4.1	General	61
8.4.2	AbsoluteMove.....	61
8.4.3	RelativeMove.....	62
8.4.4	ContinuousMove.....	63
8.4.5	GeoMove	64
8.4.6	Stop.....	66
8.4.7	GetStatus	66
8.5	Preset operations.....	67
8.5.1	General	67
8.5.2	SetPreset	67
8.5.3	GetPresets	68
8.5.4	GotoPreset	69
8.5.5	RemovePreset.....	69
8.6	Home position operations	70
8.6.1	General	70
8.6.2	GotoHomePosition.....	70
8.6.3	SetHomePosition	71

8.7	Auxiliary operations	71
8.7.1	General	71
8.7.2	SendAuxiliaryCommand.....	71
8.8	Predefined PTZ Spaces	72
8.8.1	General	72
8.8.2	Absolute position spaces	72
8.8.3	Relative translation spaces	77
8.8.4	Continuous velocity spaces	78
8.8.5	Speed spaces.....	79
8.9	Preset tour operations.....	80
8.9.1	General	80
8.9.2	GetPresetTours	81
8.9.3	GetPresetTour	81
8.9.4	GetPresetTourOptions	82
8.9.5	CreatePresetTour	82
8.9.6	ModifyPresetTour	83
8.9.7	OperatePresetTour	83
8.9.8	RemovePresetTour.....	84
8.9.9	Preset tour parameters	85
8.10	Pan/tilt control direction configuration	86
8.11	Capabilities.....	87
8.12	Events	88
8.12.1	General	88
8.12.2	PTZ presets.....	88
8.12.3	PresetTours.....	88
9	Analytics service.....	89
9.1	General.....	89
9.2	Scene description interface.....	89
9.2.1	Overview	89
9.2.2	Frame-related content	89
9.2.3	Scene elements.....	92
9.3	Rule interface	99
9.3.1	General	99
9.3.2	Rule representation	100
9.3.3	Rule description language	100
9.3.4	Operations on rules	101
9.4	Analytics modules interface	104
9.4.1	General	104
9.4.2	Analytics module configuration	105
9.4.3	Analytics module description language	105
9.4.4	Operations on analytics modules	105
9.5	GetAnalyticsModuleOptions	108
9.6	Capabilities.....	109
9.7	Events – Audio Detected.....	109
10	Real-time streaming	110
10.1	General.....	110
10.2	Media stream protocol.....	110
10.2.1	Transport format.....	110
10.2.2	Media transport	111

10.2.3	Synchronization points.....	115
10.2.4	JPEG over RTP	116
10.3	Media control protocol.....	118
10.3.1	RTSP stream control	118
10.3.2	Keep-alive method for RTSP session	120
10.3.3	RTSP audio and video synchronization.....	121
10.3.4	RTSP session for a metadata stream.....	121
10.3.5	Multicast streaming.....	122
10.3.6	RTSP message example.....	122
10.3.7	RTSP over HTTP	123
10.4	Back channel connection	123
10.4.1	General	123
10.4.2	RTSP Require tag.....	123
10.4.3	Connection setup for a bi- directional connection.....	124
10.4.4	Describe example for a server without backchannel support:	124
10.4.5	Describe example for a server with ONVIF backchannel support:	124
10.4.6	Multicast streaming.....	126
10.5	Error handling	126
Annex A	(normative) Efficient XML Interchange (EXI)	127
Annex B	(normative) Lens description	128
Annex C	(informative) Specified rules	130
C.1	General.....	130
C.2	LineDetector	130
C.3	FieldDetector	130
C.4	LoiteringDetector	131
C.5	Declarative motion detector	132
C.6	Counting rule	133
C.7	Query rule.....	134
Annex D	(informative) Cell motion detection	135
D.1	Cell motion detector.....	135
D.2	Cell motion analytics engine	136
D.2.1	General	136
D.2.2	Module configuration	137
Annex E	(normative) Motion detection.....	139
Annex F	(normative) Schema files	141
F.1	Device IO.....	141
F.2	Imaging.....	156
F.3	Media.....	162
F.4	Media 2.....	199
F.5	PTZ	221
F.6	Analytics	234
F.7	Common schema	240
F.8	Streaming metadata schema.....	279
Bibliography		284
Figure 1	– A media profile.....	14
Figure 2	– Complete profile configuration.....	15
Figure 3	– Layer structure	16

Figure 4 – Analytics architecture	19
Figure 5 – Example with four OSD configurations	39
Figure 6 – Example of screen with mask and coordinate system	42
Figure 7 – Spherical pan/tilt position space in degrees for a camera mounted on the ceiling	74
Figure 8 – Example of changes of pan/tilt control direction by E-Flip and Reverse	87
Figure 9 – Default frame coordinate system	91
Figure 10 – RTP header.....	111
Figure 11 – RTCP sequence	114
Figure 12 – RTCP Sender Report	115
Figure 13 – Media synchronization.....	115
Figure 14 – RTP/JPEG packet structure.....	116
Figure 15 – Stream control.....	119
Figure 16 – Keep alive	121
Figure B.1 – Optical mapping of angle (α) via radius (R) to normalized x/y coordinates	128
Figure B.2 – Smooth mapping using B-splines	128
Figure B.3 – Compensation of vertical axis offset.....	129
Figure D.1 – CellLayout of an 8 × 6 CellMotionEngine	138
Table 1 – Referenced namespaces (with prefix).....	20
Table 2 – Colourspace namespace values	97
Table 3 – Description of attributes of MotionInCells type	99
Table 4 – RTP header value	112
Table 5 – RTSP methods	120
Table A.1 – ONVIF defined EXI header settings.....	127
Table A.2 – ONVIF defined EXI configuration settings	127
Table C.1 – Loitering Detector rule configuration parameters	132
Table C.2 – Description of loitering event fields	132
Table C.3 – Declarative motion detector rule configuration parameters	133
Table C.4 – Description of declarative motion event fields	133
Table C.5 – Counting rule configuration parameters.....	134
Table C.6 – Description of counting event fields.....	134
Table C.7 – Query Rule configuration parameters	134
Table D.1 – Cell motion detector rule configuration parameters	136
Table D.2 – Description cell motion detected event fields.....	136
Table D.3 – Module configuration parameters	137
Table D.4 – Description of CellLayout fields.....	137
Table E.1 – Motion Region Detector Rule configuration parameters	139
Table E.2 – Motion region detector rule configuration options	140
Table E.3 – Description of the motion region detector event fields	140

INTERNATIONAL ELECTROTECHNICAL COMMISSION

VIDEO SURVEILLANCE SYSTEMS FOR USE IN SECURITY APPLICATIONS –

Part 2-31: Live streaming and control based on web services

FOREWORD

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

International Standard IEC 62676-2-31 has been prepared by IEC technical committee 79: Alarm and electronic security systems.

This first edition, together with IEC 60839-11-31 and IEC 62676-2-32, cancels and replaces IEC 62676-2-3:2013.

This edition includes the following significant technical changes with respect to IEC 62676-2-3:2013:

- a) addition of the Media2 service;
- b) additional methods for the imaging service;
- c) method duplicates from the device IO service have been removed;
- d) both the display and analytics device service are no more included.

This publication contains attached schema files. These files are intended to be used as a complement and do not form an integral part of the standard

The text of this International Standard is based on the following documents:

FDIS	Report on voting
79/620/FDIS	79/622/RVD

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

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

A list of all parts in the IEC 62676 series, published under the general title *Video surveillance systems for use in security applications*, can be found on the IEC website.

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

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

IMPORTANT – The 'colour inside' logo on the cover page of this publication indicates that it contains colours which are considered to be useful for the correct understanding of its contents. Users should therefore print this document using a colour printer.

INTRODUCTION

The goal of this document is to provide a fully interoperable network video implementation comprised of products from different network video vendors. This document describes the network video model, interfaces, data types and data exchange patterns. The document reuses existing relevant standards where available and introduces new specifications only where necessary to support the specific requirements for network video surveillance.

VIDEO SURVEILLANCE SYSTEMS FOR USE IN SECURITY APPLICATIONS –

Part 2-31: Live streaming and control based on web services

1 Scope

This document defines procedures for communication between network video clients and video transmitter devices. This new set of specifications makes it possible to build network video systems with devices and receivers from different manufacturers using common and well-defined interfaces. These interfaces cover functions such as media and imaging configuration, real-time streaming of audio and video, pan, tilt and zoom (PTZ) control as well as analytics.

The management and control interfaces defined in this document are described as web services. Annex F contains XML schema and Web Service Description Language (WSDL) definitions for the introduced network services.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

IEC 60839-11-31, *Alarm and electronic security systems – Part 11-31: Electronic access control systems – Core interoperability protocol based on Web services*

ISO 12639, *Graphic technology – Prepress digital data exchange – Tag image file format for image technology (TIFF/IT)*

INTERNET ENGINEERING TASK FORCE (IETF). RFC 1952: *GZIP file format specification version 4.3* [online]. Edited by P. Deutsch. May 1996 [viewed 2019-01-08]. Available at <http://tools.ietf.org/html/rfc1952>

INTERNET ENGINEERING TASK FORCE (IETF). RFC 2326: *Real Time Streaming Protocol (RTSP)* [online]. Edited by H. Schulzrinne et al. April 1998 [viewed 2019-01-08]. Available at <http://www.ietf.org/rfc/rfc2326.txt>

INTERNET ENGINEERING TASK FORCE (IETF). RFC 2435: *RTP Payload Format for JPEG-compressed Video* [online]. Edited by L. Berc et al. October 1998 [viewed 2019-01-08]. Available at <http://www.ietf.org/rfc/rfc2435.txt>

INTERNET ENGINEERING TASK FORCE (IETF), RFC 2818: *HTTP over TLS* [online]. Edited by E. Rescorla. May 2000 [viewed 2019-01-08]. Available at <http://www.ietf.org/rfc/rfc2818.txt>

INTERNET ENGINEERING TASK FORCE (IETF), RFC 3016: *RTP Payload Format for MPEG-4 Audio/Visual Streams* [online]. Edited by Y. Kikuchi et al. November 2000 [viewed 2019-01-08]. Available at <http://www.ietf.org/rfc/rfc3016.txt>

