

INTERNATIONAL STANDARD

NORME INTERNATIONALE



**Video surveillance systems for use in security applications –
Part 2-31: Live streaming and control based on web services**

**Systèmes de vidéosurveillance destinés à être utilisés dans les applications de
sécurité –**

**Partie 2-31: Transmission en continu en direct et contrôle basé sur les services
web**



THIS PUBLICATION IS COPYRIGHT PROTECTED

Copyright © 2019 IEC, Geneva, Switzerland

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from either IEC or IEC's member National Committee in the country of the requester. If you have any questions about IEC copyright or have an enquiry about obtaining additional rights to this publication, please contact the address below or your local IEC member National Committee for further information.

Droits de reproduction réservés. Sauf indication contraire, aucune partie de cette publication ne peut être reproduite ni utilisée sous quelque forme que ce soit et par aucun procédé, électronique ou mécanique, y compris la photocopie et les microfilms, sans l'accord écrit de l'IEC ou du Comité national de l'IEC du pays du demandeur. Si vous avez des questions sur le copyright de l'IEC ou si vous désirez obtenir des droits supplémentaires sur cette publication, utilisez les coordonnées ci-après ou contactez le Comité national de l'IEC de votre pays de résidence.

IEC Central Office
3, rue de Varembe
CH-1211 Geneva 20
Switzerland

Tel.: +41 22 919 02 11
info@iec.ch
www.iec.ch

About the IEC

The International Electrotechnical Commission (IEC) is the leading global organization that prepares and publishes International Standards for all electrical, electronic and related technologies.

About IEC publications

The technical content of IEC publications is kept under constant review by the IEC. Please make sure that you have the latest edition, a corrigendum or an amendment might have been published.

IEC publications search - webstore.iec.ch/advsearchform

The advanced search enables to find IEC publications by a variety of criteria (reference number, text, technical committee,...). It also gives information on projects, replaced and withdrawn publications.

IEC Just Published - webstore.iec.ch/justpublished

Stay up to date on all new IEC publications. Just Published details all new publications released. Available online and once a month by email.

IEC Customer Service Centre - webstore.iec.ch/csc

If you wish to give us your feedback on this publication or need further assistance, please contact the Customer Service Centre: sales@iec.ch.

Electropedia - www.electropedia.org

The world's leading online dictionary on electrotechnology, containing more than 22 000 terminological entries in English and French, with equivalent terms in 16 additional languages. Also known as the International Electrotechnical Vocabulary (IEV) online.

IEC Glossary - std.iec.ch/glossary

67 000 electrotechnical terminology entries in English and French extracted from the Terms and Definitions clause of IEC publications issued since 2002. Some entries have been collected from earlier publications of IEC TC 37, 77, 86 and CISPR.

A propos de l'IEC

La Commission Electrotechnique Internationale (IEC) est la première organisation mondiale qui élabore et publie des Normes internationales pour tout ce qui a trait à l'électricité, à l'électronique et aux technologies apparentées.

A propos des publications IEC

Le contenu technique des publications IEC est constamment revu. Veuillez vous assurer que vous possédez l'édition la plus récente, un corrigendum ou amendement peut avoir été publié.

Recherche de publications IEC -

webstore.iec.ch/advsearchform

La recherche avancée permet de trouver des publications IEC en utilisant différents critères (numéro de référence, texte, comité d'études,...). Elle donne aussi des informations sur les projets et les publications remplacées ou retirées.

IEC Just Published - webstore.iec.ch/justpublished

Restez informé sur les nouvelles publications IEC. Just Published détaille les nouvelles publications parues. Disponible en ligne et une fois par mois par email.

Service Clients - webstore.iec.ch/csc

Si vous désirez nous donner des commentaires sur cette publication ou si vous avez des questions contactez-nous: sales@iec.ch.

Electropedia - www.electropedia.org

Le premier dictionnaire d'électrotechnologie en ligne au monde, avec plus de 22 000 articles terminologiques en anglais et en français, ainsi que les termes équivalents dans 16 langues additionnelles. Egalement appelé Vocabulaire Electrotechnique International (IEV) en ligne.

Glossaire IEC - std.iec.ch/glossary

67 000 entrées terminologiques électrotechniques, en anglais et en français, extraites des articles Termes et Définitions des publications IEC parues depuis 2002. Plus certaines entrées antérieures extraites des publications des CE 37, 77, 86 et CISPR de l'IEC.

INTERNATIONAL STANDARD

NORME INTERNATIONALE



**Video surveillance systems for use in security applications –
Part 2-31: Live streaming and control based on web services**

**Systèmes de vidéosurveillance destinés à être utilisés dans les applications de
sécurité –
Partie 2-31: Transmission en continu en direct et contrôle basé sur les services
web**

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

COMMISSION
ELECTROTECHNIQUE
INTERNATIONALE

ICS 13.320

ISBN 978-2-8322-7035-6

**Warning! Make sure that you obtained this publication from an authorized distributor.
Attention! Veuillez vous assurer que vous avez obtenu cette publication via un distributeur agréé.**

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.

IECNORM.COM : Click to view the full PDF of IEC 62676-2-31:2019

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).

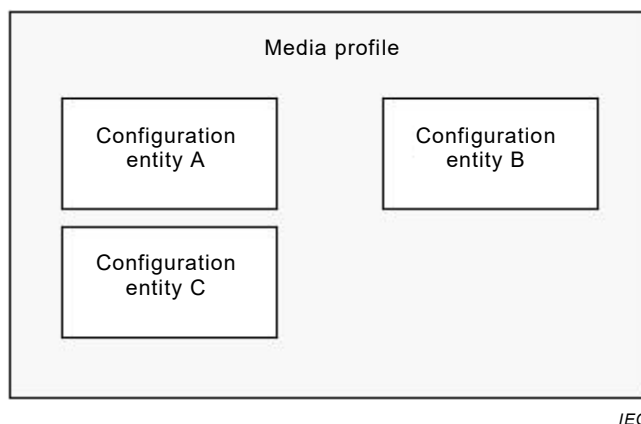


Figure 1 – A media profile

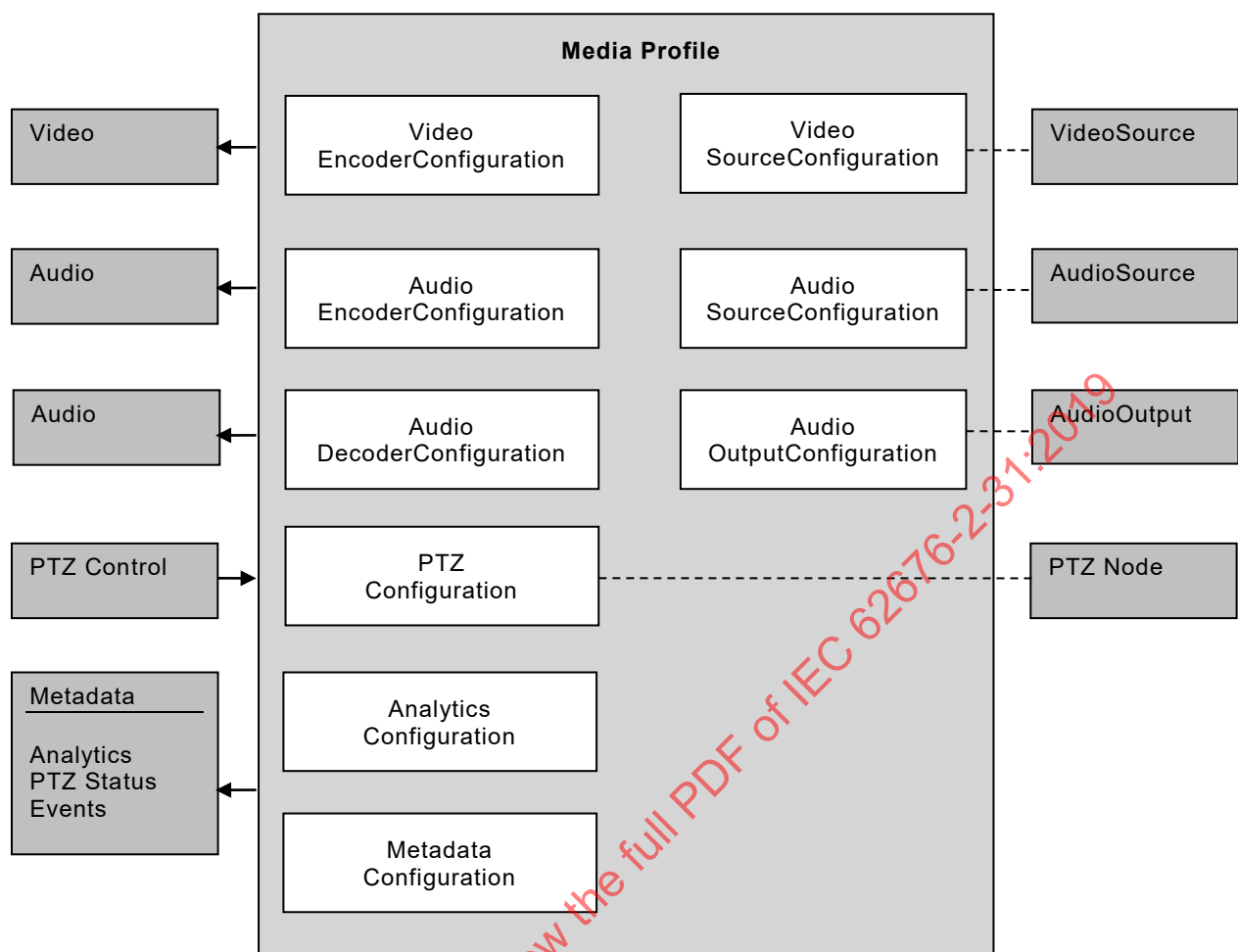
A mediaprofile consists of a set of interconnected configuration entities as depicted in Figure 1. Configurations are provided by the device and can be either static or created dynamically by the device. For example, the dynamic configurations can be created by the device depending on current available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

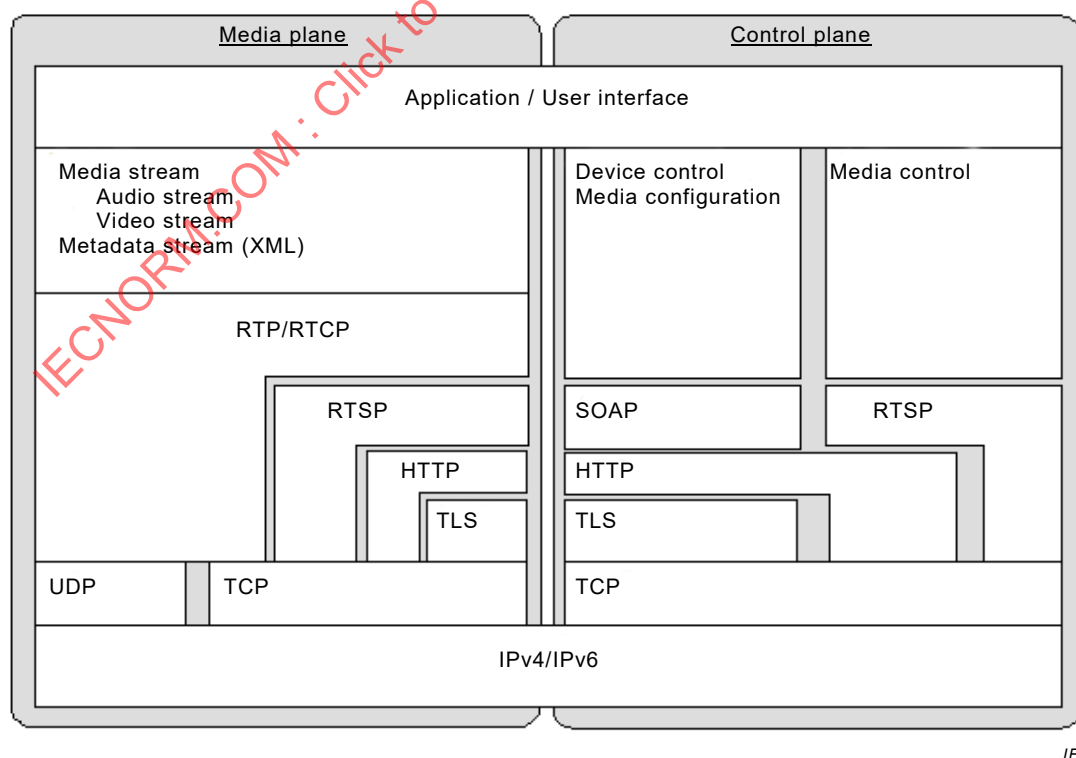
A device may support changing of the video source mode. A change of the video source mode typically requires a reboot of the device owing to hardware dependencies. For example, a device's capability for maximum resolution (1920 × 1080 @ 16:9 or 2048 × 1536 @ 4:3) and frame rate (20 fps or 30 fps) can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

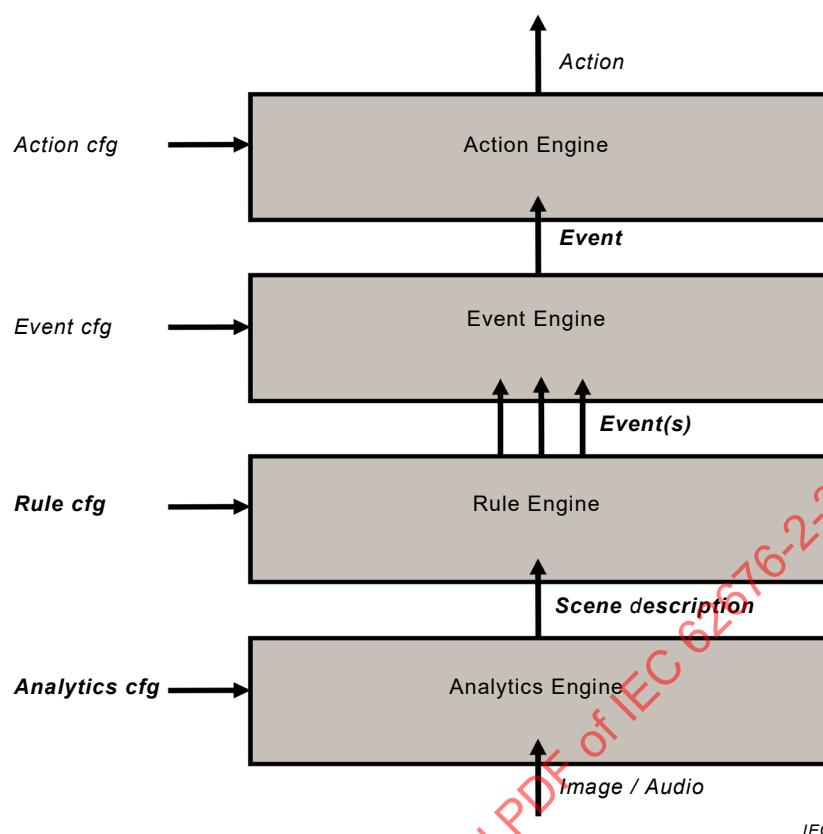


Figure 4 – Analytics architecture

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
tptz	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.

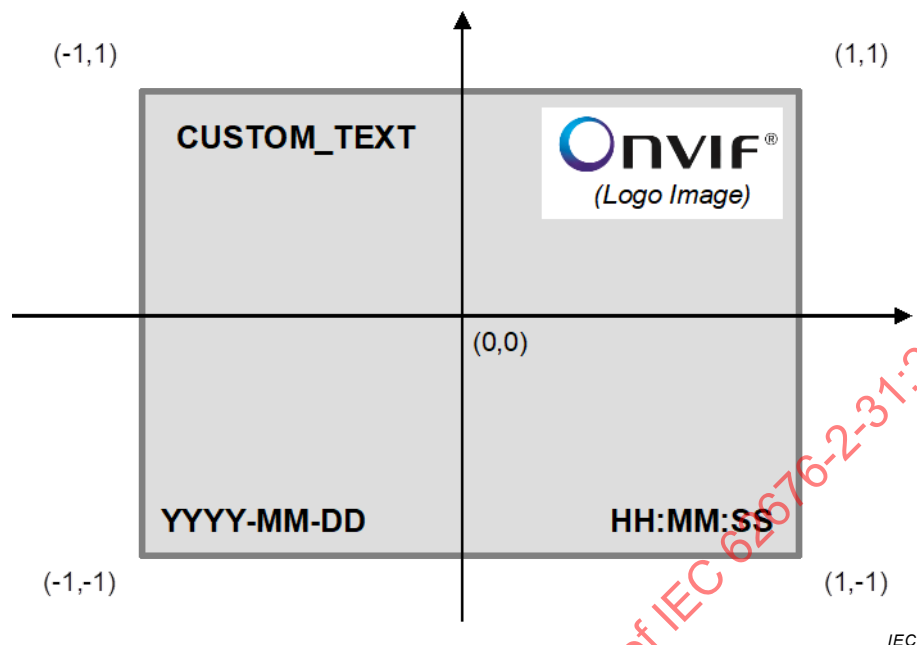


Figure 5 – Example with four OSD configurations

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.

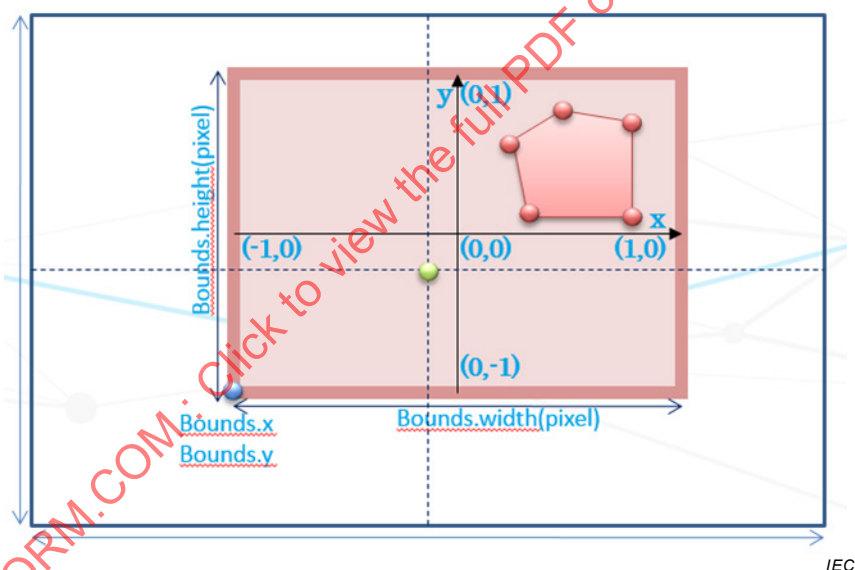


Figure 6 – Example of screen with mask and coordinate system

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.

Topic: tns1:Media/ConfigurationChanged

```
<tt:MessageDescription>
  <tt:Source>
    <tt:SimpleItemDescription Name="Token" Type="tt:ReferenceToken"/>
    <tt:SimpleItemDescription Name="Type" Type="xs:string"/>
  </tt:Source>
</tt:MessageDescription>
```

6.13.3 ActiveConnections

A device that supports the media service should provide the ActiveConnections monitoring event to inform a client about the current usage of its media profiles. An ONVIF compliant device shall use the following topic and message format:

Topic: tns1:Monitoring/Profile/ActiveConnections

```
<xs:complexType name="ProfileStatus">
  <xs:sequence>
    <xs:element name="ActiveConnections" type="tt:ActiveConnection" minOccurs="0"
      maxOccurs="unbounded"/>
    <xs:element name="Extension" type="tt:ProfileStatusExtension" minOccurs="0" />
  </xs:sequence>
</xs:complexType>

<xs:complexType name="ActiveConnection">
  <xs:sequence>
    <xs:element name="CurrentBitrate" type="xs:float"/>
    <xs:element name="CurrentFps" type="xs:float"/>
    <xs:any namespace="##any" processContents="lax" minOccurs="0"
      maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<tt:MessageDescription IsProperty="true">
  <tt:Source>
    <tt:SimpleItemDescription Name="Profile" Type="tt:ReferenceToken"/>
  </tt:Source>
  <tt>Data>
    <tt:ElementItemDescription Name="Status" Type="tt:ProfileStatus"/>
  </tt>Data>
</tt:MessageDescription>
```

6.14 Deviations of media service version 1

6.14.1 General

The previous subclauses define the interface of the media service version 2 which is an extension of the media service 1. This subclause defines the deviations of version 1. For interface details, please refer to Clause F.3.

6.14.2 Profile management

- CreateProfile

The method allows to only create an empty profile, which then has to be populated with a sequence of Add<entity>Configuration calls.

- AddConfiguration

Instead of a single method supporting the addition of multiple configurations, separate methods are defined for adding each of the configurations.

- RemoveConfiguration

Instead of a single method supporting the removal of multiple configurations, separate methods are defined for each of the configurations.

- GetProfiles

The method with the same name allows the retrieval of all media profiles of the device. A second method called `GetProfile` allows a dedicated profile to be retrieved.

6.14.3 Configuration listing

- `Get<entity>Configuration`

For each of these methods, three methods are implemented. These are

- `Get<entity>Configurations`
Retrieve all configurations of type entity.
- `Get<entity>Configuration`
Retrieve one configuration specified by its reference token.
- `GetCompatible<entity>Configurations`
Retrieve all configurations that are compatible with the current profile definition.

6.14.4 Privacy masks

The related set of operations is not supported.

7 Imaging service

7.1 General

The imaging service provides operations used to control and configure imaging properties on a device. A device that has one or more video sources should support the imaging service as defined in Clause F.2. The imaging settings are part of the `VideoSource` entity. This means that imaging parameters directly affect a specific video source.

7.2 Imaging settings

7.2.1 Parameters

The imaging service provides operations to get or set imaging parameters and the valid ranges for those parameters. Some parameters have no effect if a specific mode is not set. Some of the parameters included in the settings require a specific imaging capability that can be requested through the `GetOptions` command. The following settings are available through the imaging service operations:

BacklightCompensation: Enables/disables BLC mode (on/off)

- On
 - Optional level parameter (unspecified unit).
- Off

Brightness: Adjusts the image brightness (unspecified unit).

ColorSaturation: Adjusts the colour saturation in the image (unspecified unit).

Sharpness: Adjusts the sharpness in the image (unspecified unit).

Contrast: Adjusts the image contrast (unspecified unit).

Exposure:

- Auto – Enables the exposure algorithm on the device:
 - Priority – Sets the exposure priority mode (low noise/framerate).

- Window – Rectangular exposure mask.
- Min/MaxExposureTime – Exposure time range allowed to be used by the algorithm.
- Min/MaxGain – The sensor gain range that is allowed to be used by the algorithm.
- Min/MaxIris – The iris range allowed to be used by the algorithm.
- Manual – Disables the exposure algorithm on the device:
 - ExposureTime – The fixed exposure time used by the image sensor (μ s).
 - Gain – The fixed gain used by the image sensor (dB).
 - Iris – The fixed attenuation of input light affected by the iris (dB). 0dB maps to a fully opened iris.

Focus:

- Auto (parameters that apply to automatic mode only):
 - Near/FarLimit – Limits for focus lens (m).
- Manual (parameters that apply to manual mode only):
 - Default speed – The default speed for focus move operation (when the speed parameter not is present). Manual control is done through the move command, see 7.4.