INTERNET ENGINEERING TASK FORCE (IETF), RFC 3550: *RTP: A Transport Protocol for Real-Time Applications* [online]. Edited by H. Schulzrinne et al. July 2003 [viewed 2019-01-08]. Available at <http://www.ietf.org/rfc/rfc3550.txt>

INTERNET ENGINEERING TASK FORCE (IETF), RFC 3551: *RTP Profile for Audio and Video Conferences with Minimal Control* [online]. Edited by H. Schulzrinne et al. July 2003 [viewed 2019-01-08]. Available at <http://www.ietf.org/rfc/rfc3551.txt>

INTERNET ENGINEERING TASK FORCE (IETF), RFC 3640: *RTP Payload Format for Transport of MPEG-4 Elementary Streams* [online]. Edited by J. van der Meer et al. November 2003 [viewed 2019-01-08]. Available at <http://www.ietf.org/rfc/rfc3640.txt>

INTERNET ENGINEERING TASK FORCE (IETF), RFC 3984: *RTP Payload Format for H.264 Video* [online]. Edited by T. Stockhammer et al. February 2005 [viewed 2019-01-08]. Available at <http://www.ietf.org/rfc/rfc3984>

INTERNET ENGINEERING TASK FORCE (IETF), RFC 4566: *SDP: Session Description Protocol* [online]. Edited by M. Handley et al. July 2006 [viewed 2019-01-08]. Available at <http://www.ietf.org/rfc/rfc4566.txt>

INTERNET ENGINEERING TASK FORCE (IETF), RFC 6455: *The WebSocket Protocol* [online]. Edited by I. Fette et al. December 2011 [viewed 2019-01-08]. Available at <http://www.ietf.org/rfc/rfc6455.txt>

INTERNET ENGINEERING TASK FORCE (IETF), RFC 7798: *RTP Payload Format for High Efficiency Video Coding (HEVC)* [online]. Edited by Y.-K. Wang et al. March 2016 [viewed 2019-01-08]. Available at <http://www.ietf.org/rfc/rfc7798.txt>

Apple, *Tunneling QuickTime RTSP and RTP over HTTP*

Available at

http://www.opensource.apple.com/source/QuickTimeStreamingServer/QuickTimeStreamingServer-412.42/Documentation/RTSP_Over_HTTP.pdf¹

3 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

- IEC Electropedia: available at <http://www.electropedia.org/>
- ISO Online browsing platform: available at <http://www.iso.org/obp>

3.1

media profile

profile that maps a video or an audio source or an audio output to a video or an audio encoder, an audio decoder configuration and PTZ and analytics configurations

¹ QuickTime is the trademark of a product supplied by Apple Inc. This information is given for the convenience of users of this document and does not constitute an endorsement by IEC of the product named. Equivalent products may be used if they can be shown to lead to the same results.

3.2

metadata

all streaming data except video and audio, including video analytics results, PTZ position data and other metadata (such as textual data from POS applications)

3.3

PTZ node

low-level PTZ entity that maps to the PTZ device and its capabilities

3.4

analytics

algorithms or programs used to analyze video and audio data to generate a high-level data description

4 Overview

4.1 General

This specification defines a standardized interface for video surveillance cameras and encoders based on the basic web service communication mechanism defined in IEC 60839-11-31. The services in this specification define the technical interface details for devices and clients in order to be compliant to ONVIF Profile S and ONVIF Profile T.

4.2 Device IO

This specification extends the Device IO definitions from IEC 60839-11-31 by adding interfaces to retrieve the following entities:

- VideoOutputs
- VideoSources
- AudioOutputs
- AudioSources

4.3 Imaging configuration

The imaging service provides configuration and control data for imaging-specific properties. WSDL is part of the framework and provided in the imaging WSDL file.

The service includes the following operations:

- Get and set imaging configurations such as exposure time, gain and white balance.
- Get imaging configuration options that include valid ranges for imaging parameters.
- Move a focus lens.
- Get current position and move status of a focus lens.
- Retrieve and configure imaging presets.

4.4 Media configuration

4.4.1 Media profiles

Media configurations are handled through the media service. Media configurations are used to determine the streaming properties of requested media streams as defined in this specification. Sets of media configurations are called media profiles. See Figure 1 for an abstract example.

Real-time video and audio streaming configurations are controlled using media profiles. A media profile maps a video and/or audio source to a video and/or an audio encoder, PTZ and

analytics configurations. An ONVIF compliant device supporting the media service presents different available profiles depending on its capabilities (the set of available profiles can change dynamically, though).

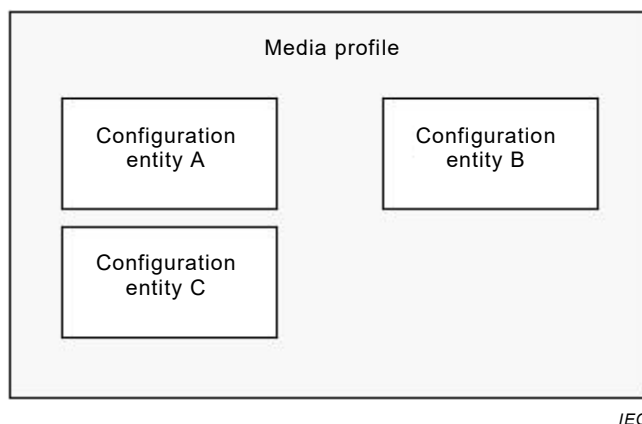


Figure 1 – A media profile

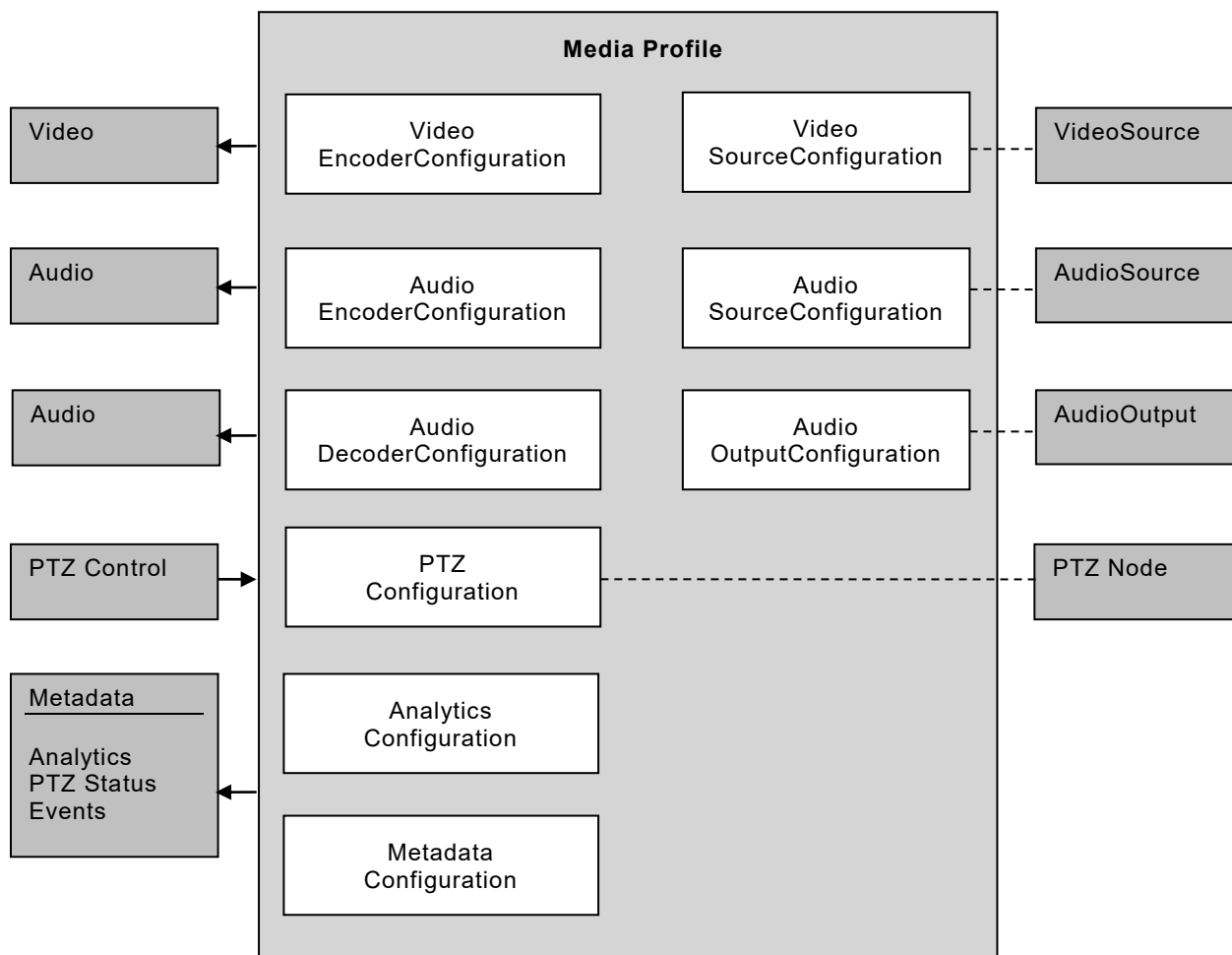
A mediaprofile consists of a set of interconnected configuration entities as depicted in Figure 1. Configurations are provided by the device and can be either static or created dynamically by the device. For example, the dynamic configurations can be created by the device depending on current available encoding resources. A configuration entity is one of the following:

- video source configuration;
- audio source configuration;
- video encoder configuration;
- audio encoder configuration;
- PTZ configuration;
- analytics configuration;
- metadata configuration;
- audio output configuration;
- audio decoder configuration.

A profile consists of all or a subset of these configuration entities. Depending on the capabilities of the device, a particular configuration entity can be part of a profile or not. For example, a profile with an audio source and an audio encoder configuration can exist only in a device with audio support.

A device may provide "ready-to-use" profiles for the most common media configurations that the device offers. The profile contains a "fixed" attribute that indicates if a profile can be deleted or not. The fixed attribute does not signal that a profile is immutable. Hence, it shall be possible to add or remove configurations to or from a fixed profile. Whether a profile is fixed or not is defined by the device.

An example of a complete profile configuration is illustrated in Figure 2.



IEC

Figure 2 – Complete profile configuration

A media profile describes how and what to present to the client in a media stream as well as how to handle PTZ input and analytics.

The following commands manage media profiles:

- **CreateProfile** – Creates a new media profile.
- **GetProfiles** – Get one or all existing media profiles.
- **DeleteProfile** – Deletes a specific media profile.
- **AddConfiguration** – Adds or replaces configuration entities of a media profile.
- **RemoveConfiguration** – Removes one or more configuration entities from a media profile.

The following commands manage configuration entities:

- **Get<entity>ConfigurationOptions** – Gets the valid property values for a specific configuration entity.
- **Get<entity>Configurations** – Gets one or more configuration entities. The client may request a specific configuration by providing the configuration token. It may also specify a profile token to get all compatible configurations. If no token is specified, all existing configurations are returned.
- **Set<entity>Configuration** – Updates the settings of a configuration entity.

Where <entity> is the type of configuration entity. For example, the complete command to get a video encoder configuration is GetVideoEncoderConfiguration.

The following commands initiate and manipulate a video/audio stream:

- GetStreamUri – Requests a valid streaming URI for a specific media profile and protocol.
- StartMulticastStreaming – Starts multicast streaming using a specified media profile.
- StopMulticastStreaming – Stops a multicast stream.
- SetSynchronizationPoint – Inserts a synchronization point (I-frame, etc.) in active streams.
- GetSnapshotUri – Requests a URI for a specific media profile that can be used to obtain JPEG snapshots.

4.4.2 Video source mode

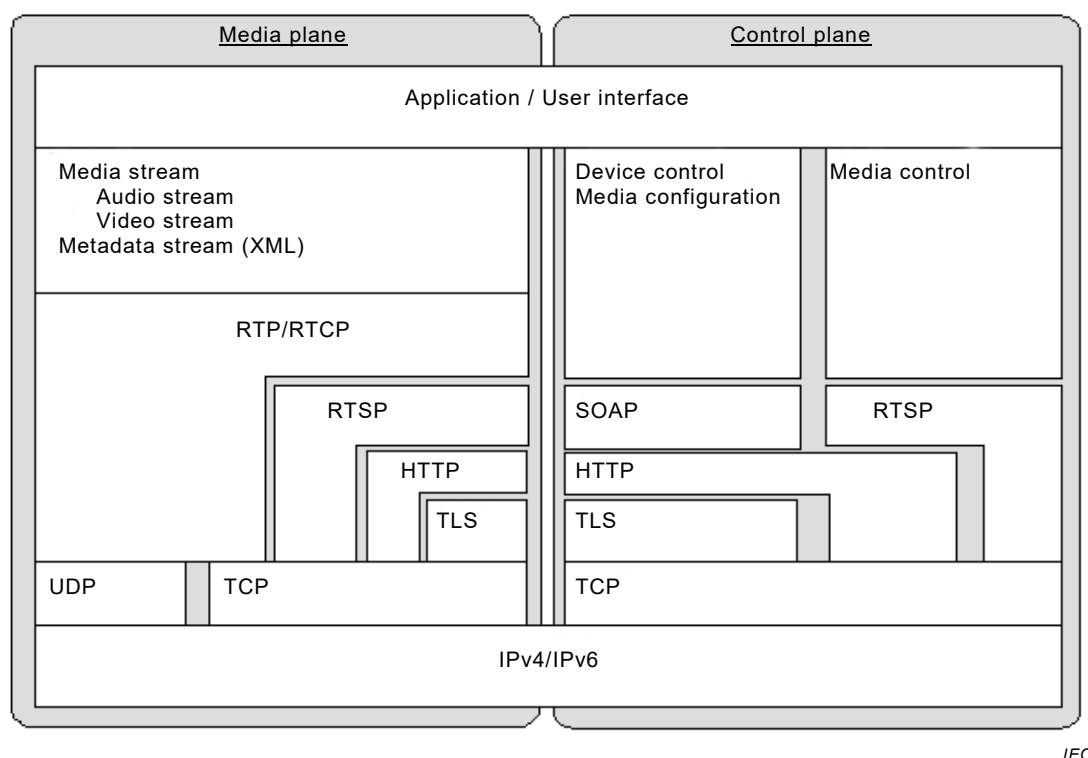
A device may support changing of the video source mode. A change of the video source mode typically requires a reboot of the device owing to hardware dependencies. For example, a device's capability for maximum resolution (1920 × 1080 @ 16:9 or 2048 × 1536 @ 4:3) and frame rate (20 fps or 30 fps) can be changed by selecting each video source mode.

The following commands manage the video source mode:

- GetVideoSourceModes – Get a list of video source modes.
- SetVideoSourceMode – Set video source mode to the specified mode.

4.5 Real-time streaming

This document defines media streaming options and formats. A distinction is made between the media plane and the control plane, as illustrated in Figure 3. A set of media streaming (audio, video and metadata) options, all based on RTP [RFC 3550], are described in order to provide interoperable media streaming services.



IEC

Figure 3 – Layer structure

The metadata streaming container format allows well-defined, real-time streaming of analytics, PTZ status and notification data.

Media configuration is done over SOAP/HTTP and is covered by the media configuration service as discussed in 4.6.

Media control is accomplished over RTSP as defined in RFC 2326. This document uses RTP, RTCP and RTSP profiling, as well as JPEG over RTP extensions and multicast control mechanisms.

This document introduces extensions to RFC 2326 to allow bi-directional streaming connections.

Streaming configurations for the following video codecs are provided:

- JPEG over RTP;
- MPEG-4 [ISO/IEC 14496-2];
- H.264 [ISO/IEC 14496-10];
- H.265 [ISO/IEC 23008-2];

and for the following audio codecs:

- G.711 [ITU-T G.711];
- G.726 [ITU-T G.726];
- AAC [ISO/IEC 14496-3].

4.6 PTZ Control

The PTZ model groups the possible movements of the PTZ unit into a pan/tilt component and into a zoom component. To steer the PTZ unit, the service provides absolute move, relative move and continuous move operations. Different coordinate systems and units are used to feed these operations.

The PTZ service provides an AbsoluteMove operation to move the PTZ device to an absolute position. The service expects the absolute position as an argument referencing an absolute coordinate system. The speed of the pan/tilt movement and the zoom movement can be specified optionally. Speed values are positive scalars and do not contain any directional information. It is not possible to specify speeds for pan and tilt separately without knowledge about the current position. This approach to specifying a desired position generally produces a non-smooth and non-intuitive action.

A RelativeMove operation is introduced by the PTZ service in order to steer the dome relative to the current position, but without the need to know the current position. The operation expects a positional translation as an argument referencing a relative coordinate system. This specification distinguishes between relative and absolute coordinate systems, since there are cases where no absolute coordinate system exists for a well-defined relative coordinate system. An optional speed argument can be added to the RelativeMove operation with the same meaning as for the AbsoluteMove operation.

Finally, the PTZ device can be moved continuously via the ContinuousMove command in a certain direction with a certain speed. Thereby, a velocity vector represents both the direction and the speed information. The latter is expressed by the length of the vector.

The pan/tilt and zoom coordinates can be uniquely specified by augmenting the coordinates with appropriate space URIs. A space URI uniquely represents the underlying coordinate system. Subclause 8.8 defines a standard set of coordinate systems. A PTZ node shall implement these coordinate systems if the corresponding type of movement is supported by

the PTZ node. In many cases, the pan/tilt position is represented by pan and tilt angles in a spherical coordinate system. A digital PTZ, operating on a fixed megapixel camera, may express the camera's viewing direction by a pixel position on a static projection plane. Therefore, different coordinate systems are needed in this case in order to capture the physical or virtual movements of the PTZ device. Optionally, the PTZ node may define its own device specific coordinate systems to enable clients to take advantage of the specific properties of this PTZ node.

The PTZ node description retrieved via the `GetNode` or `GetNodes` operation contains all coordinate systems supported by a specific PTZ Node. Each coordinate system belongs to one of the following groups:

- `AbsolutePanTiltPositionSpace`;
- `RelativePanTiltTranslationSpace`;
- `ContinuousPanTiltVelocitySpace`;
- `PanTiltSpeedSpace`;
- `AbsoluteZoomPositionSpace`;
- `RelativeZoomTranslationSpace`;
- `ContinuousZoomVelocitySpace`;
- `ZoomSpeedSpace`.

If the PTZ node does not support the coordinate systems of a certain group, the corresponding move operation will not be available for this PTZ node. For instance, if the list does not contain an `AbsolutePanTiltPositionSpace`, the `AbsoluteMove` operation shall fail when an absolute pan/tilt position is specified. The corresponding command section describes those spaces that are required for a specific move command.

4.7 Analytics

Analytic applications are divided into image/audio analysis and application-specific parts. The interface between these two parts produces an abstraction that describes the scene based on the objects present. The application-specific part performs a comparison of the scene descriptions and of the scene rules (such as virtual lines whose crossing is prohibited, or polygons that define a protected area). Other rules can represent intra-object behaviour, such as objects following other objects (to form a tailgating detection). Such rules can also be used to describe prohibited object motion, which can be used to establish a speed limit.

These two separate parts, referred to as the video/audio analytics engine and as the rule engine, together with the events and actions, form the analytics architecture according to this specification as illustrated in Figure 4.

The analytics architecture consists of elements and interfaces. Each element provides a functionality corresponding to a semantically unique entity of the complete video/audio analytics solution. Interfaces are unidirectional and define an information entity with a unique content. Only the interfaces are subject to this specification. Central to this architecture is the ability to distribute any elements or sets of adjacent elements to any device in the network.