IrCutFilter: Toggles the IR cut filter state between on, off and auto. The auto state lets the exposure algorithm handle when the IR cut filter should be turned on or off.

- On: enable IR cut filter. Typically day mode.
- Off: disable IR cut filter. Typically night mode.
- Auto: IR cut filter is automatically activated by the device.

IrCutFilterAutoAdjustment: Optional parameters applied to only auto mode to adjust timing of toggling IR cut filter

- BoundaryType – Specifies which boundaries to automatically toggle IR cut filter that the following parameters are applied to:
 - Common: Applied to both boundaries automatically toggling IR cut filter on and off
 - ToOn/ToOff: Applied individually to one of the boundaries automatically toggling IR cut filter to on/off
- BoundaryOffset – Adjusts boundary exposure level for toggling IR cut filter to on/off specified with unitless normalized value from +1,0 to -1,0. Zero is default and -1,0 is the darkest adjustment (unitless).
- ResponseTime – Delay time of toggling IR cut filter to on/off after crossing of the boundary exposure levels.

Whitebalance:

- Auto whitebalancing mode (auto/manual).
- Manual (parameters that apply to manual mode only):
 - Rgain (unitless).
 - Bgain (unitless).

WideDynamicRange: Wide dynamic range (on/off):

- On
 - Optional level parameter (unitless).
- Off

Image stabilization: Enables/disables image stabilization feature (on/off/auto):

- On: Enable Image Stabilization feature.
 - Optional level parameter (unspecified unit).
- Off: Disable Image Stabilization feature
- Auto: Image Stabilization feature is automatically activated by the device

ToneCompensation: Enables/disables Tone Compensation mode.

- On: Enable Tone Compensation feature.
 - Optional level parameter - Optional level parameter to adjust level of compensation specified with normalized value from 0,0 to +1,0 (Unitless).
- Off: Disable Tone Compensation feature.
- Auto: Enable Tone Compensation feature with adjusting level automatically.

Defogging: Enables/disables Defogging function.

- On: Enable Defogging feature.
 - Optional level parameter - Optional level parameter to adjust level of compensation specified with normalized value from 0,0 to +1,0 (unitless).
- Off: Disable Defogging feature.
- Auto: Enable Defogging feature with adjusting level automatically.

Noise Reduction: Adjusts the noise reduction level in the image with normalized value from 0,0 to +1,0 (Unitless). Level = 0 means no noise reduction or minimal noise reduction.

7.2.2 GetImagingSettings

This operation requests the imaging setting for a video source on the device. A device implementing the imaging service shall support this command.

REQUEST:

- **VideoSourceToken [tt:ReferenceToken]**
Reference to the video source for which the imaging settings are requested

RESPONSE:

- **ImagingSettings [tt:ImagingSettings20]**
Imaging settings for the video source as defined in 7.2.1.

FAULTS:

- **env:Sender – ter:InvalidArgVal – terNoSource**
The requested video source does not exist.
- **env:Receiver – ter:ActionNotSupported – ter:NoImagingForSource**
The requested video source does not support imaging settings.

ACCESS CLASS:

READ_MEDIA

If the video source supports any of the imaging settings as defined by the ImagingSettings type in the ONVIF Schema, then it should be possible to retrieve the imaging settings from the device through the GetImagingSettings command.

7.2.3 SetImagingSettings

This operation sets the imaging settings for a video source on a device. A device implementing the imaging service shall support this command.

The possible configurable parameters of the imaging settings are described in 7.2.1. Settings options are obtained through the command defined in 7.2.4

REQUEST:

- **VideoSourceToken [tt:ReferenceToken]**
Reference to the VideoSource.
- **ImagingSettings [tt:ImagingSettings20]**
The imaging settings to be applied.
- **ForcePersistence – optional [xs:boolean]**
Deprecated indication of whether the configuration persists after reboot.

RESPONSE:

This message is empty

FAULTS:

- **env:Sender – ter:InvalidArgVal – terNoSource**
The requested VideoSource does not exist.
- **env:Receiver – ter:ActionNotSupported – terNoImagingForSource**
The requested VideoSource does not support imaging settings.
- **env:Sender – ter:InvalidArgVal – terSettingsInvalid**
The requested settings are incorrect.

ACCESS CLASS:

ACTUATE

7.2.4 GetOptions

This operation gets the valid ranges for the imaging parameters that have device specific ranges. A device implementing the imaging service shall support this command. The command shall return all supported parameters and their ranges such that these can be applied to the SetImagingSettings command.

For read-only parameters that cannot be modified via the SetImagingSettings command, only a single option or identical Min and Max values shall be provided.

REQUEST:

- **VideoSourceToken [tt:ReferenceToken]**
Reference to the VideoSource

RESPONSE:

- **ImagingOptions [tt:ImagingOptions20]**
Valid ranges for the imaging parameters that are categorized as device-specific.

FAULTS:

- **env:Sender – ter:InvalidArgVal – terNoSource**
The requested VideoSource does not exist.
- **env:Receiver – ter:ActionNotSupported- terNoImagingForSource**
The requested VideoSource does not support imaging settings.

ACCESS CLASS:

READ_MEDIA**7.3 Imaging Presets****7.3.1 General**

Video source manufacturers can offer preset imaging configurations, which contain different parameter settings, tuned for optimum imaging performance under given scene circumstances. The Imaging Presets functionality is similar in function to what compact photography cameras offer today to the non-professional user, in order to easily be able to obtain optimum pictures under given scene conditions (e.g. Portrait, Landscape, Sports, Macro, Silhouette).

Three commands enable the use of Imaging Presets.

7.3.2 GetPresets

The GetPresets command requests the current predefined list of Imaging Settings (Presets) offered by the manufacturer for a given video source.

The output is a list containing the available Imaging Presets. If the device offers no support for Imaging Presets, GetPresetsResponse shall return an empty list.

REQUEST:

- **VideoSourceToken**
Reference token to VideoSource for which the available Imaging Presets should be requested.

RESPONSE:

- **Preset – unbounded [timg:ImagingPreset]**
The available list of imaging presets supported by the device. Any device with Imaging Presets capability shall offer at least one Imaging Preset. If the device offers no support for Imaging Presets, GetPresetsResponse shall return an empty list.

FAULTS:

- **env:Sender – ter:InvalidArgVal – terNoSource**
The requested VideoSource does not exist.

ACCESS CLASS:

READ_MEDIA**7.3.3 GetCurrentPreset**

The GetCurrentPreset command shall request the Imaging Preset that is currently applied to the specified video source, i.e. it shall request which of the predefined set of Imaging Settings (and/or Thermal Settings) was last applied to the video source. The output is the current Imaging Preset.

If Imaging Presets are not supported, or if the video source configuration does not match any of the existing Imaging Presets, the output of GetCurrentPreset shall be Empty.

REQUEST:

- **VideoSourceToken [tt:ReferenceToken]**

Reference token to VideoSource for which the available Imaging Presets should be requested.

RESPONSE:

- **CurrentPreset – unbounded [timg:ImagingPreset]**

The current (i.e. last applied) Imaging Preset applied to the specified Video Source. Return is empty if no capability or if current configuration does not match any existing Imaging Preset.

FAULTS:

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

The requested VideoSource does not exist.

ACCESS CLASS:

READ_MEDIA

7.3.4 SetCurrentPreset

The SetCurrentPreset command shall request a given Imaging Preset to be applied to the specified video source. SetCurrentPreset shall only be available for Video Sources with Imaging Presets capability.

Imaging Presets are defined by the manufacturer and offered as a tool to simplify Imaging Settings adjustments for specific scene patterns. When the new Imaging Preset is applied by SetCurrentPreset, as a response, the device shall adjust the video source settings to match those values defined by the specified Imaging Preset.

REQUEST:

- **VideoSourceToken [tt:ReferenceToken]**

Reference token to VideoSource for which the Imaging Presets should be set.

- **PresetToken [tt:ReferenceToken]**

Preset to be applied.

RESPONSE:

This message is empty

FAULTS:

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

The requested VideoSource does not exist.

ACCESS CLASS:

ACTUATE

7.4 Focus operations

7.4.1 Move

The Move command moves the focus lens in an absolute, a relative or in a continuous manner from its current position. The speed argument is optional for absolute and relative control, but required for continuous. If no speed argument is used, the default speed is used. Focus adjustments through this operation will turn off the autofocus. A device with support for remote focus control should support absolute, relative or continuous control through the Move operation. The supported MoveOptions are signalled via the GetMoveOptions command.

At least one focus control capability is required for this operation to be functional.

The move operation contains the following commands:

Absolute – Requires position parameter and optionally takes a speed argument. A unitless type is used by default for focus positioning and speed. Optionally, if supported, the position may be requested in m^{-1} units.

Relative – Requires distance parameter and optionally takes a speed argument. Negative distance means negative direction.

Continuous – Requires a speed argument. Negative speed argument means negative direction.

REQUEST:

- **VideoSourceToken [tt:ReferenceToken]**
Reference token to VideoSource.
- **Focus [tt:FocusMove]**
Focal operation.

RESPONSE:

This message is empty

FAULTS:

- **env:Sender – ter:InvalidArgVal – terNoSource**
The requested VideoSource does not exist.
- **env:Receiver – ter:ActionNotSupported- terNoImagingForSource**
The requested VideoSource does not support imaging settings.

ACCESS CLASS:

ACTUATE

7.4.2 GetMoveOptions

The GetMoveOptions command retrieves the focus lens move options to be used in the Move command as defined in 7.4. A device that supports the imaging service shall support the GetMoveOptions command. The response to the command shall include all supported Move Operations. If focus move is not supported at all, the response shall be empty.

REQUEST:

- **VideoSourceToken [tt:ReferenceToken]**
Reference token to VideoSource.

RESPONSE:

- **MoveOptions [tt:MoveOptions20]**

The valid ranges for the focus lens move command.

FAULTS:

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

The requested VideoSource does not exist.

- **env:Receiver – ter:ActionNotSupported- terNoImagingForSource**

The requested VideoSource does not support imaging settings.

ACCESS CLASS:

READ_MEDIA

7.4.3 Stop

The Stop command stops all ongoing focus movements of the lens. A device with support for remote focus control as signalled via the GetMoveOptions should support this command. The operation will not affect ongoing autofocus operation.

REQUEST:

- **VideoSourceToken [tt:ReferenceToken]**

Reference token to VideoSource.

RESPONSE:

This message is empty

FAULTS:

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

The requested VideoSource does not exist.

- **env:Receiver – ter:ActionNotSupported- terNoImagingForSource**

The requested VideoSource does not support imaging settings.

ACCESS CLASS:

ACTUATE

7.4.4 GetImagingStatus

The GetStatus command requested the current imaging status from the device. A device with support for remote focus control as signalled via the GetMoveOptions shall support this command.

The imaging status contains:

- Focus position, move status and error information.
 - The focus position is represented in a unitless type.
 - Move status may be in a MOVING, IDLE or UNKNOWN state.
 - Error information provided as a string, for example a positioning error indicated by the hardware.

REQUEST:

- **VideoSourceToken [tt:ReferenceToken]**
Reference token to VideoSource.

RESPONSE:

- **ImagingStatus [tt:ImagingStatus20]**
The imaging status information.

FAULTS:

- **env:Sender – ter:InvalidArgVal – terNoSource**
The requested VideoSource does not exist.
- **env:Receiver – ter:ActionNotSupported- terNoImagingForSource**
The requested VideoSource does not support imaging settings.

ACCESS CLASS:

READ_MEDIA

7.5 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:

ImageStabilization: Indicates whether or not ImageStabilization feature is supported.

ImagingPresets: Indicates whether or not ImagingPresets feature is supported.

REQUEST:

This message is empty

RESPONSE:

- **Capabilities [timg:Capabilities]**
Set of indicators for function groups as described above.

FAULTS:

None

ACCESS CLASS:

PRE_AUTH

8 PTZ service

8.1 General

The PTZ service provides interfaces for the configuration and operation of movable cameras, also called domes. For the schema definition see Clause F.5.

8.2 PTZ node

8.2.1 General

A PTZ-capable device can have multiple PTZ nodes. The PTZ nodes can represent mechanical PTZ drivers, uploaded PTZ drivers or digital PTZ drivers. PTZ nodes are the lowest level entities in the PTZ control API and reflect the supported PTZ capabilities. The PTZ node is referenced either by its name or by its reference token. The PTZ service does not provide operations to create or manipulate PTZ nodes.

The following properties shall be provided for all PTZ nodes:

- Token – A unique identifier used to reference PTZ nodes.
- Name – A name given by the installer.
- SupportedPTZSpaces – A list of coordinate systems available for the PTZ node. For each Coordinate System, the PTZ node shall specify its allowed range.
- MaximumNumberOfPresets – All preset operations shall be available for this PTZ node if one preset is supported.
- HomeSupported – A Boolean operator specifying the availability of a home position. If set to "true", Home Position operations shall be available for this PTZ node.
- AuxiliaryCommands – A list of supported Auxiliary commands. If the list is not empty, the Auxiliary Operations shall be available for this PTZ node. A device may use auxiliary commands that are described in IEC 60839-11-31.
- MaximumNumberOfPresetTours – Indicates number of preset tours that can be created. Required preset tour operations shall be available for this PTZ node if one or more preset tours are supported.

8.2.2 GetNodes

A PTZ-capable device shall implement this operation and return all PTZ nodes available on the device.

REQUEST:

This is an empty message.

RESPONSE:

- **PTZNode – optional, unbounded [tt:PTZNode]**
List of the existing PTZ nodes on the device.

FAULTS:

None

ACCESS CLASS:

READ_MEDIA

8.2.3 GetNode

A PTZ-capable device shall implement the GetNode operation and return the properties of the requested PTZ node, if it exists. Otherwise, the device shall respond with an appropriate fault message.

REQUEST:

- **NodeToken [tt:ReferenceToken]**
Reference to the requested PTZ node.

RESPONSE:

- **PTZNode [tt:PTZNode]**
The requested PTZ node properties.

FAULTS:

- **env:Sender – ter:InvalidArgVal – ter:NoEntity**
No such PTZ node on the device.

ACCESS CLASS:

READ_MEDIA

8.3 PTZ configuration

8.3.1 General

The PTZConfiguration contains a reference to the PTZ node in which it belongs. This reference cannot be changed by a client.

The following elements are part of the PTZ Configuration:

- **PTZNodeToken** – A mandatory reference to the PTZ node that the PTZ configuration belongs to.
- **DefaultAbsolutePanTiltPositionSpace** – If the PTZ node supports absolute pan/tilt movements, it shall specify one absolute pan/tilt position space as default.
- **DefaultRelativePanTiltTranslationSpace** – If the PTZ node supports relative pan/tilt movements, it shall specify one relative pan/tilt translation space as default.
- **DefaultContinuousPanTiltVelocitySpace** – If the PTZ node supports continuous pan/tilt movements, it shall specify one continuous pan/tilt velocity space as default.
- **DefaultPanTiltSpeedSpace** – If the PTZ node supports absolute or relative movements, it shall specify one pan/tilt speed space as default.
- **DefaultAbsoluteZoomPositionSpace** – If the PTZ node supports absolute zoom movements, it shall specify one absolute zoom position space as default.
- **DefaultRelativeZoomTranslationSpace** – If the PTZ node supports relative zoom movements, it shall specify one relative zoom translation space as default.
- **DefaultContinuousZoomVelocitySpace** – If the PTZ node supports continuous zoom movements, it shall specify one continuous zoom velocity space as default.
- **DefaultPTZSpeed** – If the PTZ node supports absolute or relative PTZ movements, it shall specify corresponding default pan/tilt and zoom speeds.
- **DefaultPTZTimeout** – If the PTZ node supports continuous movements, it shall specify a default timeout, after which the movement stops.
- **PanTiltLimits** – The pan/tilt limits element should be present for a PTZ node that supports an absolute pan/tilt. If the element is present, it signals the support for configurable pan/tilt limits. If limits are enabled, the pan/tilt movements shall always stay within the specified range. The pan/tilt limits are disabled by setting the limits to -INF or +INF.
- **ZoomLimits** – The zoom limits element should be present for a PTZ node that supports absolute zoom. If the element is present, it signals the supports for configurable zoom limits. If limits are enabled, the zoom movements shall always stay within the specified range. The Zoom limits are disabled by settings the limits to -INF and +INF.
- **MoveRamp** – The optional acceleration ramp used by the device when moving.
- **PresetRamp** – The optional acceleration ramp used by the device when recalling presets.
- **PresetTourRamp** – The optional acceleration ramp used by the device when executing PresetTours.

The default position/translation/velocity spaces are introduced to allow clients sending move requests without the need to specify a certain coordinate system. The default speeds are introduced to control the speed of move requests (absolute, relative, preset), where no explicit speed has been set.

The allowed pan and tilt range for pan/tilt limits are defined by a two-dimensional space range that is mapped to a specific absolute pan/tilt position space. At least one pan/tilt position space is required by PTZNode to support pan/tilt limits. The limits apply to all supported absolute, relative and continuous pan/tilt movements. The limits shall be checked within the coordinate system for which the limits have been specified. What this means in practice is that even if movements are specified in a different coordinate system, the requested movements shall be transformed to the coordinate system of the limits where the limits can be checked. When a relative or continuous movements is specified, which would leave the specified limits, the PTZ unit has to move along the specified limits. The zoom limits have to be interpreted accordingly.

8.3.2 GetConfigurations

A PTZ-capable device shall return all available PTZConfigurations through the GetConfigurations operation.

REQUEST:

This is an empty message.

RESPONSE:

- **PTZConfiguration – optional, unbounded [tt:PTZConfiguration]**
List of all existing PTZConfigurations on the device.

FAULTS:

None

ACCESS CLASS:

READ_MEDIA

8.3.3 GetConfiguration

A PTZ-capable device shall return the requested PTZ configuration, if it exists, through the GetConfiguration operation.

REQUEST:

- **ConfigurationToken [tt:ReferenceToken]**
Reference to the requested PTZ configuration.

RESPONSE:

- **PTZConfiguration[tt:PTZConfiguration]**
The requested PTZ node configuration.

FAULTS:

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

ACCESS CLASS:

READ_MEDIA

8.3.4 GetConfigurationOptions

A PTZ-capable device shall implement the GetConfigurationOptions operation. It returns the list of supported coordinate systems including their range limitations. Therefore, the options can differ depending on whether the PTZ configuration is assigned to a profile (see 6.3.6) containing VideoSourceConfiguration. In this case, the options may contain additional coordinate systems referring to the image coordinate system described by the VideoSourceConfiguration. Each listed coordinate system belongs to one of the groups listed in 8.3.1. If the PTZ node supports continuous movements, it shall return a timeout range within which timeouts are accepted by the PTZ node.

REQUEST:

- **ConfigurationToken [tt:ReferenceToken]**
Reference to the requested PTZ configuration.

RESPONSE:

- **PTZConfigurationOptions[tt:PTZConfigurationOptions]**
The requested PTZ node configuration options.

FAULTS:

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

ACCESS CLASS:

READ_MEDIA

8.3.5 SetConfiguration

A PTZ-capable device shall implement the SetConfiguration operation. The ForcePersistence flag indicates if the changes remain after reboot of the device.

REQUEST:

- **ConfigurationToken [tt:ReferenceToken]**
Reference to the PTZ configuration to be modified.
- **PTZConfigurationOptions[tt:PTZConfigurationOptions]**
The requested PTZ node configuration options.
- **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:NoConfig**
The requested configuration does not exist.
- **env:Sender – ter:InvalidArgVal – ter:ConfigModify**
The configuration parameters cannot be set.
- **env:Receiver – ter:Action – ter:ConfigurationConflict**
The new settings conflict with other uses of the configuration.

ACCESS CLASS:

ACTUATE

8.3.6 GetCompatibleConfigurations

A device signalling support for GetCompatibleConfigurations via the capability GetCompatibleConfigurations shall return all available PTZConfigurations that can be added to the referenced media profile through the GetCompatibleConfigurations operation.

A device providing more than one PTZConfiguration or more than one VideoSourceConfiguration or which has any other resource interdependency between PTZConfiguration entities and other resources listable in a media profile should implement this operation. The PTZConfiguration entities returned by this operation shall not fail on adding them to the referenced media profile.

REQUEST:

- **ProfileToken [tt:ReferenceToken]**
Token of an existing media profile.

RESPONSE:

- **PTZConfiguration – optional, unbounded [tt:PTZConfiguration]**
List of compatible PTZ node configurations.

FAULTS:

- **env:Receiver – ter:ActionNotSupported – ter:OptionalActionNotSupported**
This method is not implemented by the device.

ACCESS CLASS:

READ_MEDIA

8.4 Move operations

8.4.1 General

This subclause describes three operations to move the PTZ unit absolutely, relatively or continuously. All operations require a ProfileToken referencing a Media Profile that includes a PTZConfiguration.

All move commands shall be implemented non-blocking, meaning they shall not wait until the requested move operation has finished. The last move operation can be overwritten by sending another move request.

Owing to the wide range of physical devices this specification covers, a specific response time to PTZ move operations is not required. However, a device shall strive for minimal latency regarding PTZ move command request-to-response time. Note that the device controls command delay and video encoding delay; network delay and client delay also affect the user experience but cannot be controlled by the device.

8.4.2 AbsoluteMove

If a PTZ node supports absolute pan/tilt or absolute zoom movements, it shall support the AbsoluteMove operation. The position argument of this command specifies the absolute position to which the PTZ unit moves. It splits into an optional pan/tilt element and an optional zoom element. If the pan/tilt position is omitted, the current pan/tilt movement shall not be affected by this command. The same holds for the zoom position.

The spaces referenced within the position shall be absolute position spaces supported by the PTZ node. If the space information is omitted, the corresponding default spaces of the PTZ configuration, a part of the specified media profile, is used. A device may support absolute pan/tilt movements, absolute zoom movements or no absolute movements by providing only absolute position spaces for the supported cases.

An existing Speed argument overrides DefaultSpeed of the corresponding PTZ configuration during movement to the requested position. If spaces are referenced within the Speed argument, they shall be speed spaces supported by the PTZ node.

The operation shall fail if the requested absolute position is not reachable.

REQUEST:

- **ProfileToken [tt:ReferenceToken]**
Reference to an existing media profile.
- **Position [tt:PTZVector]**
Vector specifying the absolute target position.
- **Speed – optional [tt:PTZSpeed]**
Optional speed vector.

RESPONSE:

This is an empty message.

FAULTS:

- **env:Sender – ter:InvalidArgVal – ter:NoProfile**
The requested profile token ProfileToken does not exist.
- **env:Sender – ter:InvalidArgVal – ter:NoPTZProfile**
The requested profile token does not reference a PTZ configuration.
- **env:Sender – ter:InvalidArgVal – ter:SpaceNotSupported**
A space is referenced that is not supported by the PTZ node.
- **env:Sender – ter:InvalidArgVal – ter:InvalidPosition**
The requested position is out of bounds.
- **env:Sender – ter:InvalidArgVal – ter:InvalidSpeed**
The requested speed is out of bounds.

ACCESS CLASS:

ACTUATE

8.4.3 RelativeMove

If a PTZ node supports relative pan/tilt or relative zoom movements, then it shall support the RelativeMove operation. The translation argument of this operation specifies the difference from the current position to the position to which the PTZ device is instructed to move. The operation is split into an optional pan/tilt element and an optional zoom element. If the pan/tilt element is omitted, the current pan/tilt movement shall not be affected by this command. The same holds for the zoom element.

The spaces referenced within the translation element shall be translation spaces supported by the PTZ node. If the space information is omitted for the translation argument, the corresponding default spaces of the PTZ configuration, which is part of the specified media

profile, is used. A device may support relative pan/tilt movements, relative zoom movements or no relative movements by providing only translation spaces for the supported cases.

An existing speed argument overrides DefaultSpeed of the corresponding PTZ configuration during movement by the requested translation. If spaces are referenced within the speed argument, they shall be speed spaces supported by the PTZ node.

The command can be used to stop the PTZ unit at its current position by sending zero values for pan/tilt and zoom. Stopping shall have the very same effect independent of the relative space referenced.

If the requested translation leads to an absolute position which cannot be reached, the PTZ node shall move to a reachable position along the border of valid positions.

REQUEST:

- **ProfileToken [tt:ReferenceToken]**
Reference to an existing media profile.
- **Translation [tt:PTZVector]**
Vector specifying the positional translation relative to the current position.
- **Speed – optional [tt:PTZSpeed]**
Optional speed vector.

RESPONSE:

This is an empty message.

FAULTS:

- **env:Sender – ter:InvalidArgVal – ter:NoProfile**
The requested profile token ProfileToken does not exist.
- **env:Sender – ter:InvalidArgVal – ter:NoPTZProfile**
The requested profile token does not reference a PTZ configuration.
- **env:Sender – ter:InvalidArgVal – ter:SpaceNotSupported**
A space is referenced which is not supported by the PTZ node.
- **env:Sender – ter:InvalidArgVal – ter:InvalidTranslation**
The requested translation is out of bounds.
- **env:Sender – ter:InvalidArgVal – ter:InvalidSpeed**
The requested speed is out of bounds.

ACCESS CLASS:

ACTUATE

8.4.4 ContinuousMove

A PTZ-capable device shall support continuous movements. The velocity argument of this command specifies a signed speed value for the pan, tilt and zoom. The combined Pan/Tilt element is optional and the Zoom element itself is optional. If the Pan/Tilt element is omitted, the current pan/tilt movement shall not be affected by this command. The same holds for the zoom element. The spaces referenced within the velocity element shall be velocity spaces supported by the PTZ node. If the space information is omitted for the velocity argument, the corresponding default spaces of the PTZ configuration belonging to the specified media profile is used. A device may support continuous pan/tilt movements and/or continuous zoom movements by providing only velocity spaces for the supported cases.

An existing timeout argument overrides the DefaultPTZTimeout parameter of the corresponding PTZ configuration for this move operation. The timeout parameter specifies how long the PTZ node continues to move.

A device shall stop movement in a particular axis (pan, tilt, or zoom) when zero is sent as the ContinuousMove parameter for that axis. Stopping shall have the same effect independent of the velocity space referenced. This command has the same effect on a continuous move as the stop command specified in 8.4.6.

If the requested velocity leads to absolute positions which cannot be reached, the PTZ node shall move to a reachable position along the border of its range. A typical application of the continuous move operation is controlling the PTZ via a joystick.

REQUEST:

- **ProfileToken [tt:ReferenceToken]**
Reference to an existing media profile.
- **Velocity [tt:PTZSpeed]**
Speed vector specifying the velocity of pan, tilt and zoom.
- **Timeout – optional [xs:duration]**
Optional timeout.

RESPONSE:

This is an empty message.

FAULTS:

- **env:Sender – ter:InvalidArgVal – ter:NoProfile**
The requested profile token ProfileToken does not exist.
- **env:Sender – ter:InvalidArgVal – ter:NoPTZProfile**
The requested profile token does not reference a PTZ configuration.
- **env:Sender – ter:InvalidArgVal – ter:SpaceNotSupported**
A space is referenced which is not supported by the PTZ node.
- **env:Sender – ter:InvalidArgVal – ter:InvalidTranslation**
The requested translation is out of bounds.
- **env:Sender – ter:InvalidArgVal – ter:TimeoutNotSupported**
The specified timeout argument is not within the supported timeout range.
- **env:Sender – ter:InvalidArgVal – ter:InvalidVelocity**
The requested speed is out of bounds.

ACCESS CLASS:

ACTUATE

8.4.5 GeoMove

A device signalling GeoMove in one of its PTZ nodes shall support this command.

The optional AreaHeight and AreaWidth parameters can be added to the request, so that the PTZ-capable device can internally determine the zoom factor. In the case where both AreaHeight and AreaWidth are not provided, the unit will not change the zoom. AreaHeight and AreaWidth are expressed in metres.

An existing speed argument overrides the DefaultSpeed of the corresponding PTZ configuration during movement by the requested translation. If spaces are referenced within the speed argument, they shall be speed spaces supported by the PTZ node.

If the PTZ-capable device does not support automatic retrieval of the geolocation, it shall be configured by using SetGeoLocation before it can perform geo-referenced commands. If the client requests a GeoMove command before the geolocation of the device is configured, the device shall return an error.

Depending on the kinematics of the PTZ-capable device, the requested position may not be reachable. In this situation, the device shall return an error, signalling that it cannot perform the requested action owing to physical limitations.

REQUEST:

- **ProfileToken [tt:ReferenceToken]**
Reference to an existing media profile.
- **Target [tt:GeoLocation]**
Target coordinates.
- **Speed – optional [tt:PTZSpeed]**
Speed vector specifying the velocity of pan, tilt and zoom.
- **AreaWidth – optional [xs:float]**
Optional area to be shown.
- **AreaHeight – optional [xs:float]**
Optional area to be shown.

RESPONSE:

This is an empty message.

FAULTS:

- **env:Sender – ter:InvalidArgVal – ter:NoProfile**
The requested profile token ProfileToken does not exist.
- **env:Sender – ter:InvalidArgVal – ter:NoPTZProfile**
The requested profile token does not reference a PTZ configuration.
- **env:Sender – ter:InvalidArgVal – ter:GeoMoveNotSupported**
The device does not support geo move.
- **env:Sender – ter:InvalidArgVal – ter:UnreachablePosition**
The requested translation is out of bounds.
- **env:Sender – ter:InvalidArgVal – ter:TimeoutNotSupported**
The requested position is not reachable.
- **env:Sender – ter:InvalidArgVal – ter:GeoLocationUnknown**
The unit is not able to perform GeoMove because its geolocation is not configured or available.

ACCESS CLASS:

ACTUATE

8.4.6 Stop

A PTZ-capable device shall support the Stop operation. If no stop filter arguments are present, this command stops all ongoing pan, tilt and zoom movements. The Stop operation can be filtered to stop a specific movement by setting the corresponding stop argument.

REQUEST:

- **ProfileToken [tt:ReferenceToken]**
Reference to an existing media profile.
- **PanTilt – optional [xs:boolean]**
Stop pan and tilt operation (defaults to true).
- **Zoom – optional [xs:boolean]**
Stop zoom operation (defaults to true).

RESPONSE:

This is an empty message.

FAULTS:

- **env:Sender – ter:InvalidArgVal – ter:NoProfile**
The requested profile token ProfileToken does not exist.
- **env:Sender – ter:InvalidArgVal – ter:NoPTZProfile**
The requested profile token does not reference a PTZ configuration.

ACCESS CLASS:

ACTUATE

8.4.7 GetStatus

A PTZ-capable device shall be able to report its PTZ status through the GetStatus command. The PTZ status contains the following information:

- Position (optional) – Specifies the absolute position of the PTZ unit together with the space references. The default absolute spaces of the corresponding PTZ configuration shall be referenced within the position element. This information shall be present if the device signals support via the capability StatusPosition.
- MoveStatus (optional) – Indicates if the pan/tilt/zoom device unit is currently moving, idle or in an unknown state. This information shall be present if the device signals support via the capability MoveStatus. The state Unknown shall not be used during normal operation, but is reserved to initialization or error conditions.
- Error (optional) – States a current PTZ error condition. This field shall be present if the MoveStatus signals Unknown.
- UTC Time – Specifies the UTC time when this status was generated.

REQUEST:

- **ProfileToken [tt:ReferenceToken]**
Reference to an existing media profile.

RESPONSE:

- **PTZStatus [tt:PTZStatus]**
PTZStatus for the requested MediaProfile.

FAULTS:

- **env:Sender – ter:InvalidArgVal – ter:NoProfile**
The requested profile token ProfileToken does not exist.
- **env:Sender – ter:InvalidArgVal – ter:NoPTZProfile**
The requested profile token does not reference a PTZ configuration.
- **env:Receiver – ter:Action – ter:NoStatus**
No PTZ status is available in the requested Media Profile.

ACCESS CLASS:

READ_MEDIA**8.5 Preset operations****8.5.1 General**

This section describes operations that manage the presets of a PTZ node. These operations shall be implemented for PTZ nodes supporting presets. All operations require a ProfileToken referencing a media profile that includes a PTZConfiguration.

8.5.2 SetPreset

The SetPreset command saves the current device position parameters so that the device can move to the saved preset position through the GotoPreset operation.

If the PresetToken parameter is absent, the device shall create a new preset. Otherwise it shall update the stored position and optionally the name of the given preset. If creation is successful, the response contains the PresetToken which uniquely identifies the preset. An existing preset can be overwritten by specifying the PresetToken of the corresponding preset. In both cases (overwriting or creation), an optional PresetName can be specified. The operation fails if the PTZ device is moving during the SetPreset operation.

The device may internally save additional states such as imaging properties in the PTZ preset which then should be recalled in the GotoPreset operation. A device shall accept a valid SetPresetRequest that does not include the optional element PresetName.

Devices may require unique preset names and reject a request that contains an already existing PresetName by responding with the error message ter:PresetExist.

REQUEST:

- **ProfileToken [tt:ReferenceToken]**
Reference to an existing media profile.
- **PresetToken – optional [tt:ReferenceToken]**
Optional existing preset token to update a preset position.
- **PresetName – optional [xs:string]**
Optional name to be assigned to the preset position.

RESPONSE:

- **PresetToken [tt:ReferenceToken]**
Reference token assigned by the device to the preset.

FAULTS:

- **env:Sender – ter:InvalidArgVal – ter:NoProfile**
The requested profile token ProfileToken does not exist.
- **env:Sender – ter:InvalidArgVal – ter:NoPTZProfile**
Reference to an existing preset token.
- **env:Sender – ter:InvalidArgVal – ter:PresetExist**
The requested name already exists for another preset.
- **env:Sender – ter:InvalidArgVal – ter:InvalidPresetName**
The PresetName is either too long or contains invalid characters.
- **env:Sender – ter:InvalidArgVal – ter:NoToken**
The requested preset token does not exist.
- **env:Receiver – ter:Action – ter:MovingPTZ**
Preset cannot be set while PTZ unit is moving.
- **env:Receiver – ter:Action – ter:TooManyPresets**
Maximum number of Presets reached.

ACCESS CLASS:

ACTUATE

8.5.3 GetPresets

The GetPresets operation returns the saved presets consisting of the following elements:

- Token – A unique identifier to reference the preset.
- Name – An optional mnemonic name.
- PTZ Position – An optional absolute position. If the PTZ node supports absolute pan/tilt position spaces, the pan/tilt position shall be specified. If the PTZ node supports absolute zoom position spaces, the zoom position shall be specified.

REQUEST:

- **ProfileToken [tt:ReferenceToken]**
Reference to an existing media profile.

RESPONSE:

- **Preset – optional, unbounded [tt:PTZPreset]**
List of presets for the requested MediaProfile.

FAULTS:

- **env:Sender – ter:InvalidArgVal – ter:NoProfile**
The requested profile token ProfileToken does not exist.
- **env:Sender – ter:InvalidArgVal – ter:NoPTZProfile**
The requested profile token does not reference a PTZ configuration.

ACCESS CLASS:

READ_MEDIA

8.5.4 GotoPreset

The GotoPreset operation recalls a previously set preset. If the speed parameter is omitted, the default speed of the corresponding PTZ configuration shall be used. The speed parameter can only be specified when speed spaces are available for the PTZ node. The GotoPreset command is a non-blocking operation and can be interrupted by other move commands.

REQUEST:

- **ProfileToken [tt:ReferenceToken]**
Reference to an existing media profile.
- **PresetToken [tt:ReferenceToken]**
Reference to an existing preset token.
- **Speed – optional [PTZSpeed]**
Optional speed.

RESPONSE:

This is an empty message.

FAULTS:

- **env:Sender – ter:InvalidArgVal – ter:NoProfile**
The requested profile token ProfileToken does not exist.
- **env:Sender – ter:InvalidArgVal – ter:NoPTZProfile**
Reference to an existing preset token.
- **env:Sender – ter:InvalidArgVal – ter:NoToken**
The requested preset token does not exist.
- **env:Sender – ter:InvalidArgVal – ter:SpaceNotSupported**
A space that is not supported by the PTZ node is referenced .
- **env:Sender – ter:InvalidArgVal – ter:InvalidSpeed**
The requested speed is out of bounds.

ACCESS CLASS:

ACTUATE

8.5.5 RemovePreset

The RemovePreset operation removes a previously set preset.

REQUEST:

- **ProfileToken [tt:ReferenceToken]**
Reference to an existing media profile.
- **PresetToken [tt:ReferenceToken]**
Existing preset token to be removed.

RESPONSE:

This is an empty message.

FAULTS:

- **env:Sender – ter:InvalidArgVal – ter:NoProfile**
The requested profile token ProfileToken does not exist.
- **env:Sender – ter:InvalidArgVal – ter:NoPTZProfile**
Reference to an existing preset token.
- **env:Sender – ter:InvalidArgVal – ter:NoToken**
The requested preset token does not exist.

ACCESS CLASS:

ACTUATE

8.6 Home position operations

8.6.1 General

Subclause 8.6 describes operations used to manage the home position of a PTZ node. These operations shall be implemented for PTZ nodes supporting home positions. All operations require a ProfileToken referencing a media profile that includes a PTZConfiguration.

The "home" position can be set by the SetHome operation or is a fixed position of the PTZ unit.

8.6.2 GotoHomePosition

This operation moves the PTZ unit to its home position. If the speed parameter is omitted, the default speed of the corresponding PTZ configuration shall be used. The speed parameter can only be specified when speed spaces are available for the PTZ node. The command is non-blocking and can be interrupted by other move commands.

REQUEST:

- **ProfileToken [tt:ReferenceToken]**
Reference to an existing media profile.
- **Speed – optional [PTZSpeed]**
Optional speed.

RESPONSE:

This is an empty message.

FAULTS:

- **env:Sender – ter:InvalidArgVal – ter:NoProfile**
The requested profile token ProfileToken does not exist.
- **env:Sender – ter:InvalidArgVal – ter:NoHomePosition**
No home position has been defined for this profile.
- **env:Sender – ter:InvalidArgVal – ter:NoPTZProfile**
Reference to an existing preset token.
- **env:Sender – ter:InvalidArgVal – ter:InvalidSpeed**
The requested speed is out of bounds.

ACCESS CLASS:

ACTUATE

8.6.3 SetHomePosition

The SetHome operation saves the current position parameters as the home position, so that the GotoHome operation can request that the device move to the home position.

The SetHomePosition command shall return with a failure if the "home" position is fixed and cannot be overwritten. If the SetHomePosition is successful, it shall be possible to recall the home position with the GotoHomePosition command.

REQUEST:

- **ProfileToken [tt:ReferenceToken]**
Reference to an existing media profile.

RESPONSE:

This is an empty message.

FAULTS:

- **env:Sender – ter:InvalidArgVal – ter:NoProfile**
The requested profile token ProfileToken does not exist.
- **env:Sender – ter:InvalidArgVal – ter:NoPTZProfile**
Reference to an existing preset token.
- **env:Receiver – ter:Action – ter:CannotOverwriteHome**
The home position is fixed and cannot be overwritten.

ACCESS CLASS:

ACTUATE

8.7 Auxiliary operations

8.7.1 General

This section describes operations to manage auxiliary commands of a PTZ node, such as an infrared (IR) lamp, a heater or a wiper.

These operations shall be implemented for PTZ nodes indicating auxiliary commands in the node properties. All operations require a ProfileToken referencing a media profile that includes PTZConfiguration.

8.7.2 SendAuxiliaryCommand

This operation is used to call an auxiliary operation on the device. The supported commands can be retrieved via the PTZ node properties. The auxiliary command should match the supported command listed in the PTZ node; no other syntax is supported. If the PTZ node lists the tt:IRLamp command, then the parameter of auxiliary command shall conform to the syntax specified in IEC 60839-11-31. SendAuxiliaryCommand shall be implemented when the PTZ node supports auxiliary commands.

REQUEST:

- **ProfileToken [tt:ReferenceToken]**
Reference to an existing media profile.
- **AuxiliaryData [tt:AuxiliaryData]**
Auxiliary command to be applied.

RESPONSE:

- **AuxiliaryResponse[tt:AuxiliaryData]**
Auxiliary response information.

FAULTS:

- **env:Sender – ter:InvalidArgVal – ter:NoProfile**
The requested profile token ProfileToken does not exist.
- **env:Sender – ter:InvalidArgVal – ter:NoPTZProfile**
Reference to an existing preset token.

ACCESS CLASS:

ACTUATE**8.8 Predefined PTZ Spaces****8.8.1 General**

Spaces are used to specify absolute, relative and continuous movements. Whereas absolute movements require an absolute position, relative movements are specified by a position translation. Continuous movements require the specification of a velocity (relative movement over time). For these three cases, different coordinate systems are used describing the desired movement. The generic spaces do not absolutely specify the underlying PTZ model, so that it can be applied to any PTZ hardware.

8.8.2 Absolute position spaces**8.8.2.1 General**

The absolute position spaces are used when the client wants to move the camera to a certain position. The absolute movement from a current position A to an arbitrarily chosen position B doesn't have to follow a specific path. Instead, the PTZ device may choose the shortest path in order to reach the target destination.

8.8.2.2 Generic pan/tilt position space

The generic pan/tilt position space shall be provided by every PTZ node that supports absolute pan/tilt, since it does not relate to a specific physical range. Instead, the range should be defined as the full range of the PTZ unit normalized to the range –1 to 1 resulting in the following space description:

```

<tt:AbsolutePanTiltPositionSpace>
  <tt:SpaceURI>
    http://www.onvif.org/ver10/tptz/PanTiltSpaces/PositionGenericSpace
  </tt:SpaceURI>
  <tt:Xrange>
    <tt:Min>-1.0</tt:Min>
    <tt:Max>1.0</tt:Max>
  </tt:Xrange>
  <tt:Yrange>
    <tt:Min>-1.0</tt:Min>
    <tt:Max>1.0</tt:Max>
  </tt:Yrange>
</tt:AbsolutePanTiltPositionSpace>

```

8.8.2.3 Spherical pan/tilt position space in degrees

Figure 7 shows a camera with pan and tilt mechanics and the corresponding spherical coordinate system. The space description assumes that the dome is mounted on the ceiling. The definition of a pan movement is the rotation of the camera module around the pan axis. Thereby, the tilt axis is also rotated in the same direction in the plane orthogonal to the pan axis, so that it is still orthogonal to the camera lens axis. Tilt movement is the rotation of the camera module around the tilt axis. With the tilt axis, the camera direction can be changed from looking downward to looking at the horizon. Some devices may support a camera which can look above the horizon.

The angles describing the rotation around pan and tilt axis are referred to as pan and tilt angles, where pan is represented by the X coordinate of the position vector and tilt is represented by the Y coordinate of the position vector. Both angles are specified in degrees. The initial position of this coordinate system is when the direction of the camera lens is parallel to the ceiling. The pan and tilt angles in this initial direction are zero (0,0).

When starting from the initial direction and increasing the pan angle, objects that have previously been in the centre of the image will move towards the left of the image. When starting from the initial direction and increasing the tilt angle, objects which have been previously in the middle of the image move towards the bottom of the image.

The maximum range for pan and tilt angles are between -180° and $+180^\circ$ ². The device can restrict the tilt range arbitrarily. The tilt angle of a camera can change its direction in the space of a hemisphere like a dome camera that is typically bounded from 0° to -90° . If a device cannot pan the full range, it may limit the pan range to an appropriate interval.

² -180° and $+180^\circ$ is the same pan position in space and the camera can choose to represent the pan position using either of them.

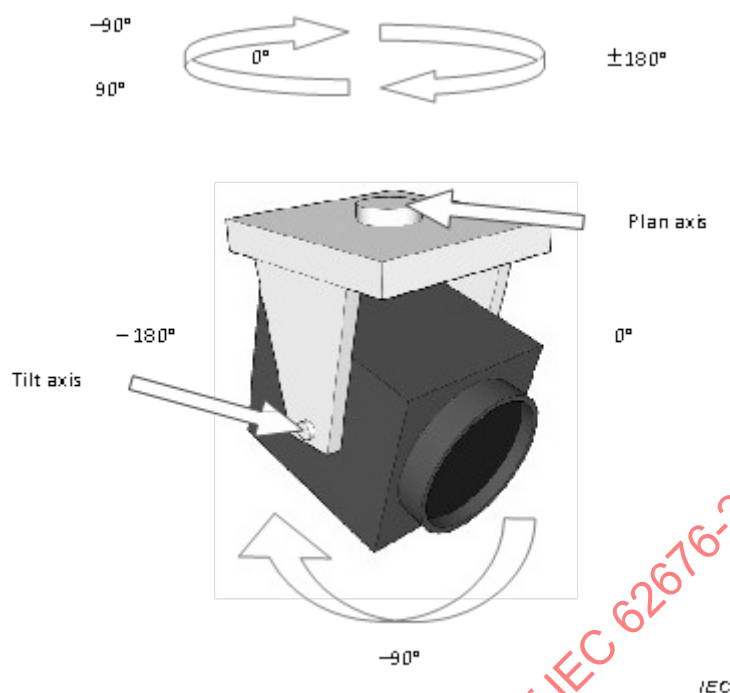


Figure 7 – Spherical pan/tilt position space in degrees for a camera mounted on the ceiling

A camera that does not support e-flip (or has disabled it) and allows an absolute tilt range between 0° and -180° shall not rotate the image and directions when requesting a tilt movement of less than -90° .

If a camera supports e-flip and a client requests a tilt movement that passes the nadir, the following behaviour is to be expected:

- There could be hysteresis ($\pm\alpha^\circ$) room around the position of the nadir, such that the image and direction rotation doesn't occur exactly at the nadir, which could generate irritating oscillation.
- When the camera rotates the image and move directions (after passing the nadir $\pm\alpha^\circ$), the following changes to the coordinate system will also occur:
 - The new pan angle will be $P_{\text{new}} = P_{\text{old}} + 180^\circ$
 - The new tilt angle will be $T_{\text{new}} = -180^\circ - T_{\text{old}}$, $-180 \leq T_{\text{old}} \leq -90$
 - If ignoring the hysteresis area, there is a one-to-one mapping between viewing directions and pan/tilt coordinates, even though there are two mechanical positions of the camera matching the same viewing direction but with a rotated image. For example, the two mechanical directions $P_1: 35$, $T_1: -20$ and $P_2: -145$, $T_2: -160$, would be represented by the coordinate $P: 35$, $T: -20$ since a tilt of $< -90 - \alpha$ would update the coordinates as described above.