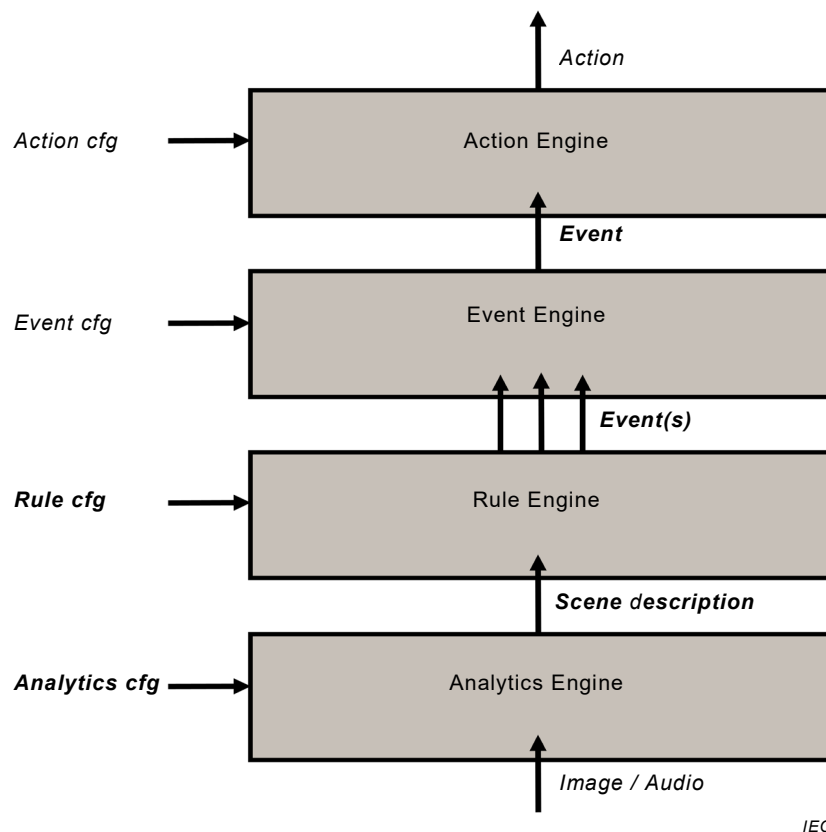


Figure 4 – Analytics architecture

The following interfaces are defined in this document:

- analytics configuration interface;
- scene description;
- rule configuration interface.

This specification defines a configuration framework for the analytics engine, called ONVIF Analytics Service. This framework enables a client to contact a device implementing the ONVIF Analytics Service for supported analytics modules and their configurations. Configurations of such modules can be dynamically added, removed or modified by a client, allowing a client to run multiple analytics modules in parallel if supported by the device.

The output from the analytics engine is called scene description. The scene description represents the abstraction of the scene in terms of metadata for the objects, either static or dynamic, that are part of the scene. This specification defines an XML-based scene description interface including data types.

Rules describe how the scene description is interpreted and how to react on that information. The specification defines standard rule syntax and methods to communicate configuration of these rules from the application to the device.

A device supporting ONVIF Analytics Service shall implement the scene description interface and allow events to be dispatched using the event service. If the device additionally supports a rule engine, then it shall implement the rules and analytics modules interface.

Event and action engine interfaces and configuration is outside of the scope of this specification. The event interface is handled through the Event Service as described in IEC 60839-11-31.

A video analytics configuration can be attached to a media profile if the ONVIF Media Service is present. In that case, the video analytics configuration becomes connected to a specific video source.

4.8 Interfaces

Table 1 lists the prefix and namespaces used in this specification. For interfaces defined by this document, the relevant clause of Annex F is given. Listed prefixes are not part of this document and an implementation can use any prefix.

Table 1 – Referenced namespaces (with prefix)

Prefix	Namespace URI	Reference
env	http://www.w3.org/2003/05/soap-envelope	
ter	http://www.onvif.org/ver10/error	IEC 60839-11-31
xs	http://www.w3.org/2001/XMLSchema	
tt	http://www.onvif.org/ver10/schema	Clause F.7
tmd	http://www.onvif.org/ver10/deviceIO/wsdI	Clause F.1
tr2	http://www.onvif.org/ver20/media/wsdI	Clause F.4
timg	http://www.onvif.org/ver20/imaging/wsdI	Clause F.2
tpz	http://www.onvif.org/ver20/ptz/wsdI	Clause F.5
axt	http://www.onvif.org/ver20/analytics	Clause F.6

5 Device IO service

5.1 General

This service offers commands to retrieve and configure the physical inputs and outputs of a device.

A device that has physical sources and outputs shall support this service. For the schema definition, refer to Clause F.1.

5.2 VideoOutputs

5.2.1 General

The VideoOutput type represents the physical video outputs of a device that can be connected to a monitor to display the video signal. The structure contains the layout settings that can be configured using the Display Service.

5.2.2 GetVideoOutputs

This command lists all available video outputs of a device. A device that has one or more physical video outputs shall support the listing of available video outputs through the GetVideoOutputs command.

REQUEST:

This is an empty message.

RESPONSE:

- **Token – optional, unbounded [tt:ReferenceToken]**

List of tokens enumerating all available video outputs of the device.

FAULTS:

None

ACCESS CLASS:

READ_MEDIA**5.3 VideoOutputConfiguration****5.3.1 GetVideoOutputConfiguration**

This operation requests the configuration of a video output. A device that has one or more video outputs shall support the retrieval of the VideoOutputConfiguration through this command.

REQUEST:

- **VideoOutputToken [tt:ReferenceToken]**
Existing reference of a video output.

RESPONSE:

- **Configuration [tt:VideoOutputConfiguration]**
The configuration of the video output.

FAULTS:

- **env:Sender – ter:InvalidArgVal – ter:NoVideoOutput**
The requested Video Output does not exist.

ACCESS CLASS:

READ_MEDIA**5.3.2 SetVideoOutputConfiguration**

This operation modifies a video output configuration. A device that has one or more video outputs shall support the setting of its video output configuration through this command.

REQUEST:

- **Configuration [tt:VideoOutputConfiguration]**
The configuration of the video output.
- **ForcePersistence [xs:boolean]**
Deprecated modifier for temporary settings if supported by the device.

RESPONSE:

This is an empty message.

FAULTS:

- **env:Sender – ter:InvalidArgVal – ter:NoVideoOutput**
The requested Video Output does not exist.
- **env:Sender – ter:InvalidArgVal – ter:ConfigModify**
The configuration parameters cannot be set.

ACCESS CLASS:

ACTUATE

5.3.3 GetVideoOutputConfigurationOptions

This operation requests the VideoOutputConfigurationOptions of a VideoOutput. A device that has one or more video outputs shall support the retrieval of VideoOutputConfigurationOptions through this command.

REQUEST:

- **VideoOutputToken [tt:ReferenceToken]**

Existing reference of a video output.

RESPONSE:

- **VideoOutputOptions [tt:VideoOutputConfigurationOptions]**

The configuration options for the video output.

FAULTS:

- **env:Sender – ter:InvalidArgVal – ter:NoVideoOutput**

The requested video output does not exist.

5.4 VideoSources

5.4.1 General

A VideoSource represents physical video input. The structure contains the pixel resolution of the video, the framerate and imaging settings. The imaging settings can be manipulated through the ImagingService if supported and contains parameters for focus, exposure and brightness, for example.

5.4.2 GetVideoSources

This operation lists all available video sources for the device. The device that has one or more video inputs shall support the listing of available video sources through the GetVideoSources command.

REQUEST:

This is an empty message.

RESPONSE:

- **Token – optional, unbounded [tt:ReferenceToken]**

List of tokens enumerating all available video sources of the device.

FAULTS:

None

ACCESS CLASS:

READ_MEDIA

5.5 AudioOutputs

5.5.1 General

An AudioOutput represents the physical audio outputs that can be connected to a loudspeaker.

5.5.2 GetAudioOutputs

This command lists all available audio outputs of a device. A device that has one or more physical audio outputs shall support listing of available audio outputs through the GetAudioOutputs command.

REQUEST:

This is an empty message.

RESPONSE:

- **Token – optional, unbounded [tt:ReferenceToken]**
List of tokens enumerating all available audio outputs of the device.

FAULTS:

None

ACCESS CLASS:

READ_MEDIA

5.6 AudioSources

5.6.1 General

An AudioSource represents unencoded audio input and states the number of input channels.

5.6.2 GetAudioSources

This operation lists all available audio sources for the device. The device that has one or more audio sources shall support the listing of available audio inputs through the GetAudioSources command.

REQUEST:

This is an empty message.

RESPONSE:

- **Token – optional, unbounded [tt:ReferenceToken]**
List of tokens enumerating all available audio sources of the device.

FAULTS:

None

ACCESS CLASS:

READ_MEDIA

5.7 Capabilities

The capabilities reflect optional functions and functionality of a service. The information is static and does not change during device operation. The following capabilities are available:

VideoSources:	number of video sources (defaults to none).
VideoOutputs:	number of video outputs (defaults to none).
AudioSources:	number of audio sources (defaults to none).
AudioOutputs:	number of audio outputs (defaults to none).

RelayOutputs:	number of relay outputs (defaults to none).
DigitalInputs:	number of digital inputs (defaults to none).
SerialPorts:	number of serial ports (defaults to none).
DigitalInputOptions:	indicates support for DigitalInput configuration of the idle state (defaults to false)

REQUEST:

This is an empty message.

RESPONSE:

- **Capabilities [tmd:Capabilities]**
List of capabilities as defined above.

FAULTS:

None

ACCESS CLASS:

PRE_AUTH

6 Media service

6.1 General

The media service is used to configure the device media streaming properties.

The media service allows a client to configure media and other real time streaming configurations. Media configurations are handled through media profiles. An overview of the ONVIF media configuration model is given in 4.4.1.

The media service commands are divided into two major categories:

- Media configuration:
 - media profile commands,
 - video source commands,
 - video encoder commands,
 - audio source commands,
 - audio encoder commands,
 - video analytics commands,
 - metadata commands,
 - audio output commands,
 - audio decoder commands.
- Media streaming:
 - request stream URI,
 - get snapshot URI,
 - multicast control commands,
 - media synchronization point.

A basic set of operations are required for the media service; other operations are recommended to be supported. The detailed requirements are listed under the command descriptions.

Subclauses 6.2 to 6.13 describe the Media2 service. For the related schema, refer to the definitions in Clause F.4. For deviations of the Media1 service, see 6.14.

6.2 Media profile methods

6.2.1 Create media profile

This operation creates a new media profile. The media profile shall be created in the device.

A device implementing this service shall support the creation of media profiles as long as the number of existing profiles does not exceed the capability value `MaximumNumberOfProfiles`.

A created profile shall be deletable and a device shall set the "fixed" attribute to false in the returned profile.

REQUEST:

- **Name [tt:Name]**
Name of the new profile.
- **Configuration – optional, unbounded [tr2:ConfigurationRef]**
Optional list of configurations to be added to the new profile.

RESPONSE:

- **Token [tt:ReferenceToken]**
Token assigned to the newly created profile.