The spherical pan/tilt position space is defined as follows:

```

<tt:AbsolutePanTiltPositionSpace>
  <tt:SpaceURI>
    http://www.onvif.org/ver10/tptz/PanTiltSpaces/SphericalPositionSpaceDegrees
  </tt:SpaceURI>
  <tt:XRange>
    <tt:Min>-INF</tt:Min>
    <tt:Max>INF</tt:Max>
  </tt:XRange>
  <tt:YRange>
    <tt:Min>-180.0</tt:Min>
    <tt:Max>180.0</tt:Max>
  </tt:YRange>
</tt:AbsolutePanTiltPositionSpace>

```

The particular device should reference the proposed name space but the actual ranges are device-specific. The ranges indicated in the definition shall be interpreted as the maximum and minimum range limits for the space.

The Min and Max elements of the XRange are set to plus and minus infinity to indicate that the range is not bounded to either side.

8.8.2.4 Generic zoom position space

The generic zoom position space shall be provided by every PTZ node that supports absolute zoom, since it does not relate to a specific physical range. Instead, the range should be defined as the full range of the zoom normalized to the range 0 (wide) to 1 (tele). There is no assumption about how the generic zoom range is mapped to magnification, field of view or other physical zoom dimension. This results in the following space description:

```

<tt:AbsoluteZoomPositionSpace>
  <tt:SpaceURI>
    http://www.onvif.org/ver10/tptz/ZoomSpaces/PositionGenericSpace
  </tt:SpaceURI>
  <tt:Xrange>
    <tt:Min>0.0</tt:Min>
    <tt:Max>1.0</tt:Max>
  </tt:Xrange>
</tt:AbsoluteZoomPositionSpace>

```

8.8.2.5 Zoom position space in millimetres

The zoom position space in Millimetres expresses the absolute focal length of a zoom camera. The maximum and minimum focal length is camera specific. The zoom position space in millimetres is defined as follows:

```

<tt:AbsoluteZoomPositionSpace>
  <tt:SpaceURI>
    http://www.onvif.org/ver10/tptz/ZoomSpaces/PositionSpaceMillimeter
  </tt:SpaceURI>
  <tt:XRange>
    <tt:Min>0.0</tt:Min>
    <tt:Max>INF</tt:Max>
  </tt:XRange>
</tt:AbsoluteZoomPositionSpace>

```

The particular device should reference the proposed name space but the actual range is device specific. The range indicated in the definition shall be interpreted as the maximum and minimum range limits for the space.

The current zoom magnification (M) for a camera can be calculated from the focal length as follows:

$$M = \frac{f_c}{f_0}$$

where

f_0 is the minimum focal length, and

f_c is the current focal length.

The current angle of view can be calculated with the following formula:

$$\alpha = 2 \arctan \frac{d}{2f_c},$$

where

d represents the size of the sensor in the direction of interest, and

f_c is the current focal length.

8.8.2.6 Normalized digital zoom position space

8.8.2.6.1 General

The normalized digital zoom position space represents the mapping between the width of the window used for a video stream and the width of the video source (sensor). This space requires a media profile that includes VideoEncoderConfiguration and VideoSourceConfiguration.

The normalized digital zoom factor is calculated as: $Z = 1 / \left(\frac{W_w}{S_w} \right)$.

The maximum zoom level that can be represented by this space is limited by the smallest size of the window allowed by the camera.

The normalized digital position zoom space is defined as follows:

```
<tt:AbsoluteZoomPositionSpace>
  <tt:SpaceURI>
    http://www.onvif.org/ver10/tptz/ZoomSpaces/NormalizedDigitalPositionSpace
  </tt:SpaceURI>
  <tt:XRange>
    <tt:Min>1.0</tt:Min>
    <tt:Max>INF</tt:Max>
  </tt:XRange>
</tt:AbsoluteZoomPositionSpace>
```

The particular device should reference the proposed name space but the actual range is device specific. The range indicated in the definition shall be interpreted as the maximum range limit for the space.

It's important to notice that a zoom operation using this space doesn't affect the current resolution of the video stream being configured in VideoEncoderConfiguration of the media profile. Instead, it changes the width and height of the window configured in VideoSourceConfiguration.

8.8.2.6.2 Example: Zoom level for cropped area

For example, a Digital PTZ unit with a natively cropped VGA (640 × 480) window on a 3 MP sensor (2048 × 1536) would have the zoom level $Z = 1 / \left(\frac{640}{2048} \right) = 3,2$.

8.8.3 Relative translation spaces

8.8.3.1 General

A relative pan/tilt translation space moves the PTZ unit a certain translation in a certain direction without knowing the camera's current pan/tilt position.

A relative pan/tilt translation can be derived from a corresponding (digital/spherical) absolute pan/tilt position space by taking the difference of two absolute pan/tilt positions. However, there are also relative pan/tilt translations where no corresponding absolute pan/tilt space can be defined.

8.8.3.2 Generic pan/tilt translation space

The generic pan/tilt translation space shall be provided by every PTZ node that supports relative pan/tilt, since it does not relate to a specific physical range. Instead, the range should be defined as the full positive and negative translation range of the PTZ unit normalized to the range –1 to 1, where positive translation would mean clockwise rotation or movement in right/up direction resulting in the following space description:

```
<tt:RelativePanTiltTranslationSpace>
  <tt:SpaceURI>
    http://www.onvif.org/ver10/tptz/PanTiltSpaces/TranslationGenericSpace
  </tt:SpaceURI>
  <tt:Xrange>
    <tt:Min>-1.0</tt:Min>
    <tt:Max>1.0</tt:Max>
  </tt:Xrange>
  <tt:Yrange>
    <tt:Min>-1.0</tt:Min>
    <tt:Max>1.0</tt:Max>
  </tt:Yrange>
</tt:RelativePanTiltTranslationSpace>
```

8.8.3.3 Generic zoom translation space

The generic zoom translation space shall be provided by every PTZ node that supports relative zoom, since it does not relate to a specific physical range. Instead, the corresponding absolute range should be defined as the full positive and negative translation range of the zoom normalized to the range –1 to 1, where a positive translation maps to a movement in the tele direction. The translation is signed to indicate direction (negative is to wide, positive is to tele). There is no assumption about how the generic zoom range is mapped to magnification, field of view or other physical zoom dimension. This results in the following space description:

```
<tt:RelativeZoomTranslationSpace>
  <tt:SpaceURI>
    http://www.onvif.org/ver10/tptz/ZoomSpaces/TranslationGenericSpace
  </tt:SpaceURI>
  <tt:Xrange>
    <tt:Min>-1.0</tt:Min>
    <tt:Max>1.0</tt:Max>
  </tt:Xrange>
</tt:RelativeZoomTranslationSpace>
```

8.8.4 Continuous velocity spaces

8.8.4.1 General

The continuous velocity spaces are used to continuously move the PTZ unit in a certain direction.

If a camera supports e-flip, and it's enabled, the following behaviour should be expected by the pan/tilt spaces defined in 8.8.4.3:

If the tilt translation is passing the nadir position (including the room for a hysteresis $\pm\alpha^\circ$), the camera should rotate the image at nadir $\pm\alpha^\circ$ as described in 8.8.2.3 (only for cameras supporting absolute pan/tilt positioning). When a rotation occurs, the camera will continue the current movements according to the directions given when the command was issued. If the command is interrupted with a new request (after the flip), that request will be handled according to the new (flipped) direction and coordinates. A camera that does not support e-flip or has it disabled will not rotate the image and directions during a tilt movement passing the nadir.

8.8.4.2 Generic pan/tilt velocity space

The generic pan/tilt velocity space shall be provided by every PTZ node, since it does not relate to a specific physical range. Instead, the range should be defined as a range of the PTZ unit's speed normalized to the range -1 to 1 , where a positive velocity would map to a clockwise rotation or movement in the right/up direction. A signed speed can be independently specified for the pan and tilt component resulting in the following space description:

```
<tt:ContinuousPanTiltVelocitySpace>
  <tt:SpaceURI>
    http://www.onvif.org/ver10/tptz/PanTiltSpaces/VelocityGenericSpace
  </tt:SpaceURI>
  <tt:Xrange>
    <tt:Min>-1.0</tt:Min>
    <tt:Max>1.0</tt:Max>
  </tt:Xrange>
  <tt:Yrange>
    <tt:Min>-1.0</tt:Min>
    <tt:Max>1.0</tt:Max>
  </tt:Yrange>
</tt:ContinuousPanTiltVelocitySpace>
```

8.8.4.3 Pan/tilt velocity space in degrees per second

The pan/tilt velocity space in degrees per second describes a continuous rotation around the pan and tilt axis (specified in 8.8.2.3) with a constant rotation speed. Thereby, the velocity describes the signed pan and tilt angle change in degrees per second. The pan velocity is expressed by the X coordinate of the direction vector and the tilt velocity by the Y coordinate and the space is defined as follows:

```
<tt:ContinuousPanTiltVelocitySpace>
  <tt:SpaceURI>
    http://www.onvif.org/ver10/tptz/PanTiltSpaces/VelocitySpaceDegrees
  </tt:SpaceURI>
  <tt:XRange>
    <tt:Min>-INF</tt:Min>
    <tt:Max>INF</tt:Max>
  </tt:XRange>
  <tt:YRange>
    <tt:Min>-INF</tt:Min>
    <tt:Max>INF</tt:Max>
  </tt:YRange>
</tt:ContinuousPanTiltVelocitySpace>
```


The particular device should reference the proposed name space, but the actual ranges are device specific. The ranges indicated in the definition shall be interpreted as the maximum and minimum range limits for the space.

8.8.4.4 Generic zoom velocity space

The generic zoom velocity space specifies a zoom factor velocity without knowing the underlying physical model. The range should be normalized from -1 to 1 , where a positive velocity would map to the tele direction. A generic zoom velocity space description resembles the following:

```
<tt:ContinuousZoomVelocitySpace>
  <tt:SpaceURI>
    http://www.onvif.org/ver10/tptz/ZoomSpaces/VelocityGenericSpace
  </tt:SpaceURI>
  <tt:Xrange>
    <tt:Min>-1.0</tt:Min>
    <tt:Max>1.0</tt:Max>
  </tt:Xrange>
</tt:ContinuousZoomVelocitySpace>
```

8.8.5 Speed spaces

8.8.5.1 General

The speed spaces specify the speed for a pan/tilt and zoom movement when moving to an absolute position or to a relative translation. In contrast to the velocity spaces, speed spaces do not contain any directional information. The speed of a combined pan/tilt movement is represented by a single non-negative scalar value.

If relative translation space and continuous velocity space are already defined, the corresponding speed space is derived as follows: requesting a continuous movement with a velocity V for T seconds, is identical (up to acceleration and positional inaccuracies) to requesting a relative movement with relative position R and speed S , where R equals V times T and S equals the length of vector V . Therefore, speed values are always positive.

8.8.5.2 Generic pan/tilt speed space

The generic pan/tilt speed space shall be provided by every PTZ node that supports configurable speed for pan/tilt, since it does not relate to a specific physical range. Instead, the range should be defined as the full range of the speed range normalized to the range 0 (stopped) to 1 (full speed). This results in the following space description:

```
<tt:PanTiltSpeedSpace>
  <tt:SpaceURI>
    http://www.onvif.org/ver10/tptz/PanTiltSpaces/GenericSpeedSpace
  </tt:SpaceURI>
  <tt:Xrange>
    <tt:Min>0.0</tt:Min>
    <tt:Max>1.0</tt:Max>
  </tt:Xrange>
</tt:PanTiltSpeedSpace>
```

8.8.5.3 Pan/tilt speed space in degrees per second

The pan/tilt speed space in degrees per second derives from the absolute pan/tilt position space in degrees for absolute and relative movements. The pan/tilt speed space in degrees per second is defined as follows:

```
<tt:PanTiltSpeedSpace>
  <tt:SpaceURI>
    http://www.onvif.org/ver10/tptz/PanTiltSpaces/SpeedSpaceDegrees
  </tt:SpaceURI>
```

```

<tt:XRange>
  <tt:Min>0.0</tt:Min>
  <tt:Max>INF</tt:Max>
</tt:XRange>
</tt:PanTiltSpeedSpace>

```

The particular device should reference the proposed name space, but the actual range is device specific. The range indicated in the definition shall be interpreted as the maximum and minimum range limits for the space.

8.8.5.4 Generic zoom speed space

The generic zoom speed space shall be provided by every PTZ node that supports configurable speed for Zoom, since it does not relate to a specific physical range. Instead, the range should be defined as the full range of the speed range normalized to the range 0 (stopped) to 1 (full speed). This results in the following space description:

```

<tt:ZoomSpeedSpace>
  <tt:SpaceURI>
    http://www.onvif.org/ver10/tptz/ZoomSpaces/ZoomGenericSpeedSpace
  </tt:SpaceURI>
  <tt:XRange>
    <tt:Min>0.0</tt:Min>
    <tt:Max>1.0</tt:Max>
  </tt:XRange>
</tt:ZoomSpeedSpace>

```

8.8.5.5 Zoom speed space in millimetres per second

The zoom speed space in millimetres per second specifies the absolute change per second of the focal length. If the device supports the zoom speed space, the zoom speed space in millimetres per second is defined as follows:

```

<tt:ZoomSpeedSpace>
  <tt:SpaceURI>
    http://www.onvif.org/ver10/tptz/ZoomSpaces/SpeedSpaceMillimeter
  </tt:SpaceURI>
  <tt:XRange>
    <tt:Min>0.0</tt:Min>
    <tt:Max>INF</tt:Max>
  </tt:XRange>
</tt:ZoomSpeedSpace>

```

The particular device should reference the proposed name space, but the actual range is device specific. The range indicated in the definition shall be interpreted as the maximum and minimum range limit for the space.

8.9 Preset tour operations

8.9.1 General

Preset tour is a feature for PTZ-capable devices, enabling the PTZ unit to move on specified presets sequentially at a certain interval.

Subclause 8.9 describes operations that manage the preset tours. These operations shall be implemented when a PTZ node in the PTZ service indicates support of preset tours with `MaximumNumberOfPresetTours > 0` capability value. All operations require a `ProfileToken` referencing a media profile that includes a `PTZConfiguration`. All operations for preset tours shall always be persistent.

`PresetTours` can have forward and backward directions, and if the direction is omitted, it is assumed to be forward. However, it is possible to specify random execution by setting the

RandomPresetOrder attribute. If RandomPresetOrder is set to true and Direction is also present, Direction will be ignored and presets of the tour will be recalled randomly.

8.9.2 GetPresetTours

A device supporting preset tours feature shall return all available preset tours through GetPresetTours.

REQUEST:

- **ProfileToken [tt:ReferenceToken]**
Reference to an existing media profile.

RESPONSE:

- **PresetTour – optional, unbounded [tt:PTZPresetTour]**
List of preset tours defined for the requested media profile.

FAULTS:

- **env:Sender – ter:InvalidArgVal – ter:NoProfile**
The requested profile token does not exist.
- **env:Sender – ter:InvalidArgVal – ter:NoPTZProfile**
The requested profile token does not reference a PTZ configuration.

ACCESS CLASS:

READ_MEDIA

8.9.3 GetPresetTour

A device supporting PresetTour shall return the requested preset tour through GetPresetTour.

REQUEST:

- **ProfileToken [tt:ReferenceToken]**
Reference to an existing media profile.
- **PresetTourToken [tt:ReferenceToken]**
Reference to an existing preset tour.

RESPONSE:

- **PresetTour [tt:PTZPresetTour]**
The requested preset tour.

FAULTS:

- **env:Sender – ter:InvalidArgVal – ter:NoProfile**
The requested profile token ProfileToken does not exist.
- **env:Sender – ter:InvalidArgVal – ter:NoPTZProfile**
The requested profile token does not reference a PTZ configuration.
- **env:Sender – ter:InvalidArgVal – ter:NoToken**
The requested preset tour token does not exist.

ACCESS CLASS:

READ_MEDIA

8.9.4 GetPresetTourOptions

A device supporting PresetTour shall provide options for how preset tours can be configured through GetPresetTourOptions.

REQUEST:

- **ProfileToken [tt:ReferenceToken]**
Reference to an existing media profile.
- **PresetTourToken – optional [tt:ReferenceToken]**
Optional reference to an existing preset tour.

RESPONSE:

- **PresetTour [tt:PTZPresetTourOptions]**
The requested preset tour options.

FAULTS:

- **env:Sender – ter:InvalidArgVal – ter:NoProfile**
The requested profile token ProfileToken does not exist.
- **env:Sender – ter:InvalidArgVal – ter:NoPTZProfile**
The requested profile token does not reference a PTZ configuration.

ACCESS CLASS:

READ_MEDIA

8.9.5 CreatePresetTour

A device supporting preset tours shall allow creating a new preset tour through CreatePresetTour.

REQUEST:

- **ProfileToken [tt:ReferenceToken]**
Reference to an existing media profile.

RESPONSE:

- **PresetTourToken [tt:ReferenceToken]**
Reference to an existing preset tour.

FAULTS:

- **env:Sender – ter:InvalidArgVal – ter:NoProfile**
The requested profile token does not exist.
- **env:Sender – ter:InvalidArgVal – ter:NoPTZProfile**
The requested profile token does not reference a PTZ configuration.
- **env:Sender – ter:InvalidArgVal – ter:TooManyPresetTours**
There is not enough space in the device to create the new preset tour for the profile

ACCESS CLASS:

ACTUATE

8.9.6 ModifyPresetTour

A device supporting preset tours shall allow modifying a preset tour through ModifyPresetTour.

REQUEST:

- **ProfileToken [tt:ReferenceToken]**
Reference to an existing media profile.
- **PresetTour [tt:PTZPresetTour]**
The preset tour parameters to be set.

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:NoPTZProfile**
The requested profile token does not reference a PTZ configuration.
- **env:Sender – ter:InvalidArgVal – ter:InvalidPresetTour**
The requested PresetTour includes invalid parameter(s).
- **env:Sender – ter:InvalidArgVal – ter:TooManyPresets**
Too many TourSpots are included in the PresetTour.
- **env:Sender – ter:InvalidArgVal – ter:NoToken**
The requested preset tour token does not exist.
- **env:Sender – ter:InvalidArgVal – ter:SpaceNotSupported**
A space is referenced which is not supported by the PTZ node

ACCESS CLASS:

ACTUATE

8.9.7 OperatePresetTour

A device supporting the PresetTour feature shall allow starting, stopping, or pausing a preset tour through OperatePresetTour.

A preset tour can be operated with the PresetTourOperation parameter of the OperatePresetTour command.

- **Start:** indicates starting the preset tour or re-starting the paused preset tour.
- **Stop:** indicates stopping the preset tour.
- **Pause:** indicates pausing the preset tour.

When receiving another OperatePresetTour start operation command for a preset tour that has already been started, the preset tour shall be restarted with the newly requested parameter.

REQUEST:

- **ProfileToken [tt:ReferenceToken]**
Reference to an existing media profile.
- **PresetTourToken [tt:PTZPresetTourToken]**
Reference to an existing preset tour.
- **Operation [tt:PTZPresetTourOperation]**
Operation information.

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:NoPTZProfile**
The requested profile token does not reference a PTZ configuration.
- **env:Sender – ter:InvalidArgVal – ter:NoToken**
The requested preset tour token does not exist.
- **env:Sender – ter:InvalidArgVal – ter:InvalidPresetTour**
The requested PresetTour includes invalid parameter(s).
- **env:Receiver – ter:InvalidArgVal – ter:ActivationFailed**
The requested preset tour cannot be activated while PTZ unit is moving or another preset tour is now activated.

ACCESS CLASS:

ACTUATE

8.9.8 RemovePresetTour

A device supporting PresetTour shall support removing preset tours through RemovePresetTour.

REQUEST:

- **ProfileToken [tt:ReferenceToken]**
Reference to an existing media profile.
- **PresetTourToken [tt:PTZPresetTourToken]**
Reference to an existing preset tour.

RESPONSE:

This is an empty message.

FAULTS:

- **env:Sender – ter:InvalidArgVal – ter:NoProfile**
The requested profile token ProfileToken does not exist.
- **env:Sender – ter:InvalidArgVal – ter:NoPTZProfile**
The requested profile token does not reference a PTZ configuration.

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

The requested preset tour token does not exist.

ACCESS CLASS:

ACTUATE

8.9.9 Preset tour parameters

A preset tour is composed of the following parameters:

PresetTour: Configuration of a preset tour.

- Name: name of the preset tour [Optional]
- Status: indicates the current status of the preset tour. This parameter in the argument of the ModifyPresetTour command shall be ignored.
 - Idle: preset tour is not activated.
 - Touring: preset tour is activated and now under touring.
 - Paused: preset tour is under pause at the middle of the tour by client. This status requires Start operation from a client to re-start.
- AutoStart: flag to enable the preset tour activated always. When this flag in one of preset tours is configured as True, ones in all other preset tours referring the same PTZNode shall be maintained to false. Starting and stopping the preset tour by OperatePresetTour shall change this flag to False. Pausing and resuming the preset tour by OperatePresetTour shall not change this flag.
- TourSpot: presets with its configuration for touring [Optional]
 - PresetDetail: detail configuration of the preset for the tour spot. One out of the three options below shall be chosen.
 - i) PresetToken: specifies an existing PresetToken as this tour spot.
 - ii) Home: specifies the home position as the tour spot. Only "True" to this Boolean parameter shall be valid. "False" shall be treated as an error due to an invalid argument.
 - iii) PTZPosition: specifies the vector of PTZ node as the tour spot
 - Speed: PTZ speed moving to this tour spot. [Optional]
 - StayTime: time duration of stay on the spot. [Optional]
- PresetTourStartingCondition: Configuration of the starting preset tour.
 - RecurringTime: parameter to specify how many times the preset tour is recurred. [Optional]
 - RecurringDuration: parameter to specify how long duration time the preset tour is recurred. [Optional]
 - i) If both the conditions RecurringTime and RecurringDuration are configured, the preset tour shall finish with one of the conditions satisfied earlier. If both are omitted, PresetTour shall be recurred without limitation.
 - PresetTourDirection: a device refers to this flag to choose which direction the preset tour goes. It is an omissible parameter and Forward shall be chosen if it is omitted. [Optional]
 - i) Forward: preset tour goes in forward order.
 - ii) Backward: preset tour goes in backward order.
- Token: unique identifier of the preset tour.