FAULTS:

- **env:Receiver – ter:Action – ter:MaxNVTPProfiles**
The maximum number of supported profiles supported by the device has been reached.
- **env:Receiver – ter:Action – ter:ConfigurationConflict**
Other configurations of the media profile conflict with the one to add and adding it would cause a conflicting media profile.
- **env:Sender – ter:InvalidArgVal – ter:NoConfig**
One of the provided configurations indicated by their ConfigurationToken does not exist.

ACCESS CLASS:

ACTUATE

6.2.2 Get media profiles

An endpoint can ask for the existing media profiles of a device using the `GetProfiles` command. Both pre-configured and dynamically created profiles can be retrieved using this command.

The Token parameter controls which profiles are returned:

- If no Token is provided, this command lists all configured profiles of a device.
- If a Token is provided, the command either lists the referenced profile or responds with an error.

The Type parameter controls which configurations are returned and has no effect on the number of profiles returned:

- If no Type is provided, the returned profiles shall contain no configuration information.
- If a single Type with value 'All' is provided, the returned profiles shall include all associated configurations.
- Otherwise, the requested list of configurations shall, for each profile, include the configurations present as Type.

A device implementing this service shall support the retrieval of media profiles through the GetProfiles command.

REQUEST:

- **Token – optional [tt:ReferenceToken]**
Optional token to retrieve exactly one profile.
- **Type – optional, unbounded [xs:string]**
If one or more types are passed, only the corresponding configurations will be returned.

RESPONSE:

- **Profiles – optional, unbounded [tr2:MediaProfile]**
List of profiles. Each profile contains a set of configuration entities defining a specific configuration that can be used for media streaming, analytics, metadata streaming, etc.

FAULTS:

- **env:Sender – ter:InvalidArgVal – ter:NoProfile**
The requested profile token does not exist.

ACCESS CLASS:

READ_MEDIA

NOTE A client can request an enumeration of profiles without any configuration details by not passing a Type parameter.

6.2.3 Add one or more configurations to a profile

This operation adds one or more configurations to an existing media profile. If one of the configurations already exists in the media profile, it will be replaced. A device supporting the Media2 service shall support this command.

REQUEST:

- **ProfileToken [tt:ReferenceToken]**
Token of an existing profile.
- **Name – optional [tt:Name]**
The device shall update the name of the profile when this option is provided.
- **Configuration – optional, unbounded [tr2:ConfigurationRef]**
Optional list of configurations to be added to the profile.

RESPONSE:

This is an empty message.

FAULTS:

- **env:Sender – ter:InvalidArgVal – ter:NoProfile**
The requested profile token does not exist.

- **env:Sender – ter:InvalidArgVal – ter:NoConfig**
One of the provided configurations indicated by their ConfigurationToken does not exist.
- **env:Receiver – ter:Action – ter:ConfigurationConflict**
Other configurations of the media profile conflicts with the one to add and adding it would cause a conflicting media profile.

ACCESS CLASS:

ACTUATE

6.2.4 Remove one or more configurations from a profile

This operation removes one or more configurations from an existing media profile. Removing a non-existing configuration shall be ignored and not result in an error. A device supporting the Media2 service shall support this command.

A device shall return an error if a client attempts to remove VideoSourceConfiguration from the Profile when a VideoEncoderConfiguration is present. To avoid this error condition, a client may first remove VideoEncoderConfiguration, or may remove VideoSourceConfiguration and VideoEncoderConfiguration together in a single RemoveConfiguration command.

A device shall return an error if a client attempts to remove AudioSourceConfiguration from the Profile when an AudioEncoderConfiguration is present. To avoid this error condition, a client may first remove AudioEncoderConfiguration, or may remove AudioSourceConfiguration and AudioEncoderConfiguration together in a single RemoveConfiguration command.

REQUEST:

- **ProfileToken [tt:ReferenceToken]**
Token of an existing profile.
- **Configuration – unbounded [tr2:ConfigurationRef]**
List of configurations to be removed from the profile.

RESPONSE:

This is an empty message.

FAULTS:

- **env:Sender – ter:InvalidArgVal – ter:NoProfile**
The requested profile token does not exist.
- **env:Receiver – ter:Action – ter:ConfigurationConflict**
Other configurations of the media profile are dependant on this configuration and removing it would cause a conflicting state of the media profile.

ACCESS CLASS:

ACTUATE

6.2.5 Delete media profile

This operation deletes a profile. The device shall support the deletion of a media profile through the DeleteProfile command.

REQUEST:

- **Token [tt:ReferenceToken]**

Token of an existing profile.

RESPONSE:

This is an empty message.

FAULTS:

- **env:Sender – ter:InvalidArgVal – ter:NoProfile**
The requested profile token does not exist.
- **env:Sender – ter:Action – ter:DeletionOfFixedProfile**
A fixed Profile cannot be deleted.

ACCESS CLASS:

ACTUATE

6.3 Media configurations

6.3.1 General

A media profile consists of a set of media configurations. Media profiles are used by a client to configure properties of a media stream from a device.

A device shall provide at least one media profile at boot. A device should provide "ready-to-use" profiles for the most common media configurations that the device offers.

A profile consists of a set of interconnected configuration entities. Configurations are provided by the device and can either be static or created dynamically by the device. For example, the dynamic configurations can be created by the device depending on current available encoding resources. Subclauses 6.3.2 to 6.3.10 enumerate the available configurations, each defining entity-specific constraints. A profile consists of all or a subset of these configuration entities. Depending on the capabilities of the device, a particular configuration entity can be part of a profile or not. For example, a profile with an audio source and an audio encoder configuration can exist only in a device with audio support.

All configurations have the following base parameters

- Token – The identifier of the configuration. This parameter is read-only and cannot be modified by clients.
- Name – A configurable name of up to 64 characters.
- UseCount – Read-only property indicating the number of referenced profiles. Deprecated in Media2; devices may support this value.

6.3.2 Video source configuration

VideoSourceConfiguration contains a reference to VideoSource and a Bounds structure containing either the whole VideoSource pixel area or a sub-portion of it. Bounds and VideoSource define the image that is streamed to a client.

The Rotate option specifies an optional rotation of the area defined by the Bounds parameters. Note that, in case of, for example, a 90° rotation, the width parameter corresponds to the height of the video and vice versa.

All coordinate systems (for example Privacy Masks in the Media2 Service and Motion Regions in the Analytics service) that apply to a video source configuration are based on the resulting image after applying the Bounds structure and then Rotate option to the source image.

The LensDescription option allows the geometric distortion of the Video Source to be described. For details, see Annex B.

The SceneOrientation options allow a description of the orientation of the scene the video source is capturing. The Scene Orientation can be Below (from the ceiling), Above (from the floor or a table) and Horizon (on a wall). Some devices may support detecting the SceneOrientation option automatically.

6.3.3 Video encoder configuration

VideoEncoderConfiguration contains the following parameters for configuring the encoding of video data:

- Encoding – The encoding used to compress the video. See the IANA MIME Type Name reference for a full list of values.
- Resolution – The pixel resolution of the encoded video data.
- Quality – Determines the quality of the video. A high value within supported quality range means higher quality.
- RateControl – Defines parameters to configure the bitrate [kbps] as well as an EncodingInterval parameter (interval at which images are encoded and transmitted) and a TargetFrameRate [fps] parameter to configure the output framerate.
- Encoding profile and GOV length [frame].

The structure of VideoEncoderConfiguration also contains multicast parameters.

If the whole RateControl parameter structure is missing, the current state of rate control is undefined and vendor-specific. A device that supports disabling of the rate control mechanisms shall reflect that by omitting the RateControl element when disabled; otherwise it shall return the current values used for RateControl. If RateControl is missing, the respective options define whether a RateControl element can be (re-)added.

6.3.4 Audio source configuration

AudioSourceConfiguration contains a reference to AudioSource.

6.3.5 Audio encoder configuration

AudioEncoderConfiguration contains the following parameters for encoding audio data:

- Encoding – The encoding used to compress audio. See the IANA MIME Type Name reference for a full list of values.
- Bitrate – The output bitrate [kbps].
- SampleRate – The output sample rate [kHz].

The structure of AudioEncoderConfiguration also contains multicast parameters.

6.3.6 PTZ Configuration

A profile with a PTZ Configuration enables the streaming of PTZ status in the metadata stream. Additionally, the media profile can be used for controlling PTZ movement as defined in the PTZ service specification.

6.3.7 Analytics configuration

A profile containing AnalyticsConfiguration enables streams using that media profile to contain analytics data (in the metadata) as defined by the submitted configuration reference. For the configuration of analytics, refer to the ONVIF Analytics Service Specification.

A device shall return an error (ter:NoConfig) if a client attempts to add VideoAnalyticsConfiguration to the profile when the required source configuration (e.g. VideoSourceConfiguration) is not present. To avoid this error condition, a client may first add a required source configuration (e.g. VideoSourceConfiguration), or may add a required source configuration (e.g. VideoSourceConfiguration) and VideoAnalyticsConfiguration together in a single AddConfiguration command. A device can deny adding a video analytics configuration before a video source configuration. In this case, it should respond with a ConfigurationConflict fault.

When an analytics configuration is present in a profile, the metadata configuration can activate the streaming of the scene description within the RTP streams (see 6.3.8).

A device may not allow referencing the very same AnalyticsConfiguration from multiple media profiles with different VideoSourceConfigurations. If the device allows it, it shall generate individual scene descriptions for each profile, since the coordinate system of a scene description relates to a specific VideoSourceConfiguration. Also masking and geometrical rules relate to the coordinate system of the VideoSourceConfiguration. This can require separate processing of the whole video analytics for each VideoSourceConfiguration, even if they refer to the very same VideoSource.

6.3.8 Metadata configuration

A profile containing MetadataConfiguration enables the streaming of metadata. Metadata can consist of events, PTZ status, and/or analytics data.

MetadataConfiguration contains parameters for selecting the data to include in the metadata stream. The choices include PTZ status, PTZ position, events as defined by a subscription and analytics data. The event subscription data is described in the section "Event Handling" of the ONVIF Core Specification. The analytics parameters define which data to include from the analytics engine part of the profile; see 6.3.7.

The structure also contains multicast parameters used to configure and control multicast of the metadata stream. Devices supporting compressed metadata shall signal available compression algorithm as defined in the MetadataCompressionType. Currently defined compression types are "GZIP" and "EXI". For details on the EXI configuration, see Annex A.