8.10 Pan/tilt control direction configuration

Pan/tilt control direction configuration is one set of parameters in PTZ configuration used to configure the movement directions of pan and tilt in the two following ways:

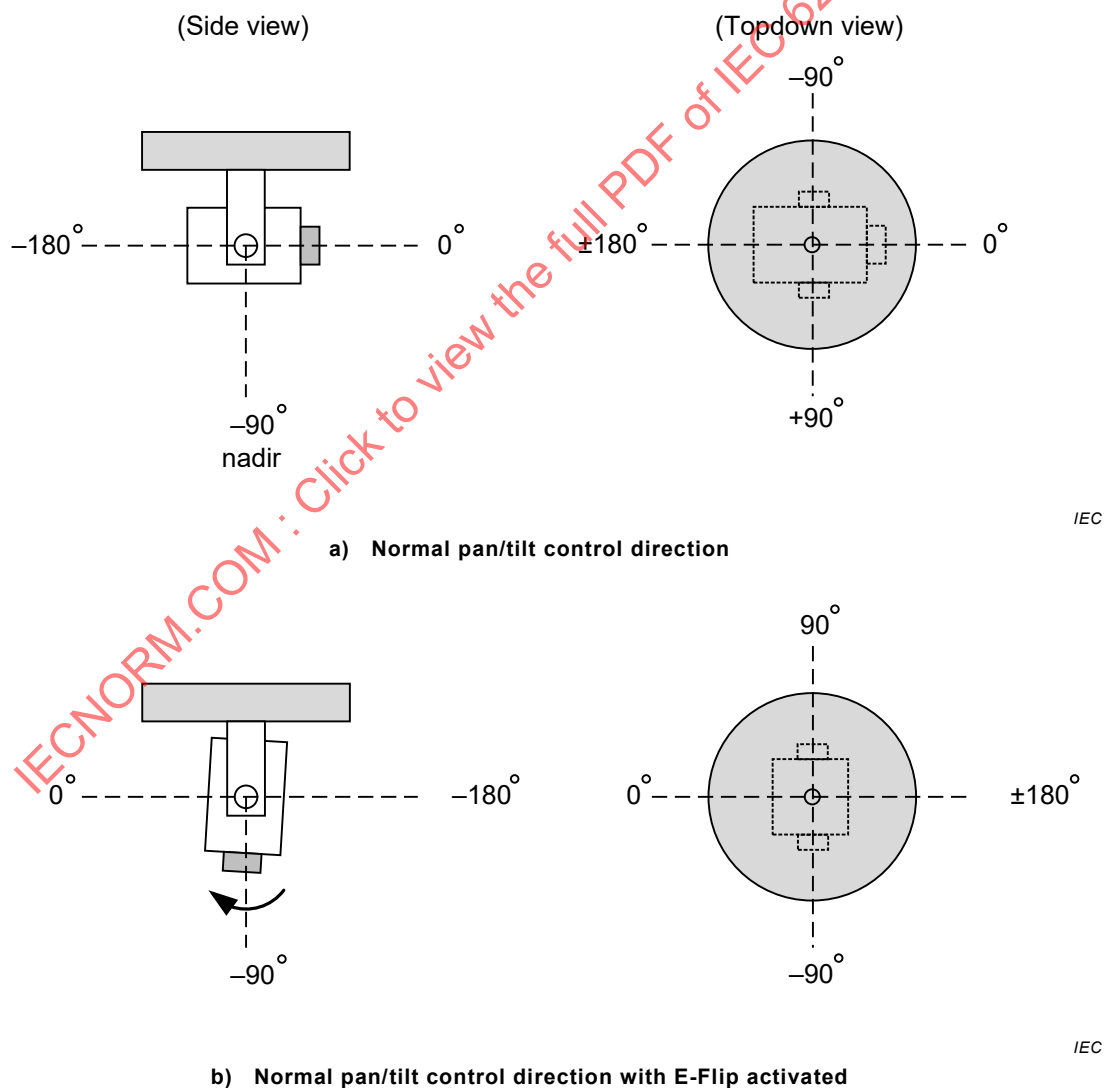
1) E-Flip

E-Flip is the flipping of the control directions of PT when the lens passes the nadir or the zenith direction during a tilt movement. This functionality is useful when controlling domes using a human joystick control, where a client can track an object that passes the nadir and doesn't have to bother about inverted controls.

2) Reverse

The Reverse parameter is for switching the positive and negative positions of the pan/tilt control direction. This feature can be used to configure appropriate control direction of PT based on directions that the device is mounted, for example, in the ceiling downwards or on the surface upwards.

Figure 8 shows an example how the control directions of pan/tilt are changed by E-Flip feature and Reverse feature.



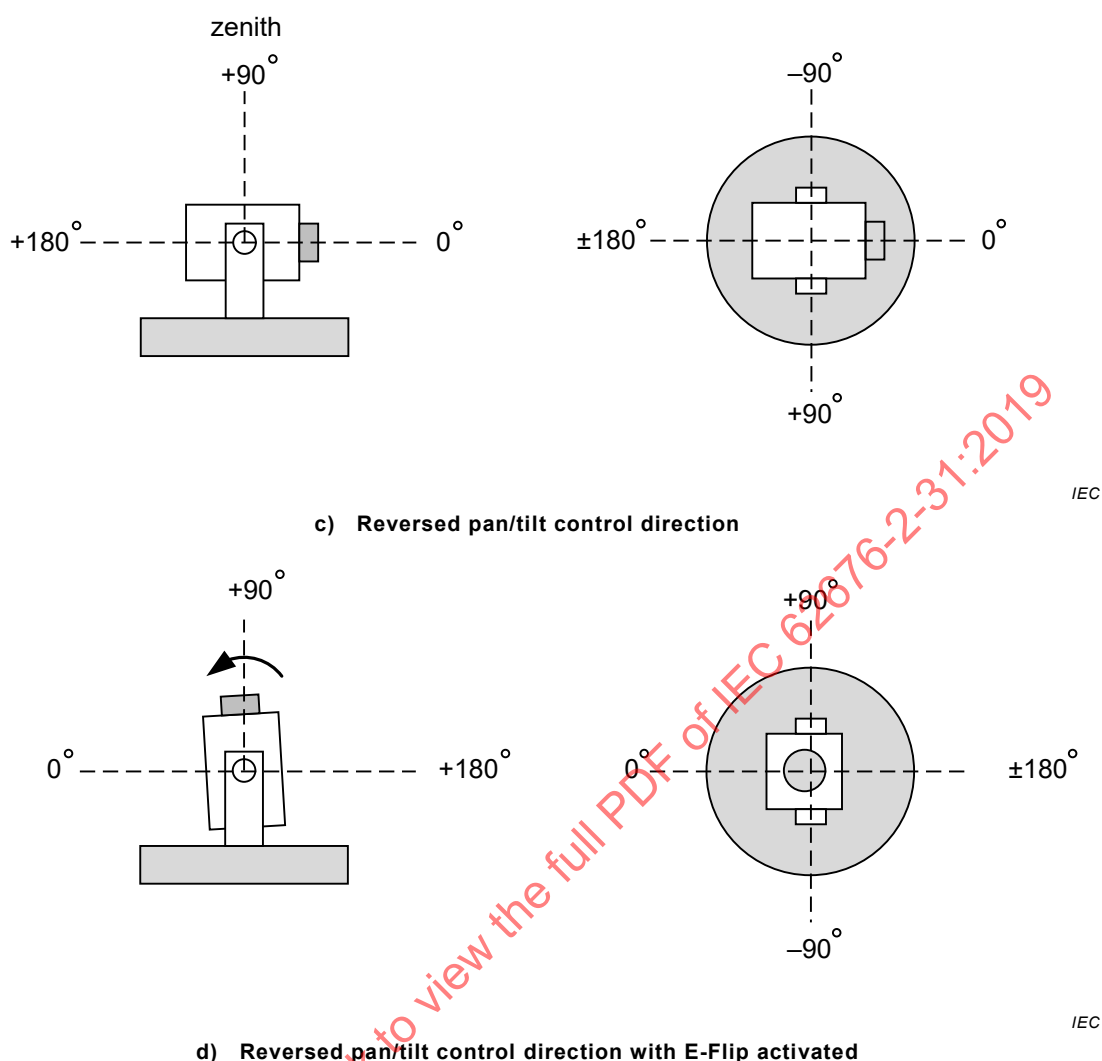


Figure 8 – Example of changes of pan/tilt control direction by E-Flip and Reverse

If the control direction of pan/tilt is changed, it is recommended to adapt the orientation of the video image via the Rotation parameter of VideoSourceConfiguration to align pan/tilt and video.

8.11 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:

EFlip:	Indicates whether or not E-Flip is supported.
Reverse:	Indicates whether or not reversing of pan/tilt control direction is supported.
GetCompatibleConfigurations:	Indicates the support for GetCompatibleConfigurations command.
MoveStatus:	Indicates that the PTZStatus includes MoveStatus information.
StatusPosition:	Indicates that the PTZStatus includes Position information.

REQUEST:

This is an empty message.

RESPONSE:

- **Capabilities [tptz:Capabilities]**

List of the above defined capabilities.

FAULTS:

None

ACCESS CLASS:

PRE_AUTH

8.12 Events

8.12.1 General

A device supporting PTZ service dispatches events listed in Subclause 8.12 through the event service.

8.12.2 PTZ presets

A device supporting the PTZ service specifies handling of PTZ presets. Since the move operations are non-blocking, a client is not informed when the PTZ preset has been reached. Therefore, a device supporting the PTZ service should support the following events, which inform subscribers about the status of preset movements.

Topic:

```
tns1:PTZController/PTZPresets/Invoked
tns1:PTZController/PTZPresets/Reached
tns1:PTZController/PTZPresets/Aborted
tns1:PTZController/PTZPresets/Left
```

The typical sequence of events is that first a client requests a certain preset. When the device accepts this request, it will send out an invoked event. The invoked event has to follow either a reached event or an aborted event. The former is used when the PTZ unit was able to reach the invoked preset position, the latter in any other case. A reached event has to follow a left event, as soon as the PTZ unit moves away from the preset position.

The message structure of these events is given by the following message description:

```
<tt:MessageDescription>
  <tt:Source>
    <tt:SimpleItemDescription Name="PTZConfigurationToken"
      Type="tt:ReferenceToken"/>
  </tt:Source>
  <tt:Data>
    <tt:SimpleItemDescription Name="PresetToken" Type="tt:ReferenceToken"/>
    <tt:SimpleItemDescription Name="PresetName" Type="tt:Name"/>
  </tt:Data>
</tt:MessageDescription>
```

8.12.3 PresetTours

When a PTZ node in a device supporting the PTZ service indicates support of Preset Tour with MaximumNumberOfPresetTours greater than 0 capability value, the device shall support the following topic to inform subscribers about preset tour events. Whenever a change in Preset Tours occurs, the PTZ service shall dispatch this event:

Topic: tns1:PTZController/PTZPresetTours/Configuration

```
<tt:MessageDescription isProperty=True>
  <tt:Source>
    <tt:SimpleItemDescription Name="ProfileToken" Type="tt:ReferenceToken"/>
    <tt:SimpleItemDescription Name="PresetTourToken" Type="tt:ReferenceToken"/>
  </tt:Source>
  <tt:Data>
    <tt:SimpleItemDescription Name="StateChange" Type="xs:boolean"/>
    <tt:ElementItemDescription Name="PresetTour" Type="tt:PTZPresetTour"/>
  </tt:Data>
</tt:MessageDescription>
```

Note that the StateChange flag in the data part becomes "true" when the mentioned Preset Tour includes any changes of its state.

9 Analytics service

9.1 General

This clause covers the following main areas of this architecture:

- analytics module interface;
- scene description;
- rules interface.

The analytics service allows fine-grained configuration of individual rules and individual analytics modules (see 9.2.3.5 and 9.3.4.7). Subclause 9.2 introduces the XML-based scene description, which can be streamed as metadata to clients via RTP as defined in 10.2.2.1.2. Additional rule and analytics module definitions are located in Annex C, Annex D and Annex E. For the schema definition, see Clause F.5.

9.2 Scene description interface

9.2.1 Overview

This specification defines the XML schema that shall be used to encode scene descriptions by a device. The scope of the Scene Description covers basic Scene Elements that can be displayed in a video overlay to the end-user as well as a framework for vendor-specific extensions. Annex A shows additional Scene Elements that may be used for processing vendor-specific rules.

9.2.2 Frame-related content

9.2.2.1 General

The input of the analytics engine is images from a video source. The extracted scene elements are associated with the image from which they were extracted. An extracted scene is distinguished from the general description of the video source processed by the analytics engine (information such as video input line, video resolution, frame cropping, frame rate), the temporal frame association within the input stream, and the spatial positioning of elements within a frame.

The temporal and spatial relation of scene elements with respect to the selected video source is discussed in 9.2.2.2 and 9.2.2.3. The appearance and behaviour of tracked objects is discussed in 9.2.3.2. Interactions between objects such as splits and merges are described in 9.2.3.3.

A PTZ device can put information about the pan, tilt and zoom at the beginning of a frame, allowing a client to estimate the 3D coordinates of scene elements. Next, the image coordinate system can be adapted with an optional transformation node, which is described in 9.2.2.2. Finally, multiple object descriptions can be placed, and their association can be specified within an ObjectTree node. Below, the definitions are included for convenience³:

```
<xs:complexType name="Frame">
  <xs:sequence>
    <xs:element name="PTZStatus" type="tt:PTZStatus"
      minOccurs="0"/>
    <xs:element name="Transformation" type="tt:Transformation"
      minOccurs="0"/>
    <xs:element name="Object" type="tt:Object" minOccurs="0"
      maxOccurs="unbounded"/>
    <xs:element name="ObjectTree" type="tt:ObjectTree" minOccurs="0"/>
    ...
  </xs:sequence>
  <xs:attribute name="UtcTime" type="xs:dateTime" use="required"/>
  ...
</xs:complexType>

<xs:element name="Frame" type="tt:Frame">
```

Subclause 9.2.2.2 describes how frames processed by the analytics algorithm are referenced within the analytics stream.

9.2.2.2 Temporal relation

Since multiple scene elements can be extracted from the same image, scene elements are listed below a frame node that establishes the link to a specific image from the video input. The frame node contains a mandatory UtcTime attribute. This UtcTime timestamp shall enable a client to map the frame node exactly to one video frame. For example, the RTP timestamp of the corresponding encoded video frame shall result in the same UTC timestamp after conversion. The synchronization between video and metadata streams is further described in 10.2.2.2.2.

Example:

```
<tt:Frame UtcTime="2008-10-10T12:24:57.321">
  ...
</tt:Frame>
...
<tt:Frame UtcTime="2008-10-10T12:24:57.521">
  ...
</tt:Frame>
```

9.2.2.3 Spatial relation

Most scene elements refer to some part in an image from which information has been extracted. For instance, when tracking objects over time, their position within each frame shall be specified. These positions shall relate to a coordinate system. The default coordinate system is shown in Figure 9.

³ Please note that the schema is included here for information purposes only. See Clause F.8 for the normative schema definition.

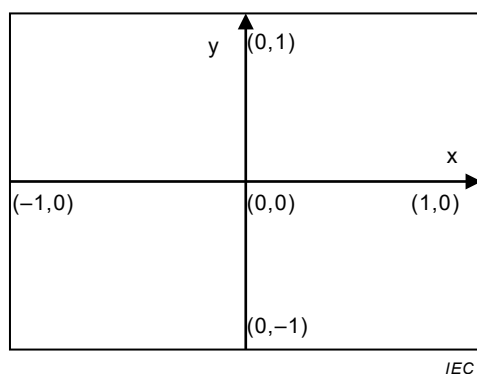


Figure 9 – Default frame coordinate system

This specification allows modification of the coordinate system for individual nodes of the XML tree. As a result, each frame node starts with the default coordinate system. Each child node inherits the most recent coordinate system of its parent. A transformation node modifies the most recent coordinate system of its parent. Coordinate specifications are always related to the most recent coordinate system of the parent node.

The specification defines transformation nodes for scaling and translation. The scene description contains placeholders where these transformation nodes are placed⁴.

```
<xs:complexType name="Transformation">
  <xs:sequence>
    <xs:element name="Translate" type="Vector" minOccurs="0"/>
    <xs:element name="Scale" type="Vector" minOccurs="0"/>
    ...
  </xs:sequence>
</xs:complexType>
```

It follows a mathematical description of coordinate systems and transformations. A coordinate system consists of a translational vector $t = \begin{pmatrix} t_x \\ t_y \end{pmatrix}$ and scaling $s = \begin{pmatrix} s_x \\ s_y \end{pmatrix}$. A point $p = \begin{pmatrix} p_x \\ p_y \end{pmatrix}$ given

with respect to this coordinate system is transformed into the corresponding point $q = \begin{pmatrix} q_x \\ q_y \end{pmatrix}$ of

the default coordinate system by the following formula: $\begin{pmatrix} q_x \\ q_y \end{pmatrix} = \begin{pmatrix} p_x \cdot s_x + t_x \\ p_y \cdot s_y + t_y \end{pmatrix}$. Similarly, a vector v given with respect to the coordinate system is transformed into the corresponding vector w of the default coordinate system by: $\begin{pmatrix} w_x \\ w_y \end{pmatrix} = \begin{pmatrix} v_x \cdot s_x \\ v_y \cdot s_y \end{pmatrix}$.

A transformation node has an optional scaling vector $u = \begin{pmatrix} u_x \\ u_y \end{pmatrix}$ and an optional translational vector $v = \begin{pmatrix} v_x \\ v_y \end{pmatrix}$. If the scaling is not specified, its default value $u = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$ is assumed. Similarly,

⁴ Note that the schema is included here for information purposes only. See Clause F.8 for the normative schema definition.

the default value for the translation is $v = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$. The transformation node modifies the top-most coordinate system in the following way:

$\begin{pmatrix} t'_x \\ t'_y \end{pmatrix} = \begin{pmatrix} v_x \cdot s_x + t_x \\ v_y \cdot s_y + t_y \end{pmatrix}$, $\begin{pmatrix} s'_x \\ s'_y \end{pmatrix} = \begin{pmatrix} u_x \cdot s_x \\ u_y \cdot s_y \end{pmatrix}$, where $\begin{pmatrix} t'_x \\ t'_y \end{pmatrix}$ and $\begin{pmatrix} s'_x \\ s'_y \end{pmatrix}$ replace the top-most coordinate system.

For example, the coordinates of the scene description are given in a frame coordinate system, where the lower-left corner has coordinates (0,0) and the upper-right corner coordinates (320,240). The frame node resembles the following code where the scaling is set to the doubled reciprocal of the frame width and the frame height:

```
<tt:Frame UtcTime="2008-10-10T12:24:57.321">
  <tt:Transformation>
    <tt:Translate x="-1.0" y="-1.0"/>
    <tt:Scale x="0.00625" y="0.00834"/>
  </tt:Transformation>
  ...
</tt:Frame>
```

9.2.3 Scene elements

9.2.3.1 General

Subclause 9.2.3 focuses on scene elements generated by object tracking algorithms and defines object handling and object shapes for them.

Frames where no objects have been detected can be skipped within the scene description to save bandwidth, as long as the last frame in the scene description is empty as well. It is recommended that the device regularly sends the scene description even if it is empty. This is in order to indicate that the analytics engine is operational. The device shall send a scene description if SynchronizationPoint is requested for the corresponding stream.

When the receiver of a scene description receives an empty frame, the receiver should assume that all subsequent frames are empty as well until the next non-empty frame is received. When the last received frame is non-empty, the receiver should assume that a description of the next processed frame will be transmitted.

9.2.3.2 Objects

Objects are identified via their ObjectID. Features relating to one particular object are collected in an object node with the corresponding ObjectID as an attribute. Associations of objects, such as Object Renaming, Object Splits, Object Merges and Object Deletions are expressed in a separate ObjectTree node. An ObjectID is implicitly created with the first appearance of the ObjectID within an object node⁵.

⁵ Note that the schema is included here for information purposes only. See Clause F.8 for the normative schema definition.

```

<xs:complexType name="ObjectId">
  <xs:attribute name="ObjectId" type="xs:integer"/>
</xs:complexType>

<xs:complexType name="Object">
  <xs:complexContent>
    <xs:extension base="ObjectId">
      <xs:sequence>
        <xs:element name="Appearance" type="Appearance" minOccurs="0"/>
        <xs:element name="Behaviour" type="Behaviour" minOccurs="0"/>
        ...
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

The object node has two placeholders for appearance and behaviour information. The appearance node starts with an optional transformation node that can be used to change from a frame-centric coordinate system to an object-centric coordinate system. Next, the shape of an object can be specified. If an object is detected in a frame, the shape information should be present in the appearance description. The analytics algorithm may add object nodes for currently not visible objects, if it is able to infer information for this object otherwise. In such cases, the shape description may be omitted.

Object features such as shape (see 9.2.3.4), colour (see 9.2.3.5) and object class (see 9.2.3.6) can be added to the appearance node.

This specification defines two standard behaviours for objects: Removed or Idle. These behaviours shall be listed as child nodes of the behaviour node of an object. The presence of a removed or idle node does not automatically delete the corresponding ObjectId, making it possible to reuse the same ObjectId when the object starts moving again.

An object marked with the removed behaviour specifies the place from where the real object was removed. The marker should not be used as the behaviour of the removed object. It is possible to detect the removal even if the action of taking away the object was not detected.

Objects previously in motion can be marked as Idle to indicate that the object stopped moving. As long as such objects don't change, they will not be listed in the scene description anymore. When an Idle object appears again in the scene description, the Idle flag is removed automatically.

Example:

```

...
<tt:Frame UtcTime="2008-10-10T12:24:57.321">
  <tt:Transformation>
    <tt:Translate x="-1.0" y="-1.0"/>
    <tt:Scale x="0.003125" y="0.00416667"/>
  </tt:Transformation>
  <tt:Object ObjectId="12">
    <tt:Appearance>
      <tt:Shape>
        <tt:BoundingBox left="20.0" top="30.0" right="100.0" bottom="80.0"/>
        <tt:CenterOfGravity x="60.0" y="50.0"/>
      </tt:Shape>
    </tt:Appearance>
  </tt:Object>
</tt:Frame>
...
<tt:Frame UtcTime="2008-10-10T12:24:57.421">
  <tt:Transformation>
    <tt:Translate x="-1.0" y="-1.0"/>
    <tt:Scale x="0.003125" y="0.00416667"/>

```

```

</tt:Transformation>
<tt:Object ObjectId="12">
  <tt:Appearance>
    <tt:Shape>
      <tt:BoundingBox left="20.0" top="30.0" right="100.0" bottom="80.0"/>
      <tt:CenterOfGravity x="60.0" y="50.0"/>
    </tt:Shape>
  </tt:Appearance>
  <tt:Behaviour>
    <tt:Idle/>
  </tt:Behaviour>
</tt:Object>
</tt:Frame>

...

<tt:Frame UtcTime="2008-10-10T12:24:57.521">
  <tt:Transformation>
    <tt:Translate x="-1.0" y="-1.0"/>
    <tt:Scale x="0.003125" y="0.00416667"/>
  </tt:Transformation>
</tt:Frame>

...

<tt:Frame UtcTime="2008-10-10T12:24:57.621">
  <tt:Transformation>
    <tt:Translate x="-1.0" y="-1.0"/>
    <tt:Scale x="0.003125" y="0.00416667"/>
  </tt:Transformation>
  <tt:Object ObjectId="12">
    <tt:Appearance>
      <tt:Shape>
        <tt:BoundingBox left="25.0" top="30.0" right="105.0" bottom="80.0"/>
        <tt:CenterOfGravity x="65.0" y="50.0"/>
      </tt:Shape>
    </tt:Appearance>
  </tt:Object>
</tt:Frame>

...