6.3.9 Audio output configuration

6.3.9.1 General

The audio output configuration contains the following parameters:

- SourceToken: a reference to an existing audio output.
- OutputLevel: a parameter to configure the output volume
- SendPrimacy: a parameter that can be used for devices with a half duplex audio in/output to configure the active transmission direction (see 6.3.9.2).

6.3.9.2 Audio channel modes

An audio channel can support different types of audio transmission. For full duplex operation, no special handling is required. In half duplex operation, the transmission direction needs to be switched.

An optional Send-Primacy parameter inside the AudioOutputConfiguration indicates which direction is currently active. A client can switch between different modes by setting AudioOutputConfiguration.

The following modes for the Send-Primacy are defined:

- www.onvif.org/ver20/HalfDuplex/Server
The server is allowed to send audio data to the client. The client shall not send audio data via the backchannel to the device in this mode.
- www.onvif.org/ver20/HalfDuplex/Client
The client is allowed to send audio data via the backchannel to the server. The device shall not send audio data to the client in this mode.
- www.onvif.org/ver20/HalfDuplex/Auto
It is up to the device how to deal with sending and receiving audio data.

Acoustic echo cancellation is outside the scope of this document.

6.3.10 Audio decoder configuration

The audio decoder configuration does not contain any parameter to configure the decoding. The encodings supported by an audio decoder configuration can be retrieved via the method `GetAudioDecoderConfigurationOptions`.

6.4 Media Configuration Methods

6.4.1 General

For each supported entity, a device shall provide the set of `Get<entity>Configuration`, `Set<entity>Configuration` and `Get<entity>ConfigurationOptions` commands.

6.4.2 Get configurations

The `Get<entity>Configuration` operation allows the actual settings of one or more configurations to be retrieved. The syntax and semantics of the request message are the same for all configuration entities:

- If a configuration token is provided, the device shall respond with the requested configuration or provide an error if it does not exist.
- In case only a profile token is provided, the device shall respond with all configurations that are compatible to the provided media profile.
- If no tokens are provided, the device shall respond with all available configurations.

REQUEST:

- **ProfileToken – optional [tt:ReferenceToken]**
Optional profile token to retrieve all compatible configurations.
- **ConfigurationToken – optional [tt:ReferenceToken]**
Optional token to retrieve exactly one configuration.

RESPONSE:

- **Configurations – optional, unbounded [tt:<entity>Configuration]**
List of configurations.

FAULTS:

- **env:Sender – ter:InvalidArgVal – ter:NoProfile**
The requested profile token does not exist.
- **env:Sender – ter:InvalidArgVal – ter:NoConfig**
The requested configuration does not exist in the given context.

ACCESS CLASS:

READ_MEDIA

6.4.3 Modify a configuration

The Set<entity>Configuration operation modifies a configuration. The change can have an immediate effect on running streams, but the changes are not guaranteed to take effect unless the client restarts any affected stream. Client methods for changing a running stream are out of scope of this specification.

REQUEST:

- **Configuration [tt:<entity>Configuration]**

The Configuration element contains the modified configuration. The configuration shall exist in the device.

RESPONSE:

This message is empty.

FAULTS:

- **env:Sender – ter:InvalidArgVal – ter:NoConfig**
The requested configuration does not exist in the given context.
- **env:Sender – ter:InvalidArgVal – ter:ConfigModify**
The configuration parameters are not possible to set.
- **env:Receiver – ter:Action – ter:ConfigurationConflict**
The new settings conflict with other uses of the configuration.

ACCESS CLASS:

ACTUATE

Note for VideoEncoderConfiguration: if necessary, the device may adapt parameter values for Quality and RateControl elements without returning an error. A device shall adapt an out of range BitrateLimit instead of returning a fault.

Note for the AudioSourceConfiguration: if the new settings invalidate any parameters already negotiated using RTSP, for example, by changing codec type, the device shall not apply these settings to existing streams. Instead it shall either continue to stream using the old settings or stop sending data on the affected streams.

6.4.4 Get configuration options

The Get<entity>ConfigurationOptions operation returns the available parameters and their valid ranges to the client. Any combination of the parameters obtained using a given media profile and configuration shall be a valid input for the corresponding set configuration command.

If a configuration token is provided, the device shall return the options compatible with that configuration. If a media profile token is specified, the device shall return the options compatible with that media profile. If both a media profile token and a configuration token are specified, the device shall return the options compatible with both that media profile and that configuration. If no tokens are specified, the options shall be considered generic for the device.

REQUEST:

- **ConfigurationToken – optional [tt:ReferenceToken]**
Optional token to retrieve exactly one configuration.
- **ProfileToken – optional [tt:ReferenceToken]**
Optional profile token to retrieve all compatible configurations.

RESPONSE:

- **ConfigurationOptions – unbounded [tt:<entity>Configuration]**
Depending on the structure of tr2:Get<entity>ConfigurationOptionsResponse, the options will return one or more elements.

NOTE Entities AudioOutput, AudioSource, Metadata, and VideoSource return just one element.

FAULTS:

- **env:Sender – ter:InvalidArgVal – ter:NoProfile**
The requested profile token does not exist.
- **env:Sender – ter:InvalidArgVal – ter:NoConfig**
The requested configuration does not exist in the given context.

ACCESS CLASS:

READ_MEDIA**6.4.5 GetVideoEncoderInstances**

This command provides information on how many video encoders a device can instantiate concurrently for VideoSourceConfiguration. A device shall support this command.

The Info response contains the following information:

- Total: total number of encoder instances independent of the codec;
- Codec: number of encoder instances for each supported codec.

A device shall guarantee to instantiate the indicated number of instances concurrently. If a device limits the number of instances of each particular video encoding type, the response shall contain information per video codec.

The total sum of video encoder instances over all video sources of a device shall not exceed the value signaled via MaximumNumberOfProfiles.

For example, if a device has two VideoSourceConfigurations and if the first allows a total of two concurrent instances and the second allows only one instance, this device shall allow creation of at least three media profiles.

REQUEST:

- **ConfigurationToken [tt:ReferenceToken]**
Token of the configuration.

RESPONSE:

- **Info [tr2:EncoderInstanceInfo]**
This message contains the minimum guaranteed total number of encoder instances (applications) per VideoSourceConfiguration.

FAULTS:

- **env:Sender – ter:InvalidArgVal – ter:NoConfig**

The requested configuration does not exist in the given context.

ACCESS CLASS:

READ_MEDIA**6.5 GetStreamUri**

This operation requests a URI that can be used to initiate a live media stream using RTSP as the control protocol. The returned URI should remain valid indefinitely, even if the parameters of the profile are changed.

The following stream types are defined:

- **RtspUnicast** – RTSP streaming RTP via UDP Unicast.
- **RtspMulticast** – RTSP streaming RTP via UDP Multicast.
- **RTSP** – RTSP streaming RTP over TCP.
- **RtspOverHttp** – Tunneling of both the RTSP control channel and the RTP stream over HTTP or HTTPS.

For full compatibility with other ONVIF services, a device shall not generate URIs longer than 128 octets.

A device that signals the RTSPStreaming capability shall support this command. On a request for transport protocol **RtspOverHttp**, a device shall return a URI that uses the same port as the web service. This enables seamless NAT traversal.

REQUEST:

- **Protocol [xs:string]**

The Protocol defines how the encoded data is expected to be streamed to the client.

- **ProfileToken [tt:ReferenceToken]**

The ProfileToken element indicates the media profile to use and will define the configuration of the content of the stream.

RESPONSE:

- **Uri [xs:anyUri]**

The stable URI to be used for requesting the media stream.

FAULTS:

- **env:Sender – ter:InvalidArgVal – ter:NoProfile**

The requested profile token does not exist.

- **env:Sender – ter:InvalidArgVal – ter:InvalidStreamSetup**

The specified protocol is not supported.

- **env:Sender – ter:OperationProhibited – ter:StreamConflict**

The specified protocol causes a conflict with another stream.

- **env:Sender – ter:InvalidArgVal – ter:InvalidMulticastSettings**

Not all configurations are configured for multicast.

- **env:Receiver – ter:Action – ter:IncompleteConfiguration**

The specified media profile does contain either unused sources or encoder configurations without a corresponding source.

ACCESS CLASS:

READ_MEDIA

6.6 GetSnapshotUri

A network client uses the GetSnapshotUri command to obtain a JPEG snapshot from the device. The returned URI shall remain valid indefinitely even if the profile parameters change. The URI can be used for acquiring one or more JPEG images through an HTTP GET operation.

The image encoding will always be JPEG regardless of the encoding setting in the media profile. The JPEG settings (such as resolution or quality) should be taken from the profile if suitable. The provided image shall be updated automatically and independently from calls to GetSnapshotUri.

A device shall support this command when the SnapshotUri capability is set to "true".

REQUEST:

- **ProfileToken [tt:ReferenceToken]**

The ProfileToken element indicates the media profile to use and will define the configuration of the content of the stream.

RESPONSE:

- **Uri [xs:anyUri]**

Stable URI to be used for acquiring a snapshot in JPEG format as well as parameters defining the lifetime of the URI.

FAULTS:

- **env:Sender – ter:InvalidArgVal – ter:NoProfile**

The requested profile token does not exist.

- **env:Receiver – ter:Action – ter:IncompleteConfiguration**

The specified media profile does contain either unused sources or encoder configurations without a corresponding source.

ACCESS CLASS:

READ_MEDIA

6.7 Multicast

6.7.1 General

This specification defines two mechanism for controlling multicast streams. In addition to setting up multicast sessions via RTSP sessions as defined in 6.5, Subclause 6.7 defines a multicast streaming mechanism where the actual streaming is controlled via IGMP. Use this method with caution, since an incorrect network configuration can result in flooding the network with audio and video packets.

A device that signals support for non-RTSP controlled multicast streaming by the AutoStartMulticast capability shall support the methods defined in 6.7.

6.7.2 Start multicast streaming

This command starts multicast streaming using a specified media profile of a device. Streaming continues until StopMulticastStreaming is called for the same profile. The streaming shall be resumed after rebooting. It can be turned off using the StopMulticastStreaming method. The multicast address, port and TTL are configured in the VideoEncoderConfiguration, AudioEncoderConfiguration and MetadataConfiguration respectively.