<tt:Frame UtcTime="2008-10-10T12:24:57.721">
  <tt:Transformation>
    <tt:Translate x="-1.0" y="-1.0"/>
    <tt:Scale x="0.003125" y="0.00416667"/>
  </tt:Transformation>
  <tt:Object ObjectId="19">
    <tt:Appearance>
      <tt:Shape>
        <tt:BoundingBox left="20.0" top="30.0" right="100.0" bottom="80.0"/>
        <tt:CenterOfGravity x="60.0" y="50.0"/>
      </tt:Shape>
    </tt:Appearance>
    <tt:Behaviour>
      <tt:Removed/>
    </tt:Behaviour>
  </tt:Object>
</tt:Frame>

```

9.2.3.3 Object tree

When two objects come too close to each other, such that the analytics can no longer track them individually, an object merge should be signalled by adding a merge node to the ObjectTree node of the frame node. The merge node contains a From node listing the merging ObjectIds and a To node containing the ObjectId. The merged object is used in future frames as the tracking ID. If the analytics algorithm detects that one object is occluding the others and is able to track this object further, the occluding object should be put in the To node.

The separation of objects is indicated by a Split node. In this case, the From node contains a single ObjectId representing the object, which is split in the current frame. The objects separating from this split object are listed in the To node. The ObjectId of the From node can reappear in the To node, if this object did occlude the others and the analytics algorithm was able to track this object during the occlusion.

An object does not need to be involved in a merge operation in order to be part of a split operation. For example, if an object is moving together with a person, and the person leaves the object somewhere, the object might be detected the first time by the analytics when the person moves away from the object left behind. In such cases, the first appearance of the object can be combined with a split operation.

When a merged object reappears as an object node in a later frame without a split indication, then this object is implicitly split. The analytics algorithm, however, could not determine where the split object came from.

An analytics algorithm can track and remember a limited number of objects. In order to indicate that a certain object has been removed from the memory of the algorithm and therefore never appear again, the scene description can contain a Delete node within the ObjectTree node.

If the analytics algorithm cannot decide during a split operation the identity of an object, it should use a new ObjectId. When the algorithm has collected sufficient evidence for the identity of this object, it can change the ObjectId via the Rename operation. The Rename operation can also be used when an object re-enters the scene and the true identity is discovered after some time.

A deleted ObjectId shall not be reused within the scene description until the ObjectId container has wrapped around.

Example:

```
<tt:Frame UtcTime="2008-10-10T12:24:57.321">
  <tt:Object ObjectId="12">
    ...
  </tt:Object>
  <tt:Object ObjectId="17">
    ...
  </tt:Object>
</tt:Frame>
```

```
<tt:Frame UtcTime="2008-10-10T12:24:57.421">
  <tt:Object ObjectId="12">
    ...
  </tt:Object>
  <tt:ObjectTree>
    <tt:Merge>
      <tt:From ObjectId="12"/>
      <tt:From ObjectId="17"/>
      <tt:To ObjectId="12"/>
    </tt:Merge>
  </tt:ObjectTree>
</tt:Frame>
```

```
<tt:Frame UtcTime="2008-10-10T12:24:57.521">
  <tt:Object ObjectId="12">
    ...
  </tt:Object>
</tt:Frame>
```

```
<tt:Frame UtcTime="2008-10-10T12:24:57.621">
```

```
<tt:Object ObjectId="12">
...
</tt:Object>
<tt:Object ObjectId="17">
...
</tt:Object>
<tt:ObjectTree>
  <tt:Split>
    <tt:From ObjectId="12"/>
    <tt:To ObjectId="17"/>
    <tt:To ObjectId="12"/>
  </tt:Split>
</tt:ObjectTree>
</tt:Frame>
```

9.2.3.4 Shape descriptor

Shape information shall be placed below the optional shape node of in an object appearance node. If present, the shape node holds information where the object under consideration has been detected in the specified frame. A shape node shall at least contain two nodes representing the BoundingBox and the CenterOfGravity of the detected object.

The coarse BoundingBox is further refined with additional child nodes, each representing a shape primitive. If multiple shape primitives are present, their union defines the object's shape. In this specification, a generic polygon descriptor is provided.

Polygons that describe the shape of an object shall be simple polygons defined by a list of points.

Two consecutive points (where the last point is connected with the first one) in the list define a line segment. The order of the points shall be chosen such that the enclosed object region can be found on the left-hand side all line segments. The polyline defined by the list of points shall not be self-intersecting.

Example:

```
<tt:Frame UtcTime="2008-10-10T12:24:57.321">
  <tt:Transformation>
    <tt:Translate x="-1.0" y="-1".0/>
    <tt:Scale x="0.003125" y="0.00416667"/>
  </tt:Transformation>
  <tt:Object ObjectId="12">
    <tt:Appearance>
      <tt:Shape>
        <tt:BoundingBox left="20.0" top="30.0" right="100.0" bottom="80.0"/>
        <tt:CenterOfGravity x="60.0" y="50.0"/>
        <tt:Polygon>
          <tt:Point x="20.0" y="30.0"/>
          <tt:Point x="100.0" y="30.0"/>
          <tt:Point x="100.0" y="80.0"/>
          <tt:Point x="20.0" y="80.0"/>
        </tt:Polygon>
      </tt:Shape>
    </tt:Appearance>
  </tt:Object>
</tt:Frame>
```

9.2.3.5 Colour descriptor

A colour descriptor is defined as an optional element of the appearance node of an object node. The colour descriptor is defined by a list of colour clusters, each consisting of a colour value, an optional weight and an optional covariance matrix. The colour descriptor does not specify how the colour clusters are created. They can represent bins of a colour histogram or the result of a clustering algorithm.

Colours are represented by three-dimensional vectors. Additionally, the colour space of each colour vector can be specified by a colour space attribute. If the colour space attribute is missing, the YcbCr colour space is assumed. It refers to the 'sRGB' gamut with the RGB to YcbCr transformation of ISO/IEC 10918-1, also known as JPEG. The colour space URI for the YcbCr colour space is www.onvif.org/ver10/colourspace/YcbCr.

An example metadata containing colour information of the detected object is given below.

```
<tt:Frame UtcTime="2010-09-10T12:24:57.721">

  <tt:Object ObjectId="34">
    <tt:Appearance>
      <tt:Shape>
        <tt:BoundingBox left="20.0" top="30.0" right="100.0" bottom="80.0"/>
        <tt:CenterOfGravity x="60.0" y="50.0"/>
      </tt:Shape>
      <tt:Color>
        <tt:ColorCluster>
          <tt:Color X="58" Y="105" Z="212"/>
          <tt:Covariance XX="7.2" YY="6" ZZ="3"/>
          <tt:Weight>90</tt:Weight>
        </tt:ColorCluster>
        <tt:ColorCluster>
          <tt:Color X="165" Y="44" Z="139"/>
          <tt:Covariance XX="4" YY="4" ZZ="4"/>
          <tt:Weight>5</tt:Weight>
        </tt:ColorCluster>
      </tt:Color>
    </tt:Appearance>
  </tt:Object>

</tt:Frame>
```

Colour Descriptor contains the representative colours in detected the object/region. The representative colours are computed from the image of the detected object/region each time. Each representative colour value is a vector of the specified colour space. The representative colour weight (Weight) denotes the fraction of pixels assigned to the representative colour. Colour covariance describes the variation of colour values around the representative colour value in the colour space. Thus, the representative colour denotes a region in the colour space. Table 2 lists the acceptable values for the Colourspace attribute

Table 2 – Colourspace namespace values

Namespace URI	Description
http://www.onvif.org/ver10/colourspace/YcbCr	YcbCr colour space
http://www.onvif.org/ver10/colourspace/CIELUV	CIE LUV
http://www.onvif.org/ver10/colourspace/CIELAB	CIE 1976 (L*a*b*)
http://www.onvif.org/ver10/colourspace/HSV	HSV colour space

9.2.3.6 Object class descriptor

A class descriptor is defined as an optional element of the appearance node of an object node. The Class Descriptor is defined by a list of object classes together with a likelihood that the corresponding object belongs to this class. The sum of the likelihoods shall not exceed 1.

The following example shows an object metadata sample that contains object class information about a detected object:

```
<tt:Frame UtcTime="2010-11-10T12:24:57.721">

<tt:Object ObjectId="22">
  <tt:Appearance>
    <tt:Shape>
      <tt:BoundingBox left="20.0" top="30.0" right="100.0" bottom="80.0"/>
      <tt:CenterOfGravity x="60.0" y="50.0"/>
    </tt:Shape>
    <tt:Class>
      <tt:ClassCandidate>
        <tt:Type>Vehicle</tt:Type>
        <tt:Likelihood>0.93</tt:Likelihood >
      </tt:ClassCandidate>
    </tt:Class>
  </tt:Appearance>
</tt:Object>

</tt:Frame>
```

The new object class labels are introduced through the extensibility mechanism in the original schema by revising the ClassDescriptorExtension data type. The following is an example that introduces a new class type and uses in object metadata.

```
<tt:Frame UtcTime="2011-08-11T12:44:57.721">
  <tt:Object ObjectId="122">
    <tt:Appearance>
      <tt:Shape>
        <tt:BoundingBox left="20.0" top="30.0" right="80.0" bottom="40.0"/>
        <tt:CenterOfGravity x="50.0" y="45.0"/>
      </tt:Shape>
      <tt:Class>
        <tt:ClassCandidate>
          <tt:Type>Other</tt:Type>
          <tt:Likelihood>0.81</tt:Likelihood >
        </tt:ClassCandidate>
        <tt:Extension>
          <tt:OtherTypes>
            <tt:Type>LicensePlate</tt:Type>
            <tt:Likelihood>0.81</tt:Likelihood >
          </tt:OtherTypes>
        </tt:Extension>
      </tt:Class>
    </tt:Appearance>
  </tt:Object>
</tt:Frame>
```

The older implementations can ignore the Extension element. New implementations can parse and use the data in the Extension element. This approach keeps the ClassType data type definition while expanding the meaning when the value is "Other". The specification shall list the acceptable values for OtherType/Type element, then the implementations can validate the acceptable values.

9.2.3.7 Motion in cells descriptor

The scene description of a cell motion contains the cells where motion is detected. To decode base64Binary, the columns and rows of the cell grid are provided.

For spatial relation, the "Transformation" element is used, see 9.2.2.3. The cell has attributes as defined in Table 3. Its grid starts in the upper left-hand corner. The *X* dimension goes from left to right and the *Y* dimension goes from top to bottom, see Figure 9.

This example contains a snippet of a metadata stream using Cell Motion Detector.

```
<?xml version="1.0" encoding="UTF-8"?>
<tt:MetaDataStream
  xmlns:tt="http://www.onvif.org/ver10/schema">
```

```

<tt:VideoAnalytics>
  <tt:Frame UtcTime="2010-10-20T12:24:57.321">
    <tt:Transformation>
      <tt:Translate x="-0.66666" y="-0.6"/>
      <tt:Scale x="0.1666666" y="-0.2"/>
    </tt:Transformation>
    <tt:Extension>
      <tt:MotionInCells Columns="8" Rows="6" Cells="AAD8AA==" />
    </tt:Extension>
  </tt:Frame>
  <tt:Frame UtcTime="2010-10-20T12:24:57.621">
    ...
  </tt:Frame>
</tt:VideoAnalytics>

<tt:Event>
  <wsnt:NotificationMessage>
    <wsnt:Topic Dialect="...Concrete">
      tns1:RuleEngine/CellMotionDetector/Motion
    </wsnt:Topic>
    <wsnt:Message>
      <tt:Message UtcTime= "2010-10-20T12:24:57.628">
        <tt:Source>
          <tt:SimpleItem Name="VideoSourceConfigurationToken" Value="1"/>
          <tt:SimpleItem Name="VideoAnalyticsConfigurationToken" Value="1"/>
          <tt:SimpleItem Name="Rule" Value="MotionInDefinedCells"/>
        </tt:Source>
        <tt:Data>
          <tt:SimpleItem Name="IsMotion" Value="true"/>
        </tt:Data>
      </tt:Message>
    </wsnt:Message>
  </wsnt:NotificationMessage>
</tt:Event>

</tt:MetaDataStream>

```

Table 3 – Description of attributes of MotionInCells type

Attribute	Description
Columns	Number of columns of the cell grid (<i>X</i> dimension)
Rows	Number of rows of the cell grid (<i>Y</i> dimension).
Cells	A "1" denotes a cell where motion is detected and a "0" denotes an empty cell. The first cell is in the upper left corner. Then the cell order goes first from left to right and then from top to bottom If the number of cells is not a multiple of 8, the last byte is filled with zeros. The information is run-length encoded in accordance with Packbit coding in ISO 12639 (TIFF, Revision 6.0).

9.3 Rule interface

9.3.1 General

An XML structure is introduced in 9.3.2 to communicate the configuration of rules. Subclause 9.3.3 specifies a language to describe the configuration of a specific rule type. A device implementing a rule engine can support rules described in Appendix A. Subclause 9.3.4 introduces operations to manage rules. If the device supports a Rule Engine, it shall implement the complete rule interface.

9.3.2 Rule representation

The configuration of a rule has two required attributes: one specifies the name and the other specifies the type of the rule. The different configuration parameters are listed below the parameters element of the rule element. Each parameter is either a SimpleItem or an ElementItem. The name attribute of each item shall be unique within the parameter list. SimpleItems have an additional Value attribute containing the value of the parameter. The value of ElementItems is given by the child element of the ElementItem. It is recommended to represent as many parameters as possible with SimpleItems.

The following example shows a complete analytics configuration containing two rules:

```
<tt:RuleEngineConfiguration>
  <tt:Rule Name="MyLineDetector" Type="tt:LineDetector">
    <tt:Parameters>
      <tt:SimpleItem Name="Direction" Value="Any"/>
      <tt:ElementItem Name="Segments">
        <tt:Polyline>
          <tt:Point x="10.0" y="50.0"/>
          <tt:Point x="100.0" y="50.0"/>
        </tt:Polyline>
      </tt:ElementItem>
    </tt:Parameters>
  </tt:Rule>
  <tt:Rule Name="MyFieldDetector" Type="tt:FieldDetector">
    <tt:Parameters>
      <tt:ElementItem Name="Field">
        <tt:Polygon>
          <tt:Point x="10.0" y="50.0"/>
          <tt:Point x="100.0" y="50.0"/>
          <tt:Point x="100.0" y="150.0"/>
        </tt:Polygon>
      </tt:ElementItem>
    </tt:Parameters>
  </tt:Rule>
</tt:RuleEngineConfiguration>
```

9.3.3 Rule description language

The description of a rule contains the type information of all parameters belonging to a certain rule type and the description of the output produced by such a rule. The output of the Rule Engine are events that can either be used in an Event Engine or be subscribed to by a client.

Below, the definitions are included for convenience⁶:

```
<xs:complexType name="ConfigDescription">
  <xs:sequence>
    <xs:element name="Parameters"
      type="tt:ItemListDescription"/>
    <xs:element name="Messages" minOccurs="0" maxOccurs="unbounded">
      <xs:complexType>
        <xs:complexContent>
          <xs:extension base="tt:MessageDescription">
            <xs:sequence>
              <xs:element name="ParentTopic" type="xs:string"/>
            </xs:sequence>
          </xs:extension>
        </xs:complexContent>
      </xs:complexType>
    </xs:element>
```

⁶ Please note that the schema is included here for information purposes only. See Clause F.8 for the normative schema definition.

```

    ...
  </xs:sequence>
  <xs:attribute name="Name" type="xs:QName" use="required"/>
</xs:complexType>

<xs:complexType name="ItemListDescription">
  <xs:sequence>
    <xs:element name="SimpleItemDescription" minOccurs="0"
      maxOccurs="unbounded">
      <xs:complexType>
        <xs:attribute name="Name" type="xs:string" use="required"/>
        <xs:attribute name="Type" type="xs:QName" use="required"/>
      </xs:complexType>
    </xs:element>
    <xs:element name="ElementItemDescription" minOccurs="0"
      maxOccurs="unbounded">
      <xs:complexType>
        <xs:attribute name="Name" type="xs:string" use="required"/>
        <xs:attribute name="Type" type="xs:QName" use="required"/>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>

```

The parameters of a certain rule type are listed below the Parameters element. All parameters are either SimpleItems or ElementItems and can be described by either SimpleItemDescription or ElementItemDescription. Both ItemDescriptions contain a name attribute to identify the parameter and a type attribute to reference a specific XML schema type. In the case of the SimpleItemDescription, the type attribute shall reference a SimpleType schema definition. In the case of the ElementItemDescription, the type attribute shall reference a global element declaration of an XML schema.

The output produced by this rule type is described in multiple Messages elements. Each Messages contains a description of the message payload according to the Message Description Language detailed in the ONVIF Core Specification. Additionally, the Messages shall contain a ParentTopic element naming the Topic a client has to subscribe to in order to receive this specific output. The topic shall be specified as a Concrete Topic Expression.

9.3.4 Operations on rules

9.3.4.1 General

If the device supports a Rule Engine as defined by ONVIF, then it shall implement the following operations to manage rules. The Create/Delete/Modify operations are atomic, meaning that either all modifications can be processed, or the complete operation shall fail.

9.3.4.2 GetSupportedRules

The device shall indicate the rules it supports by implementing the subsequent operation. It returns a list of rule descriptions according to the Rule Description Language described in 9.3.3. Additionally, it contains a list of URLs that provide the location of the schema files. These schema files describe the types and elements used in the rule descriptions. Rule descriptions that reference types or elements from any ONVIF defined schema files need not explicitly list those schema files.

REQUEST:

- **ConfigurationToken [tt:ReferenceToken]**

Token of the analytics configuration for which the supported rules should be listed.

RESPONSE:

- **SupportedRules [tt:SupportedRules]**

- **Rules supported for the analytics configuration**

FAULTS:

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

ACCESS CLASS:

READ_MEDIA

9.3.4.3 Get Rules

The following operation retrieves the currently installed rules:

REQUEST:

- **ConfigurationToken [tt:ReferenceToken]**
Token of the analytics configuration for which the rules should be listed.

RESPONSE:

- **Rule – optional, unbounded [tt:Config]**
List of installed rules for the specified configuration.

FAULTS:

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

ACCESS CLASS:

READ_MEDIA

9.3.4.4 CreateRules

The following operation adds rules to an AnalyticsConfiguration. If all rules cannot be created as requested, the device responds with a fault message.

REQUEST:

- **ConfigurationToken [tt:ReferenceToken]**
Token of the analytics configuration for which the rules should be installed.
- **Rule – unbounded [tt:Config]**
List of rules to be installed for the specified configuration.

RESPONSE:

This is an empty message.

FAULTS:

- **env:Sender – ter:InvalidArgVal – ter:NoConfig**
The requested analytics configuration does not exist.
- **env:Sender – ter:InvalidArgVal – ter:InvalidRule**
The suggested rule configuration is not valid on the device.
- **env:Sender – ter:InvalidArgVal – ter:RuleAlreadyExistent**

A rule with the same name already exists in the configuration.

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

There is not enough space in the device to add the rules to the configuration.

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

The device cannot create the rules without creating a conflicting configuration.

ACCESS CLASS:

ACTUATE

9.3.4.5 ModifyRules

The following operation modifies multiple rules. If all rules cannot be modified as requested, the device responds with a fault message.

REQUEST:

- **ConfigurationToken [tt:ReferenceToken]**

Token of the analytics configuration for which the rules should be modified.

- **Rule – unbounded [tt:Config]**

List of rules to be modified for the specified configuration.

RESPONSE:

- **This is an empty message.**

FAULTS:

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

The requested analytics configuration does not exist.

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

The suggested rule configuration is not valid on the device.

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

The rule name or names do not exist.

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

The device cannot modify the rules without creating a conflicting configuration.

ACCESS CLASS:

ACTUATE

9.3.4.6 DeleteRules

The following operation deletes multiple rules. If all rules cannot be deleted as requested, the device responds with a fault message.

REQUEST:

- **ConfigurationToken [tt:ReferenceToken]**

Token of the analytics configuration for which the rules should be removed.

- **RuleName – unbounded [xs:string]**

List of rules to be removed for the specified configuration.

RESPONSE:

This is an empty message.

FAULTS:

- **env:Sender – ter:InvalidArgVal – ter:NoConfig**
The requested analytics configuration does not exist.
- **env:Sender – ter:InvalidArgVal – ter:RuleNotExistent**
The rule name or names do not exist.
- **env:Sender – ter:Action – ter:FixedRules**
At least one of the requested rules is fixed and cannot be deleted.
- **env:Receiver – ter:Action – ter:ConfigurationConflict**
The device cannot delete the rules without creating a conflicting configuration.

ACCESS CLASS:

ACTUATE

9.3.4.7 GetRuleOptions

The following operation returns the options for the supported rules that specify an Option attribute.

REQUEST:

- **RuleType – optional [xs:QName]**
Optional rule type for which the options shall be retrieved. If omitted all rule option types with all rule options supported by the device shall be returned (as appropriate).
- **ConfigurationToken [tt:ReferenceToken]**
Token of the analytics configuration for which the supported rules should be listed.

RESPONSE:

- **RuleOptions – optional, unbounded [tan:ConfigOptions]**
Rules supported for the analytics configuration

FAULTS:

- **env:Sender – ter:InvalidArgVal – ter:NoConfig**
The requested analytics configuration does not exist.
- **env:Sender – ter:InvalidArgVal – ter:RuleNotExistent**
The rule option type does not exist.

ACCESS CLASS:

READ_MEDIA

9.4 Analytics modules interface

9.4.1 General

Subclause 9.4.2 defines an XML structure that communicates the configuration of analytics modules. Subclause 9.4.3 defines the language that describes the configuration of a specific analytics module. Subclause 9.4.4 defines the operations required by the analytics modules

interface. If the device supports an analytics engine as defined by ONVIF, it shall implement the complete analytics modules interface.

9.4.2 Analytics module configuration

The analytics module configuration is identical to the rule configuration described in 9.3.2. The following example shows a possible configuration of a vendor-specific ObjectTracker. This tracker allows configuration of the minimum and maximum object size with respect to the processed frame geometry.

```
<tt:AnalyticsEngineConfig>
  <tt:AnalyticsModule Name="MyObjectTracker" Type="nn:ObjectTracker">
    <tt:Parameters>
      <tt:SimpleItem Name="MinObjectWidth" Value="0.01"/>
      <tt:SimpleItem Name="MinObjectHeight" Value="0.01"/>
      <tt:SimpleItem Name="MaxObjectWidth" Value="0.5"/>
      <tt:SimpleItem Name="MaxObjectHeight" Value="0.5"/>
    </tt:Parameters>
  </tt:AnalyticsModule>
</tt:AnalyticsEngineConfig>
```

9.4.3 Analytics module description language

The analytics module reuses the rule description language, described in 9.3.3. The following AnalyticsModuleDescription element replaces the RuleDescription element:

```
<xs:element name="AnalyticsModuleDescription"
  type="tt:ConfigDescription"/>
```

Similar to rules, analytics modules produce events and shall be listed within the analytics module description. The subsequent description corresponds to the example of 9.4.2. The example module produces a SceneTooCrowded event when the scene becomes too complex for the module.