Multicast streaming may stop when the corresponding profile is deleted or one of its configurations is altered via one of the set configuration methods.

The implementation shall ensure that the RTP stream can be decoded without setting up an RTSP control connection. Especially in the case of H.264 video, the SPS/PPS header shall be sent inband.

REQUEST:

- **ProfileToken [tt:ReferenceToken]**

The ProfileToken element indicates the media profile to use and will define the configuration of the content of the stream.

RESPONSE:

This message is empty.

FAULTS:

- **env:Sender – ter:InvalidArgVal – ter:NoProfile**

The requested profile token does not exist.

- **env:Receiver – ter:Action – ter:IncompleteConfiguration**

The specified media profile does not contain a reference to a video encoder, a video source configuration, to an audio source or to an audio encoder configuration or a reference to a metadata configuration.

ACCESS CLASS:

ACTUATE

6.7.3 Stop multicast streaming

This command stops multicast streaming using a specified media profile of a device. In case a device receives a StopMulticastStreaming request whose corresponding multicast streaming is not started, the device should reply with successful StopMulticastStreamingResponse.

REQUEST:

- **ProfileToken [tt:ReferenceToken]**

The ProfileToken element indicates the media profile to use.

RESPONSE:

This message is empty.

FAULTS:

- **env:Sender – ter:InvalidArgVal – ter:NoProfile**

The requested profile token does not exist.

ACCESS CLASS:

ACTUATE

6.8 SetSynchronizationPoint

Synchronization points allow clients to decode and correctly use all data after the synchronization point.

For example, if a video stream is configured with a large I-frame distance and a client loses a single packet, the client does not display video until the next I-frame is transmitted. In such cases, the client can request a Synchronization Point which forces the device to add an I-frame as soon as possible. Clients can request Synchronization Points for profiles. The device shall add synchronization points for all streams associated with this profile.

Similarly, a synchronization point is used to get an update on full PTZ or event status through the metadata stream.

If a video stream is associated with the profile, an I-frame shall be added to this video stream. If an event stream is associated to the profile, the synchronization point request shall be handled as described in 10.2.3. If the profile is configured for PTZ metadata, the PTZ position shall be repeated within the metadata stream.

A device shall support the request for an I-frame through the SetSynchronizationPoint command if the RTSPStreaming capability is set.

REQUEST:

- **ProfileToken [tt:ReferenceToken]**

Profile reference for which a synchronization point is requested.

RESPONSE:

This message is empty.

FAULTS:

- **env:Sender – ter:InvalidArgVal – ter:NoProfile**

The requested profile token does not exist.

ACCESS CLASS:

ACTUATE

6.9 Video source mode

6.9.1 General

A device may have the ability to change its video source mode. Different video source modes can affect which options are available in GetVideoSourceConfigurationOptions and GetVideoEncoderConfigurationOptions, such as only allowing 16 × 9 aspect ratios in one mode and only allowing 4 × 3 aspect ratios in another, or only allowing 30 fps-derived frame rates in one mode and only allowing 25 fps-derived frame rates in another. The GetVideoSourceModes command provides summary information about the different modes supported by the device.

6.9.2 GetVideoSourceModes

A device returns the information for current video source mode and settable video source modes of specified video source. A device that indicates a capability of VideoSourceMode shall support this command.

REQUEST:

- **VideoSourceToken [tt:ReferenceToken]**

The video source for which the source modes should be retrieved.

RESPONSE:

- **VideoSourceMode - unbounded [tt:VideoSourceMode]**

List of mode information with capabilities of the respective video source.

FAULTS:

- **env:Sender – ter:InvalidArgVal – ter:NoVideoSource**

The requested video source does not exist.

ACCESS CLASS:

READ_SYSTEM**6.9.3 SetVideoSourceMode**

SetVideoSourceMode changes the media profile structure relating to video source for the specified video source mode. A device that indicates a capability of VideoSourceMode shall support this command. The behaviour after changing the mode is not defined in this specification.

REQUEST:

- **VideoSourceToken [tt:ReferenceToken]**

The video source for which the source modes should be retrieved.

- **VideoSourceModeToken [tt:ReferenceToken]**

The token of the video source mode to be set.

RESPONSE:

- **Reboot [xs:boolean]**

Information about rebooting after returning response. When Reboot is set to "true", a device will reboot automatically after setting the mode.

FAULTS:

- **env:Sender – ter:InvalidArgVal – ter:NoVideoSource**

The requested video source does not exist.

- **env:Sender – ter:InvalidArgVal – ter:NoVideoSourceMode**

The requested video source mode does not exist.

ACCESS CLASS:

WRITE_SYSTEM**6.10 OSD (on-screen display)****6.10.1 General**

The OSD service provides functions to enable a client to control and configure the on-screen display of a device. The service introduces the OSD configuration with multiple types (e.g. image, text, date, and time). Also, functions to retrieve and configure the configurations are provided. All OSD configurations are related to a VideoSourceConfiguration, which will display

the OSD content. Figure 5 shows example for on screen display including the normalized screen coordinates.

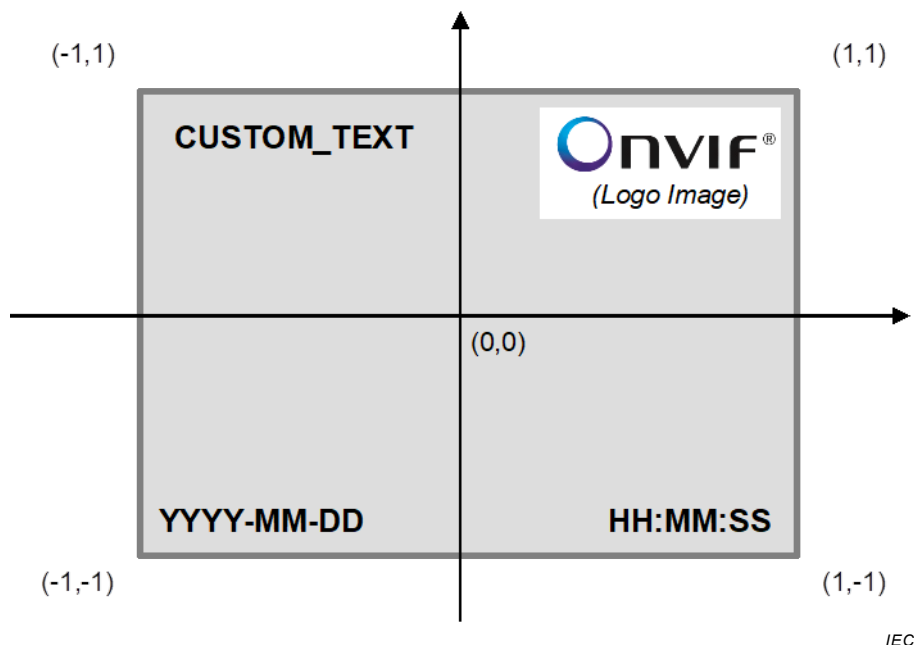


Figure 5 – Example with four OSD configurations

Subclause 6.10 defines methods to create and delete OSD configurations as well as getting, setting and querying the options. A device that signals support for OSD via the OSD capability shall support all OSD methods defined in 6.10.

Devices supporting temporary OSDTextConfiguration shall notify the TemporaryOSDText capability as defined in 6.12. A device shall by default make all OSDTextConfigurations as persistent across reboot, but when IsPersistentText attribute in OSDTextConfiguration is set as "false", OSD text content shall be cleared after reboot. OSDConfiguration shall still be valid after reboot.

6.10.2 CreateOSD

This operation creates a new OSD configuration with specified values and also makes the association between the new OSD and an existing VideoSourceConfiguration identified by its VideoSourceConfigurationToken. Any value required by a device for a new OSD configuration that is optional and not present in the CreateOSD message may be adapted to the appropriate value by the device. A device that indicates OSD capability shall support the creation of OSD as long as the number of existing OSDs does not exceed the value of MaximumNumberOfOSDs in GetOSDOptions. A created OSD configuration shall be deletable.

REQUEST:

- **OSD [tt:OSDConfiguration]**

Contains a new OSD configuration with the specified value. The device is responsible for assigning OSD token. OSD token in CreateOSDRequest can be ignored.

RESPONSE:

- **OSDToken [tt:ReferenceToken]**

Token assigned to the newly created configuration.

FAULTS:

- **env:Receiver – ter:Action – ter:MaxOSDs**
The maximum number of supported OSDs by the specific VideoSourceConfiguration has been reached.
- **env:Sender – ter:InvalidArgVal – ter:NoConfig**
One of the provided configurations indicated by their ConfigurationToken does not exist.
- **env:Receiver – ter:InvalidArgVal – ter:ConfigModify**
The configuration parameters cannot be set.

ACCESS CLASS:

ACTUATE**6.10.3 DeleteOSD**

This operation deletes an OSD configuration.

REQUEST:

- **OSDToken [tt:ReferenceToken]**
Token assigned to the newly created configuration.

RESPONSE:

This is an empty message.

FAULTS:

- **env:Sender – ter:InvalidArgVal – ter:NoConfig**
The requested OSD token OSDToken does not exist.

ACCESS CLASS:

ACTUATE**6.10.4 GetOSDs**

This operation lists existing OSD configurations for the device. The device shall support the listing of existing OSD configurations through the GetOSD command. If neither an OSD token nor a video source configuration token is provided, the device shall respond with all available OSD configurations.

REQUEST:

- **OSDToken – optional [tt:ReferenceToken]**
Token of an existing OSD configuration. If an OSD token is provided the device shall respond with the requested configuration or provide an error if it does not exist.
- **ConfigurationToken – optional [tt:ReferenceToken]**
In the case where only a video source configuration token is provided, the device shall respond with all configurations that exist for the video source configuration.

RESPONSE:

- **OSD – optional, unbounded [tt:OSDConfiguration]**
List of OSD configurations.

FAULTS:

- **env:Sender – ter:InvalidArgVal – ter:NoConfig**

The requested OSD token OSDToken does not exist.

ACCESS CLASS:

READ_MEDIA**6.10.5 SetOSD**

This operation modifies an OSD configuration. Running streams using this configuration may be immediately updated according to the new settings.

A device shall accept any combination of parameters returned by GetOSDOptions. If necessary, the device may adapt parameter values for FontColor, FontSize, and BackgroundColor elements without returning an error. Modifying the configuration token is not supported.