```
<tt:AnalyticsModuleDescription Name="nn:ObjectTracker">
  <tt:Parameters>
    <tt:SimpleItemDescription Name="MinObjectWidth" Type="xs:float"/>
    <tt:SimpleItemDescription Name="MinObjectHeight" Type="xs:float"/>
    <tt:SimpleItemDescription Name="MaxObjectWidth" Type="xs:float"/>
    <tt:SimpleItemDescription Name="MaxObjectHeight" Type="xs:float"/>
  </tt:Parameters>
  <tt:Messages>
    <tt:Source>
      <tt:SimpleItemDescription Name="VideoSourceConfiguration"
        Type="tt:ReferenceToken"/>
      <tt:SimpleItemDescription Name="AnalyticsConfiguration"
        Type="tt:ReferenceToken"/>
      <tt:SimpleItemDescription Name="AnalyticsModule" Type="xs:string"/>
    </tt:Source>
    <tt:ParentTopic>
      tns1:VideoAnalytics/nn:ObjectTracker/SceneTooCrowded
    </tt:ParentTopic>
  </tt:Messages>
</tt:RuleDescription>
```

9.4.4 Operations on analytics modules

9.4.4.1 General

If the device supports an analytics engine as defined by ONVIF, it shall support the subsequent operations to manage analytics modules. The Create/Delete/Modify operations shall be atomic; all modifications can be processed, or the complete operation shall fail.

9.4.4.2 GetSupportedAnalyticsModules

The device indicates the analytics modules it supports by implementing the GetSupportedAnalyticsModule operation. It returns a list of analytics modules according to the analytics module description language, described in 9.3.3. Additionally, it contains a list of URLs that provide the location of the schema files. These schema files describe the types and elements used in the analytics module descriptions. Analytics module descriptions that reference types or elements imported from any ONVIF defined schema files need not explicitly list those schema files.

REQUEST:

- **ConfigurationToken [tt:ReferenceToken]**

Token of the analytics configuration for which the supported modules should be listed.

RESPONSE:

- **SupportedAnalyticsModules[tt:SupportedModules]**

Analytics modules supported for the analytics configuration

FAULTS:

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

The requested analytics configuration does not exist.

ACCESS CLASS:

READ_MEDIA

9.4.4.3 GetAnalyticsModules

The following operation retrieves the currently installed analytics modules.

REQUEST:

- **ConfigurationToken [tt:ReferenceToken]**

Token of the analytics configuration for which the modules should be listed.

RESPONSE:

- **AnalyticsModule – optional, unbounded [tt:Config]**

List of installed modules for the specified configuration.

FAULTS:

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

The requested analytics configuration does not exist.

ACCESS CLASS:

READ_MEDIA

9.4.4.4 CreateAnalytics Modules

The following operation adds analytics modules to an AnalyticsConfiguration. If all analytics modules cannot be created as requested, the device responds with a fault message.

REQUEST:

- **ConfigurationToken [tt:ReferenceToken]**
Token of the analytics configuration for which the modules should be installed.
- **AnalyticsModule – unbounded [tt:Config]**
List of modules to be installed for the specified configuration.

RESPONSE:

This is an empty message.

FAULTS:

- **env:Sender – ter:InvalidArgVal – ter:NoConfig**
The requested analytics configuration does not exist.
- **env:Sender – ter:InvalidArgVal – ter:InvalidModule**
The suggested module configuration is not valid on the device.
- **env:Sender – ter:InvalidArgVal – ter:NameAlreadyExistent**
The same analytics module name exists already in the configuration.
- **env:Receiver – ter:Action – ter:TooManyModules**
There are not enough resources in the device to add the analytics modules to the configuration.
- **env:Receiver – ter:Action – ter:ConfigurationConflict**
The device cannot create the analytics modules without creating a conflicting configuration.

ACCESS CLASS:

ACTUATE**9.4.4.5 ModifyAnalyticsModules**

The following operation modifies multiple analytics modules. If all analytics modules cannot be modified as requested, the device responds with a fault message.

REQUEST:

- **ConfigurationToken [tt:ReferenceToken]**
Token of the analytics configuration for which the modules should be modified.
- **AnalyticsModule – unbounded [tt:Config]**
List of modules to be modified for the specified configuration.

RESPONSE:

This is an empty message.

FAULTS:

- **env:Sender – ter:InvalidArgVal – ter:NoConfig**
The requested analytics configuration does not exist.
- **env:Sender – ter:InvalidArgVal – ter:InvalidModule**
The suggested module configuration is not valid on the device.
- **env:Sender – ter:InvalidArgVal – ter:NameNotExistent**

The module name or names do not exist.

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

The device cannot modify the modules without creating a conflicting configuration.

ACCESS CLASS:

ACTUATE

9.4.4.6 DeleteAnalyticsModules

The following operation deletes multiple analytics modules. If all analytics modules cannot be deleted as requested, the device responds with a fault message.

REQUEST:

- **ConfigurationToken [tt:ReferenceToken]**
Token of the analytics configuration for which the modules should be removed.
- **AnalyticsModuleName – unbounded [xs:string]**
List of modules to be removed for the specified configuration.

RESPONSE:

This is an empty message.

FAULTS:

- **envx:Sender – ter:InvalidArgVal – ter:NoConfig**
The requested analytics configuration does not exist.
- **env:Sender – ter:InvalidArgVal – ter:InvalidModule**
The suggested module configuration is not valid on the device
- **env:Sender – ter:InvalidArgVal – ter:NameNotExistent**
The module name or names do not exist.
- **env:Sender – ter:Action – ter:FixedModules**
At least one of the requested modules is fixed which cannot be deleted.
- **env:Receiver – ter:Action – ter:ConfigurationConflict**
The device cannot delete the modules without creating a conflicting configuration.

9.5 GetAnalyticsModuleOptions

The following operation returns the options for the supported analytics modules that specify an Option attribute.

REQUEST:

- **AnalyticsModuleType – optional [xs:QName]**
Optional module type for which the options shall be retrieved. If omitted, all module option types with all module options supported by the device shall be returned (as appropriate).
- **ConfigurationToken [tt:ReferenceToken]**
Token of the analytics configuration for which the supported modules should be listed.

RESPONSE:

- **Options – optional, unbounded [tan:AnalyticsModuleConfigOptions]**

List of options for the specified analytics module.

FAULTS:

- **env:Sender – ter:InvalidArgVal – ter:NoConfig**
The requested analytics configuration does not exist.
- **env:Sender – ter:InvalidArgVal – ter:TypeNotExistent**
The module option type does not exist.

ACCESS CLASS:

READ_MEDIA

9.6 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:

RuleSupport:	Indication that the device supports rules interface and rules syntax as specified in 9.3.4.
AnalyticsModuleSupport:	Indication that the device supports the scene analytics module interface as specified in 9.4.
CellBasedSceneDescriptionSupported:	Indication that the device produces the cell-based scene description.
RuleOptionsSupported:	Indication that the device supports the GetRuleOptions operation on the rules interface.
AnalyticsModuleOptionsSupported:	Indication that the device supports the GetAnalyticsModuleOptions operation on the analytics module interface.

REQUEST:

This message is empty

RESPONSE:

- **Capabilities [tan:Capabilities]**
Set of indicators for function groups as described above.

FAULTS:

None

ACCESS CLASS:

PRE_AUTH

9.7 Events – Audio Detected

The analytics engine can send the following event for analyzed audio data.

```
<tt:MessageDescription IsProperty="false">
  <tt:Source>
    <tt:SimpleItemDescription Name="AudioSourceConfigurationToken"
      Type="tt:ReferenceToken"/>
    <tt:SimpleItemDescription Name="AudioAnalyticsConfigurationToken"
      Type="tt:ReferenceToken"/>
    <tt:SimpleItemDescription Name="Rule"
      Type="xs:string"/>
  </tt:Source>
  <tt:Data>
```

```

<tt:SimpleItemDescription Name="isSoundDetected" Type="xs:boolean"/>
<tt:SimpleItemDescription Name="UTCTime" Type="xs:dateTime"/>
<tt:ElementItemDescription Name="AudioClassifications"
Type="AudioClassDescriptor"/>
</tt:Data >
  <tt:ParentTopic>tns1:AudioAnalytics/Audio/DetectedSound
</tt:ParentTopic>
</tt:MessageDescription>

```

The Rule field is optional and contains the Rule instance name that generated the audio detected event.

10 Real-time streaming

10.1 General

This clause describes real-time streaming of video, audio and metadata. There is no specific service associated with the real-time streaming. The real-time configurations via web service commands are defined in the media service and ReceiverService.

10.2 Media stream protocol

10.2.1 Transport format

10.2.1.1 General

Real-time Transport Protocol (RTP) is a media transfer protocol (see 10.2.1.6). Subclauses 10.2.1.2 to 10.2.1.6 describe RTP data transfer.

10.2.1.2 RTP data transfer via UDP

UDP has the smallest overhead and is able to transfer real-time data in an efficient manner. A device shall support the RTP/UDP protocol and the device should support RTP/UDP multicasting.

10.2.1.3 RTP/TCP

This optional mode is deprecated owing to ambiguities in the interpretation of the respective RFCs. RTP/TCP protocol is defined in RFC 4571 and RFC 4572.

10.2.1.4 RTP/RTSP/TCP

The device should support media transfer using RTP/RTSP to traverse a firewall using an RTSP tunnel. This protocol shall conform to 10.12 of RFC 2326.

10.2.1.5 RTP/RTSP/HTTP/TCP

The data stream shall be sent via HTTP to traverse a firewall. A device shall support media transfer using RTP/RTSP/HTTP/TCP. And if a device supports TLS, the data stream shall be sent or received via HTTPS to traverse a firewall, and a device shall support media transfer using RTP/RTSP/HTTPS/TCP.

This protocol shall conform to 10.12 of RFC 2326 (RTSP: Embedded [Interleaved] Binary Data).

This tunnelling method shall also conform to QuickTime available from Apple Inc⁷. The mandatory parts of Apple Tunneling QuickTime RTSP and RTP over HTTP shall be implemented by a device.

10.2.1.6 RTP/RTSP/TCP/WebSocket

The device indicating support for RTSPOverWebSocket as explained in 6.12 shall support streaming media using the WebSocket protocol according to this specification. A device shall use the same port of the HTTP for WebSocket protocol for a normal connection. If a device supports TLS, then it shall support tunnelling WebSocket connections over TLS (see RFC 2818) and the port number shall be same as the HTTPS port number. For RTSP tunnelling over WebSocket, a device shall support RTP/RTSP/TCP interleaved binary data as defined in 10.12 of RFC 2326.

WebSocket protocol implementation shall conform to RFC 6455.

The mechanism to be used for establishing a WebSocket connection between a client and a server is explained in detail in Clause 7.

10.2.2 Media transport

10.2.2.1 RTP

10.2.2.1.1 General

The Real-time Transport Protocol provides real-time transfer for media streams between two end points. The RTP protocol provides support for re-ordering, de-jittering and media synchronization.

All media streams transferred by the RTP protocol shall conform to RFC 3550, RFC 3551, RFC 3984, RFC 7798, RFC 3016 or RFC 3640, and JPEG over RTP (see 10.2.3). Figure 10 shows the layout of the RTP protocol header.

0										1										2										3									
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1								
V		P	X	CC			M	PT				sequence number																											
timestamp																																							
synchronization source (SSRC) identifier																																							

Figure 10 – RTP header

An RTP header shall be filled up with values in Table 4.

⁷ 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.

Table 4 – RTP header value

Header field	Value	Description
Version (V): 2 bits	2	
Padding (P): 1 bit	0/1	If the payload includes padding octet, this should be set to "1"
Extension (X): 1 bit	0/1	If the header extension is used the Extension bit shall be set. E.g. 10.2.3 JPEG over RTP uses this extension.
CSRC count (CC): 4 bits	0	
Marker (M): 1 bit	0/1	The usage shall conform to related RFCs (e.g. RFC 3984 for H.264 Video) or to this document e.g. "JPEG over RTP" (see 10.2.3) or RTP streaming of metadata (see 10.2.2.1.2).
Payload type (PT): 7 bits	See RFC 3551, Clause 6.	
Sequence Number: 16 bits		The initial value of the "sequence number" should be random (unpredictable) to make known-plaintext attacks on encryption more difficult. This number increments by one for each RTP data packet sent
Timestamp: 32 bits		The initial value of the "timestamp" should be random (unpredictable) to make known-plaintext attacks on encryption more difficult. See 10.2.2.2.2 for further details of media synchronization. The usage of the timestamp is dependent on the codec.
SSRC 32 bits		The synchronization source for the data stream. This specification makes no restrictions on the use of this field.

10.2.2.1.2 RTP for metadata stream

Metadata streams are also transported by RTP. The usage of payload type, marker and timestamp for RTP header for the metadata stream is defined in the following way:

- A dynamic payload type (96-127) shall be used for payload type, which is assigned in the process of a RTSP session setup.
- The RTP marker bit shall be set to "1" when the XML document is closed.
- It is recommended to use an RTP timestamp representing the creation time of the RTP packet with a RTP clock rate of 90 000 Hz. Only UTC timestamps shall be used within the metadata stream. The synchronization of video and audio data streams is done using RTCP.

The Metadata payload is an XML document with root node `tt:MetaDataStream`. There is no limitation on the size of the XML document. If GZIP compression is used, the payload starts with a GZIP header in accordance with RFC 1952 followed by the compressed data. A marker bit signals the end of the compressed data. When a synchronization point (see "Synchronization Points" of the ONVIF Media Service Specification) is requested for the

stream, the previous XML document shall be closed and a new one started. It is recommended to start new XML documents after 1 s at the latest. The RTP timestamp of the metadata stream has no specific meaning. The metadata stream multiplexes metadata from different sources. This specification defines placeholders for the scene description of the video analytics, the PTZ status of the PTZ controller and the notifications of the event configuration. A device can select which of these parts should be multiplexed into the metadata during the media configuration (see "Metadata Configuration" of the ONVIF Media Service Specification). Each part can appear multiple times in an arbitrary order within the document. A metadata connection can be bi-directional using the backchannel mechanism (see 10.4).

A metadata stream contains the following elements:

- VideoAnalyticsStream;
- PTZStream;
- EventStream.

The place-holders for the different metadata sources have the following XML structure:

```
<xs:complexType name="VideoAnalyticsStream">
  <xs:choice minOccurs="0" maxOccurs="unbounded">
    <xs:element name="Frame" type="tt:Frame"/>
    ...
  </xs:choice>
</xs:complexType>

<xs:complexType name="PTZStream">
  <xs:choice minOccurs="0" maxOccurs="unbounded">
    <xs:element name="PTZStatus"/>
    ...
  </xs:choice>
</xs:complexType>

<xs:complexType name="EventStream">
  <xs:choice minOccurs="0" maxOccurs="unbounded">
    <xs:element ref="wsnt:NotificationMessage"/>
    ...
  </xs:choice>
</xs:complexType>
```

The following is an example of a metadata XML document:

```
<?xml version="1.0" encoding="UTF-8"?>
<tt:MetadataStream xmlns:tt="http://www.onvif.org/ver10/schema">
  <tt:VideoAnalytics>
    <tt:Frame UtcTime="2008-10-10T12:24:57.321">
      ...
    </tt:Frame>
    <tt:Frame UtcTime="2008-10-10T12:24:57.621">
      ...
    </tt:Frame>
  </tt:VideoAnalytics>
</tt:MetadataStream>

<?xml version="1.0" encoding="UTF-8"?>
<tt:MetadataStream xmlns:tt="http://www.onvif.org/ver10/schema">
  <tt:Event>
    <wsnt:NotificationMessage>
      <wsnt:Message>
        <tt:Message UtcTime="2008-10-10T12:24:57.628">
          ...
        </tt:Message>
      </wsnt:Message>
    </wsnt:NotificationMessage>
```