REQUEST:

- **OSD [tt:OSDConfiguration]**

The OSD element contains the modified OSD configuration. The Configuration contains an element that specifies the OSD whose configuration is to be modified. The OSD shall exist in the device.

RESPONSE:

This is an empty message.

FAULTS:

- **env:Sender – ter:InvalidArgVal – ter:NoConfig**

The requested OSD token OSDToken does not exist.

- **env:Receiver – ter:InvalidArgVal – ter:ConfigModify**

The configuration parameters cannot be set.

ACCESS CLASS:

ACTUATE

An OSDTextConfiguration without the IsPersistentText attribute shall be interpreted by the device as persistent.

6.10.6 GetOSDOptions

This operation returns the available options when the OSD parameters are reconfigured. The device shall support the listing of available OSD parameter options (for a given video source configuration) through the GetOSDOptions command. Any combination of the parameters obtained using a given video source configuration shall be a valid input for the corresponding SetOSD command.

REQUEST:

- **ConfigurationToken – optional [tt:ReferenceToken]**

The ConfigurationToken element specifies the video source configuration of which the suitable OSD options are requested. The Video Source Configuration Token shall exist in the device.

RESPONSE:

- **OSDOptions [tt:OSDConfigurationOptions]**

The OSD options which are suitable for the video source configuration specified in the request.

FAULTS:

- **env:Sender – ter:InvalidArgVal – ter:NoConfig**

The requested OSD token OSDToken does not exist.

ACCESS CLASS:

READ_MEDIA**6.11 Privacy masks****6.11.1 General**

Privacy Masks allow regions of the image to be covered. The device signals via the options how many masks and how many edges each mask can support. Additionally, devices may restrict to supporting rectangle masks only. Figure 6 shows how the mask is defined by a polygon using normalized coordinates relative to the VideoSourceConfiguration window.

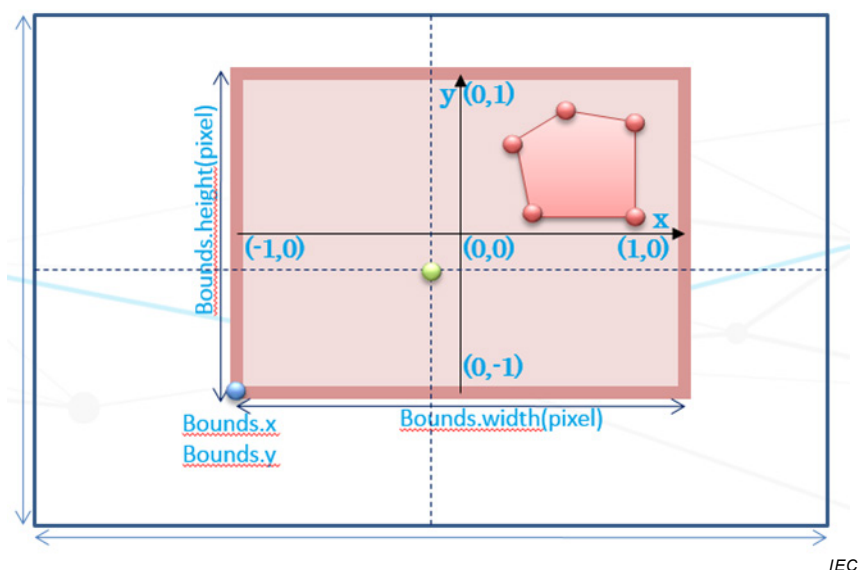


Figure 6 – Example of screen with mask and coordinate system

A device may support automatic updates of the mask location and shape depending on internal operation. Details are outside the scope of this specification.

Depending on the supported options, a device may support the following mask types:

- **Color** The device signals which colour values or ranges are supported. Additionally, a device may signal that it supports a single colour for all masks of a VideoSourceConfiguration.
- **Blurred** The masked area is defocussed.
- **Pixelized** The masked area is covered with a mosaic.

6.11.2 CreateMask

This operation creates a new mask for an existing VideoSourceConfiguration. A device that signals support for masks by the Mask capability shall support the creation of masks via this function as long as the number of existing masks does not exceed the value of MaxMasks for the given VideoSourceConfiguration.

A device shall ignore the mask token passed in the command.

REQUEST:

- **Mask [tr2:Mask]**
Contains the new mask configuration. The device is responsible for assigning the mask token.

RESPONSE:

- **Token [tt:ReferenceToken]**
Token assigned to the newly created mask.

FAULTS:

- **env:Receiver – ter:Action – ter:MaxMasks**
The maximum number of supported masks by the specific VideoSourceConfiguration has been reached.
- **env:Sender – ter:InvalidArgVal – ter:NoConfig**
The provided video source configuration does not exist.
- **env:Receiver – ter:InvalidArgVal – ter:InvalidPolygon**
The provided polygon is not supported.

ACCESS CLASS:

ACTUATE

6.11.3 DeleteMask

This operation deletes a mask configuration.

REQUEST:

- **Token [tt:ReferenceToken]**
Token assigned to the newly created mask.

RESPONSE:

This is an empty message.

FAULTS:

- **env:Sender – ter:InvalidArgVal – ter:NoConfig**
The requested Mask token does not exist.

ACCESS CLASS:

ACTUATE

6.11.4 GetMasks

This operation lists existing mask configurations for the device. A device signalling support for the Mask capability shall support the listing of existing mask configurations through this command. In case neither a mask nor configuration token is provided, the device shall respond with all available mask configurations in the device.

REQUEST:

- **MaskToken – optional [tt:ReferenceToken]**
If a mask token is provided, the device shall respond with the requested configuration or provide an error if it does not exist.
- **ConfigurationToken – optional [tt:ReferenceToken]**
If only a video source configuration token is provided, the device shall respond with all configurations that exist for the video source configuration.

RESPONSE:

- **Mask – optional, unbounded [tr2:Mask]**
List of masks.

FAULTS:

- **env:Sender – ter:InvalidArgVal – ter:NoConfig**
The requested video source configuration or Mask indicated with MaskToken does not exist.

ACCESS CLASS:

READ_MEDIA

6.11.5 SetMask

This operation modifies a mask configuration. Running streams using this configuration may be immediately updated according to the new settings.

A device signalling support for Mask via its capabilities support this command. It shall accept any combination of parameters returned by GetMaskOptions. If necessary, the device may adapt parameter values for the Color and Polygon elements without returning an error.

Note that for devices signalling SingleColorOnly, all masks of the associated VideoSource will be updated.

A device signalling RectangleOnly shall accept any polygon with four points. If the four vertices are not defining an exact rectangle, the device may adjust the vertices.

REQUEST:

- **Mask [tr2:Mask]**
Contains the new mask configuration. The device is responsible for assigning the mask token.

RESPONSE:

This is an empty message.

FAULTS:

- **env:Sender – ter:InvalidArgVal – ter:NoConfig**
The provided video source configuration does not exist.
- **env:Receiver – ter:InvalidArgVal – ter:ConfigModify**
The configuration parameters cannot be set.
- **env:Receiver – ter:InvalidArgVal – ter:InvalidPolygon**
The provided polygon is not supported.

ACCESS CLASS:

ACTUATE**6.11.6 GetMaskOptions**

This operation returns the available options when the Mask parameters are reconfigured. A device signalling support for Mask via its capabilities shall support the listing of available Mask parameter options (for a given video source configuration) via this command. Any combination of the parameters obtained using a given video source configuration shall be a valid input for the corresponding SetMask command.

REQUEST:

- **ConfigurationToken [tt:ReferenceToken]**
Token of an existing configuration for which all existing masks should be listed.

RESPONSE:

- **Options [tr2:MaskOptions]**
This message contains the Mask options which are suitable for the video source configuration specified in the request

FAULTS:

- **env:Sender – ter:InvalidArgVal – ter:NoConfig**
The requested video source configuration does not exist.

ACCESS CLASS:

READ_MEDIA**6.12 Capabilities**

The capabilities reflect optional functions and functionality of a service. The information is static and does not change during device operation.

The following profile capabilities are available:

MaximumNumberOfProfiles:	The sum of fixed and dynamic MediaProfiles supported by the device. A device implementing this service shall provide this capability.
ConfigurationsSupported:	Enumerates the configurations supported by the device as defined by tr2:ConfigurationEnumeration.
SnapshotUri:	Indicates the support for GetSnapshotUri.
Rotation:	Indicates the support for the Rotation feature.
VideoSourceMode:	Indicates the support for changing video source mode.
OSD:	Indicates support for OSD configuration.

TemporaryOSDText:	Indicates the support for temporary OSD text configuration.
Mask:	Indicates support for mask configuration.
SourceMask:	Indicates that privacy masks are only supported at the video source level and not the video source configuration level. If this is true any addition, deletion or change of a privacy mask done for one video source configuration will automatically be applied by the device to a corresponding privacy mask for all other video source configuration associated with the same video source.

The following streaming capabilities are available:

RTSPStreaming:	Indicates the support for live media streaming via RTSP.
RTPMulticast:	Indication of support of UDP multicasting as described in 10.2.1.2.
RTP_RTSP_TCP:	Indicates the support for RTP/RTSP/TCP transport as defined in 10.2.1.4.
NonAggregateControl:	Indicates support for non-aggregate RTSP control as described in 10.2.
RTSPOverWebSocket:	Indicates the support for RTSP/RTP streaming over WebSocket and provides the WebSocket URI, as described in 10.2.1.6.
AutoStartMulticast:	Indicates support for non-RTSP controlled multicast streaming.

REQUEST:

This is an empty message.

RESPONSE:

- **Capabilities [tr2:Capabilities]**
List of capabilities as defined above.

FAULTS:

None

ACCESS CLASS:

PRE_AUTH

6.13 Events

6.13.1 ProfileChange

Whenever a profile is created, deleted or one or more of its configurations are added or removed, the following event shall be generated.

```
Topic: tns1:Media/ProfileChanged
<tt:MessageDescription>
  <tt:Source>
    <tt:SimpleItemDescription Name="Token" Type="tt:ReferenceToken"/>
  </tt:Source>
</tt:MessageDescription>
```

6.13.2 ConfigurationChange

Whenever a configuration of a device changes, the device shall provide the following event. For the parameter Type, pass the appropriate ConfigurationEnumeration value.