```
</tt:Event>
</tt:MetadataStream>
```

10.2.2.2 RTCP

10.2.2.2.1 General

The RTP control protocol provides feedback on the quality of service being provided by RTP and the synchronization of different media streams. The RTCP protocol shall conform to RFC 3550. Figure 11 shows the information flow for both sender reports (SR) and receiver reports (RR).

For a feedback request, RFC 4585 and RFC 5104 should be supported.

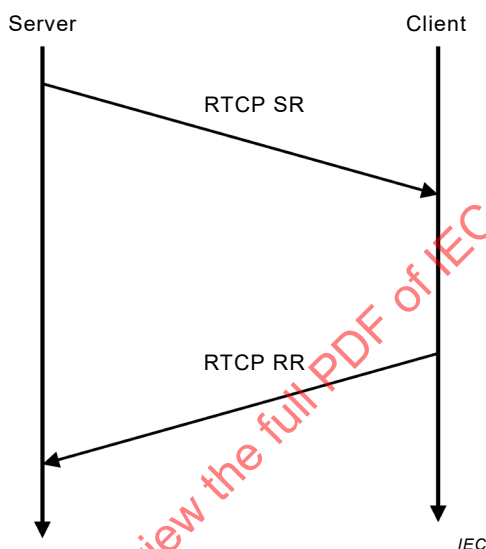


Figure 11 – RTCP sequence

10.2.2.2.2 Media synchronization

A client may receive audio and video streams simultaneously from more than one device. In this case, each stream uses a different clock (from data acquisition to packet receiving). RTCP Sender Reports (SR) are used to synchronize different media streams. RTCP SRs shall conform to RFC 3550.

The RTCP Sender Report (SR) packet has fields for the RTP timestamp and for a wall clock timestamp (absolute date and time, 64-bit NTP [Network Time Protocol]). See Figure 12.

A device shall support RTCP Sender Report for media synchronization. The client should use RTCP for the media synchronization.

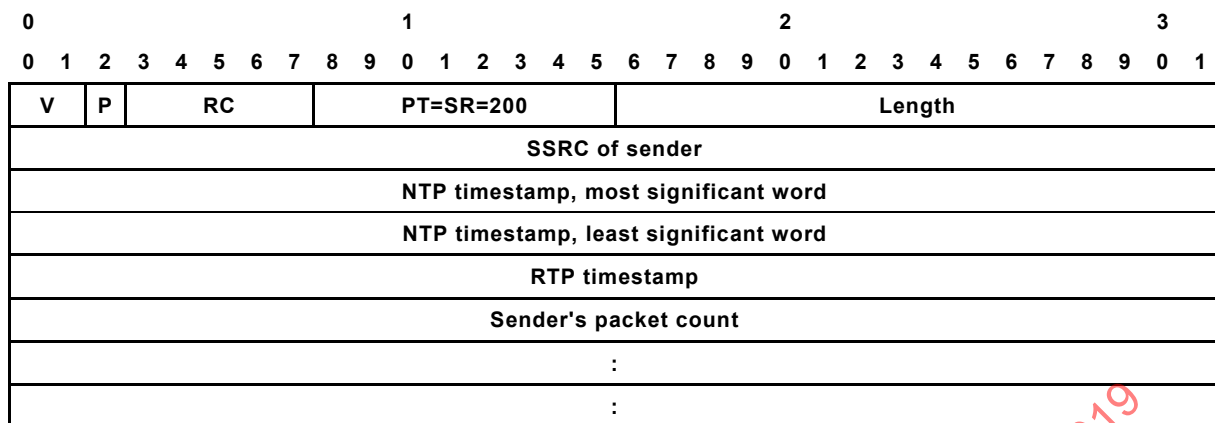


Figure 12 – RTCP Sender Report

The wall clock should be common in the device and each timestamp value should be determined properly. The client can synchronize different media streams at the appropriate timing based on the RTP clock and wall clock timestamps (see Figure 13).

In the case of multiple devices, the NTP timestamp should be common to all devices, and the NTP server should be required in the system⁸.

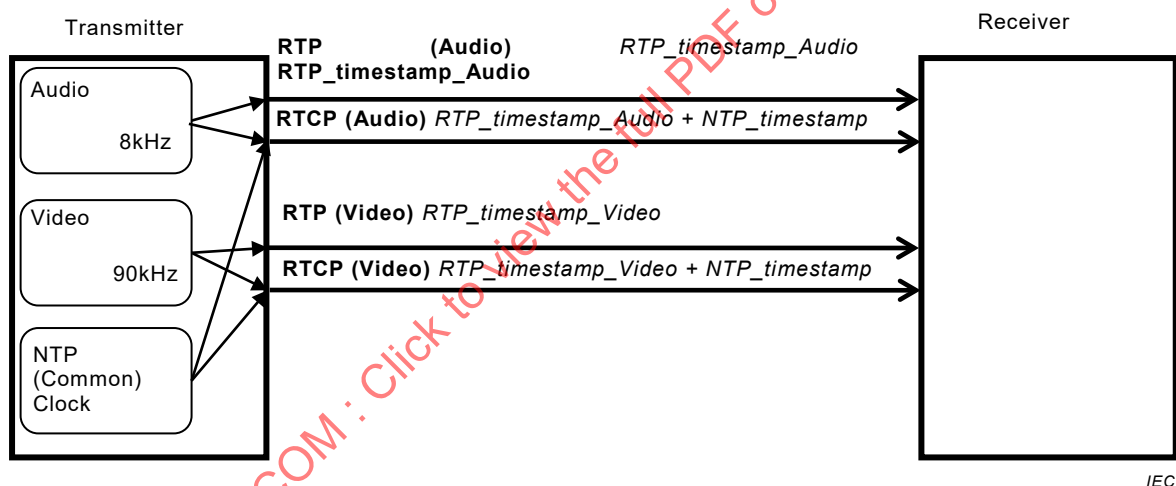


Figure 13 – Media synchronization

10.2.3 Synchronization points

Synchronization points allow clients to decode and correctly use data after the synchronization point. A synchronization point may be requested by a client in case of decoder error (e.g. in consequence of packet loss) to enforce the device to add an I-frame as soon as possible or to request the current PTZ or event status.

The Webservice-based methods require the support of the Synchronization Point request as defined in the section "Synchronization Point" of the ONVIF Media Service Specification.

⁸ The client can get information about "NTP server availability" from the devices by using the GetNTP command as defined in IEC 60839-11-31.

In addition, it is recommended that the PLI messages be supported as described in RFC 4585 in order to allow receivers as defined in the ONVIF Receiver Service Specification to request a synchronization point.

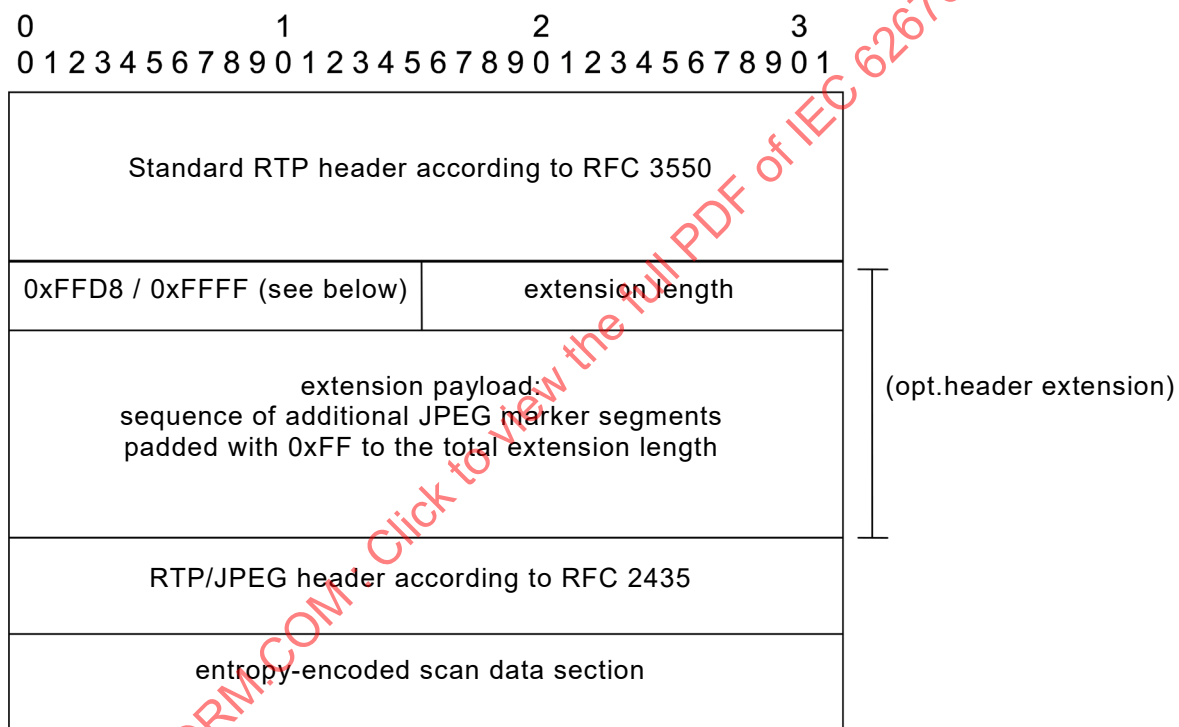
For H.264 and H.265 video, the SPS/PPS header shall be sent in-band if these have been changed during the transmission.

10.2.4 JPEG over RTP

10.2.4.1 Overall packet structure

The syntax for transmitting JPEG streams follows RFC 2435. The syntax does allow embedding additional data, beyond the limits of RFC 2435, by using an optional RTP header extension, as specified below, with some of the RTP packets. This option, however, changes the exact semantics for frames that include such packets.

The typical content of the overall format of a JPEG RTP packet is shown in Figure 14.



IEC

Figure 14 – RTP/JPEG packet structure

In order to distinguish an optional RTP header extension from possible other header extensions, the first 16 bits (the first two octets of the four-octet extension header) of an RTP shall have the value 0xFFD8 (JPEG SOI marker) for the initial packet and 0xFFFF for other RTP packets within a frame.

As required by RFC 3550, the presence of the optional header extension shall be signalled via the X-bit of the RTP header. The extension length field within the header extension counts the number of 32-bit items following as extension payloads. For example, a zero-length field following the 32-bit extension header represents an empty header extension.

The entropy-encoded scan data section may not be present in all RTP packets. A complete RTP/JPEG header however shall be present in the initial packet of every frame and all packets containing an entropy-encoded scan data section, otherwise it may be missing.

The fragment offset field within the RTP/JPEG header, in accordance with RFC 2435, should be used as if no header extension were present. Additionally, if a packet does not contain an entropy-encoded scan data segment, but contains a header extension, the fragment offset field shall not be zero if any packets containing an entropy-encoded scan data section for the same frame have been transmitted. If the initial packet of a frame contains no header extension, according to this document, its fragment offset field shall be zero, otherwise it should be zero. All packets including an RTP/JPEG header with a fragment offset of zero and a Q value between 128 and 255 shall include a quantization table header according to 3.1.8 of RFC 2435; other packets shall not include this header.

10.2.4.2 Logical decoding specification

For the decoding specification, it is assumed that the original packet order within the RTP stream has been restored in accordance with the RTP sequence numbering.

If the initial packet of a frame contains no RTP header extension as specified above, decoders shall generate the complete scan header and perform the decoding as specified by RFC 2435. The scan data sections and payloads of any header extension conforming to this specification, up to and including the next RTP packet with its marker bit set, shall be concatenated as they occur within the stream ignoring their fragment offset values.

Otherwise (at least an empty header extension as specified above is present in the initial packet of a frame), the following rules apply for each such frame:

- If the initial packet of a frame does not contain an entropy-encoded scan data segment, but contains a header extension as specified above, then decoders shall concatenate its header extension payload with (possibly empty or non-existent) header extension payload(s) conforming to this specification of the subsequent packets up to and including the first packet with the RTP marker bit set or containing an entropy-encoded scan data segment.
- The concatenated initial RTP header extension payload (sequence) shall be logically prepended with a JPEG SOI marker (0xFFD8).
- If the Q-value of the RTP/JPEG scan header within the initial packet of a frame is not zero, the quantization tables shall be pre-initialized in accordance with the rules of RFC 2435. If Q is equal to zero, the quantization tables shall be copied from the previous frame, allowing for DQT markers within this initial header extension payload (sequence) to override them.
- If this frame is the initial frame of a sequence, the Huffman tables shall be pre-initialized in accordance with RFC 2435. The Huffman tables for all subsequent frames shall be copied from the previous frame, allowing the frames to be overridden by DHT markers within the initial header extension payload (sequence).
- If the initial RTP header extension payload (sequence) supplies no DRI marker, but the RTP/JPEG header of the initial packet of a frame contains an RTP/JPEG restart marker, a DRI marker corresponding to the rules of RFC 2435 shall be appended to the initial header extension payload (sequence). Otherwise, if the initial RTP header extension (sequence) supplies a DRI marker, the marker shall take precedence over any other RTP/JPEG restart marker in accordance with RFC 2435 for the same frame. However, for compatibility with decoders conforming to RFC 2435 only, encoders normally should use an RTP/JPEG restart marker with consistent values, if restart intervals are to be used.
- DRI markers shall not be derived from previous frames.
- If the initial RTP header extension payload (sequence) supplies no SOF marker, which otherwise takes precedence, a SOF marker shall be appended to it with the following values:
 - if both the width and height field of the RTP/JPEG header are zero, the SOF marker of the previous frame shall be used;
 - otherwise, it shall be derived according to the rules of RFC 2435.

However, as long as the (rounded up) image size fits within the range as specified in RFC 2435, encoders should specify the image size within the RTP/JPEG header consistent with the values of an additional SOF header.

- If the initial header extension payload (sequence) supplies no SOS marker, a corresponding marker shall be derived in accordance with RFC 2435 and appended to it, otherwise the SOS marker in the extension takes precedence.

An SOS marker shall not be derived from previous frames.

If the SOS marker is present and not followed by entropy-encoded scan data within the extension, the marker shall be the final marker within the initial extension payload (sequence) of a frame. Necessary padding with 0xFF-octets shall not follow this marker but may precede it.

- The remaining entropy-encoded scan data and header extensions payloads shall be logically appended in the same order as they occur within the RTP stream up to the end of the frame as indicated by the RTP marker bit. A final EOI marker shall also be added if it is not yet present within the logical sequence for this frame.

For each frame, the resulting sequence up to and including the first (possibly added) EOI marker shall be a valid (possibly abbreviated) JPEG stream, resulting in one complete image from the decoding process for this frame. The meaning of any data after this first EOI marker for each frame is outside the scope of this specification.

Implementations should provide for each frame the complete JPEG headers. This holds especially for the width and height information as well as the quantization and Huffman tables. If such important information is not provided for each frame, both playback and multicast streaming may suffer from incomplete JPEG header information.

10.2.4.3 Supported colour spaces and sampling factors

A transmitter should use only greyscale and YCbCr colour space. A client shall support both greyscale and YcbCr.

The sampling factors for YcbCr shall correspond to the values supported by RFC 2435. For example, a sampling factor of 4:2:0 (preferred) or 4:2:2.

10.2.4.4 Pixel aspect ratio handling

The pixel aspect ratio of JPEG files can be specified within the JFIF marker. If the pixel aspect ratio is different from the standard 1:1 and 1:2 ratio in accordance with RFC 2435, this marker should be transmitted in the initial header extension payload (sequence) of every frame to specify the (for interlaced material field-based) pixel aspect ratio.

10.2.4.5 Interlaced handling

Interlaced video is encoded as two independent fields and signalled as specified by RFC 2435 within the RTP/JPEG header.

Both fields shall use the same colour space, sampling factors and pixel aspect ratio.

Interlaced encoding should not be used if the frame was originally scanned progressively.

10.3 Media control protocol

10.3.1 RTSP stream control

The media stream is controlled using the protocol defined in the URI as shown in Figure 15. The URI is returned in response to the GetStreamUri command defined in the ONVIF Media Service Specification.

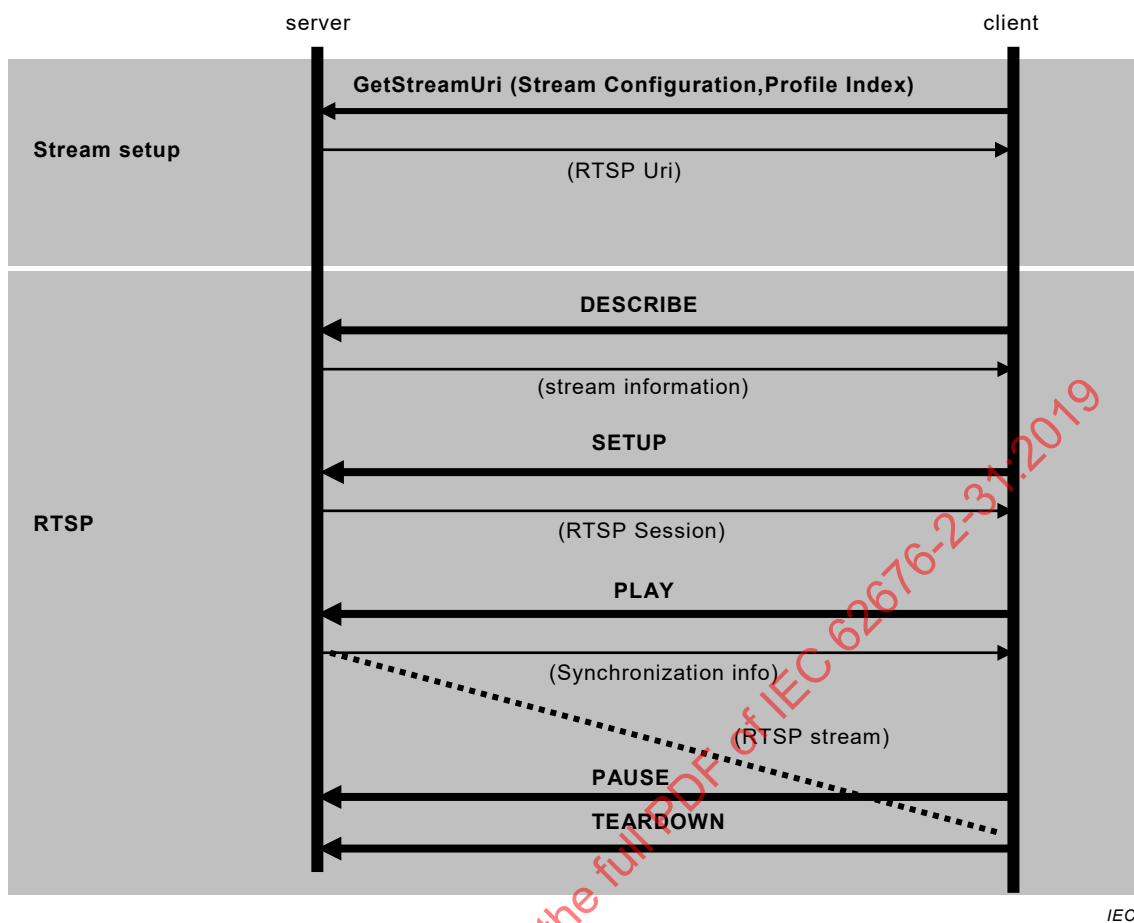


Figure 15 – Stream control

All devices and clients shall support RTSP (RFC 2326) for session initiation and playback control. RTSP shall use TCP as its transport protocol, the default TCP port for RTSP traffic is 554. The Session Description Protocol (SDP) shall be used to provide media stream information and SDP shall conform to RFC 4566. See Table 5 for requirements on individual methods.

Table 5 – RTSP methods

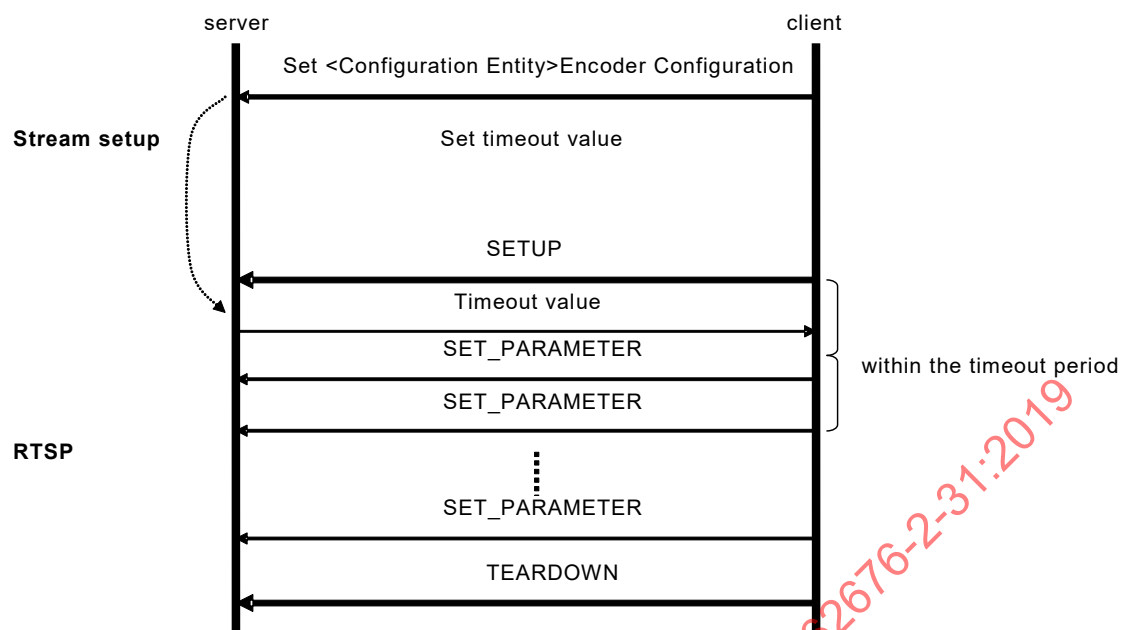
Method	Direction	SPEC	Description
OPTIONS	R->T T->R	M X	Required to get optional method capability and to allow different versions in the future.
DESCRIBE	R->T	M	Required to retrieve media parameters within the designated profile.
ANNOUNCE	R->T T->R	X	
SETUP	R->T	M	Required to set media session parameters.
PLAY	R->T	M	Required to start media stream.
PAUSE	R->T	Live: O Replay: M	Required to temporarily stop media playback. Handling multiple streams in a narrow bandwidth network, by suspending RTP stream, the traffic can be well controlled by reducing redundant data and congested network traffic can be avoided.
TEARDOWN	R->T	M	Required to release a media session.
GET_PARAMETER	R->T T->R	O	
SET_PARAMETER	R->T T->R	O O	An optional method to keep an RTSP session alive (R->T direction only).
REDIRECT	T->R	X	
RECORD	R->T	X	
Key SPEC: The method is not supported (X), mandatory (M), or optional (O).			

Devices shall support aggregate stream control, in which play commands are sent to the control URI for the whole session. Devices may support non-aggregate stream control, in which play commands are sent separately to the stream control URIs in the media sections of the SDP file. Support for non-aggregate stream control is signalled via the Media Streaming capabilities.

10.3.2 Keep-alive method for RTSP session

A RTSP client keeps the RTSP session alive and prevents it from session timeout (see 12.37 of RFC 2326). This specification recommends the following methods to keep RTSP alive for both unicast and multicast streaming.

- 1) The client can optionally set the Timeout parameter (in seconds) using the Set<configurationEntity>EncoderConfiguration command defined in the ONVIF Media Service Specification, otherwise a default value of "60" is used.
- 2) In all RTSP SETUP responses, a transmitter should include the Timeout value according to 12.37 of RFC 2326 and the transmitter should use the Timeout value for keep-alive.
- 3) To keep the RTSP session alive, a client shall call the RTSP server using any RTSP method or send RTCP receiver reports. SET_PARAMETER is the recommended RTSP method to use as shown in Figure 16.



IEC

Figure 16 – Keep alive

10.3.3 RTSP audio and video synchronization

In order that clients may immediately begin synchronizing video and audio streams, and computing absolute UTC timestamps for incoming packets for recording purposes, a transmitter should include the following header fields in the RTSP PLAY response:

- Range (12.29 of RFC 2326). This shall include a start time in clock units (3.7 of RFC 2326), not SMPTE or NPT units.
- RTP-Info (12.33 RFC 2326 12.33). This shall include an rtptime value which corresponds to the start time specified in the Range header.

Example:

```

client->>server: PLAY rtsp://example.com/onvif_camera/video RTSP/1.0
                  Cseq: 4
                  Range: npt=now-
                  Session: 12345678

server->>client: RTSP/1.0 200 OK
                  Cseq: 4
                  Session: 12345678
                  Range: clock=20100217T143720.257Z-
                  RTP-Info: url=rtsp://example.com/onvif_camera/video;
                  seq=1234;rtptime=3450012
  
```

10.3.4 RTSP session for a metadata stream

In the case of a metadata stream, the SDP description "application" shall be used in the DESCRIBE response for media type and one of these encoding names shall be used:

- "vnd.onvif.metadata" for uncompressed;
- "vnd.onvif.metadata.gzip" for GZIP compressed;
- "vnd.onvif.metadata.exi.onvif" for EXI using ONVIF default compression parameters;
- "vnd.onvif.metadata.exi.ext" for EXI using compression parameters that are sent in-band.

Example RTSP DESCRIBE message exchange between an RTSP server (server) and an RTSP client (client):

```
client->server: DESCRIBE rtsp://example.com/onvif_camera RTSP/1.0
                  Cseq: 1

server->client: RTSP/1.0 200 OK
                  Cseq: 1
                  Content-Type: application/sdp
                  Content-Length: XXX
                  v=0
                  o=- 2890844256 2890842807 IN IP4 172.16.2.93
                  s=RTSP Session
                  m=audio 0 RTP/AVP 0
                  a=control:rtsp://example.com/onvif_camera /audio
                  m=video 0 RTP/AVP 26
                  a=control:rtsp://example.com/onvif_camera /video
                  m=application 0 RTP/AVP 107
                  a=control:rtsp://example.com/onvif_camera/metadata
                  a=recvonly
                  a=rtpmap
                  a=rtpmap:107 vnd.onvif.metadata/90000
```

10.3.5 Multicast streaming

A device shall include a valid multicast address in the "c=" field of a DESCRIBE response in accordance with RFC 4566.

Remark: the optional dynamic multicast address assignment exception described in C.1.7 of RFC 2326 that allows 0.0.0.0 addresses does not apply.

Chapter 10.7 (TEARDOWN) of RFC 2326 states that a device shall stop the stream delivery for the given URI on teardown. This needs to be clarified in the case of multicast: for a multicast stream the device shall stop sending packets for a multicast configuration when no more RTSP sessions are using the same multicast configuration nor its AutoStart flag has been set.

10.3.6 RTSP message example

This example shows the message transfer between an RTSP client (client) and an RTSP server (server). The client requests one audio and one video stream from the device. The stream URI "rtsp://example.com/onvif_camera" can be retrieved using the GetStreamUri command. Refer to Section "Stream URI" of the ONVIF Media Service Specification.

```
Client->server:      DESCRIBE rtsp://example.com/onvif_camera RTSP/1.0
                      Cseq: 1

server->client:      RTSP/1.0 200 OK
                      Cseq: 1
                      Content-Type: application/sdp
                      Content-Length: XXX
                      v=0
                      o=- 2890844256 2890842807 IN IP4 172.16.2.93
                      s=RTSP Session
                      m=audio 0 RTP/AVP 0
                      a=control:rtsp://example.com/onvif_camera/audio
                      m=video 0 RTP/AVP 26
                      a=control:rtsp://example.com/onvif_camera/video

client->server:      SETUP rtsp://example.com/onvif_camera/audio RTSP/1.0
                      Cseq: 2
                      Transport: RTP/AVP;unicast;client_port=8002-8003

server->client:      RTSP/1.0 200 OK
                      Cseq: 2
```

```

Transport: RTP/AVP;unicast;client_port=8002-8003;
server_port=9004-9005
Session: 12345678; timeout=60

client->server:    SETUP rtsp://example.com/onvif_camera/video RTSP/1.0
                  Cseq: 3
                  Transport: RTP/AVP;unicast;client_port=8004-8005
                  Session: 12345678

server->client:    RTSP/1.0 200 OK
                  Cseq: 3
                  Transport: RTP/AVP;unicast;client_port=8004-8005;
                  server_port=9006-9007
                  Session: 12345678; timeout=60

client->server:    PLAY rtsp://example.com/onvif_camera RTSP/1.0
                  Cseq: 4
                  Range: npt=now-
                  Session: 12345678

server->client:    RTSP/1.0 200 OK
                  Cseq: 4
                  Session: 12345678
RTP-Info: url=rtsp://example.com/onvif_camera/video; seq=1234;rtptime=3450012,
url=rtsp://example.com/onvif_camera/audio;
seq=22434;rtptime=1234566

client->server:    TEARDOWN rtsp://example.com/onvif_camera RTSP/1.0
                  Cseq: 5
                  Session: 12345678

server->client:    RTSP/1.0 200 OK
                  Cseq: 5
                  Session: 12345678

```

10.3.7 RTSP over HTTP

The RTSP over HTTP/HTTPS shall be supported in order to traverse a firewall. See 10.2.1.5 RTP/RTSP/HTTP/TCP.

10.4 Back channel connection

10.4.1 General

Subclause 10.4 describes how a bidirectional connection can be established between a client and a server. The backchannel connection handling is done using RTSP (see RFC 2326). Therefore, a mechanism is introduced that indicates that a client wants to establish a backchannel connection. RTSP provides feature tags to deal with such functionality additions.

A device that supports bi-directional connections (for example, audio or metadata connections) shall support the introduced RTSP extensions.

10.4.2 RTSP Require tag

The RTSP standard RFC 2326 can be extended by using additional header objects. For that purpose, a Require tag is introduced to handle special functionality additions (see 1.5 [Extending RTSP] and 12.32 [Require] of RFC 2326).

The Require tag is used to determine the support of this feature. This header shall be included in any request where the server is required to understand that feature to correctly perform the request.

A device that supports backchannel and signals Audio output support via the AudioOutputs capability shall understand the backchannel tag:

- www.onvif.org/ver20/backchannel

An RTSP client that wants to build up an RTSP connection with a data backchannel shall include the Require header in its requests.

10.4.3 Connection setup for a bi- directional connection

A client shall include the feature tag in its DESCRIBE request to indicate that a bidirectional data connection shall be established.

A server that understands this Require tag shall include an additional media stream in its SDP file as configured in its media profile.

An RTSP server that does not understand the backchannel feature tag or does not support bidirectional data connections shall respond with an error code "551 Option not supported" in accordance with the RTSP standard. The client can then try to establish an RTSP connection without backchannel.

An SDP file is used to describe the session. The server shall include the a=sendonly or the a=recvonly attributes in each media section of the SDP file to indicate the direction the media data will be send.

The server shall list all supported decoding codecs as its own media section and the client chooses which one is used.

10.4.4 Describe example for a server without backchannel support:

```
Client - Server:    DESCRIBE rtsp://192.168.0.1 RTSP/1.0
                   Cseq: 1
                   User-Agent: ONVIF Rtsp client
                   Accept: application/sdp
                   Require: www.onvif.org/ver20/backchannel

Server - Client:    RTSP/1.0 551 Option not supported
                   Cseq: 1
                   Unsupported: www.onvif.org/ver20/backchannel
```

10.4.5 Describe example for a server with ONVIF backchannel support:

```
Client - Server:    DESCRIBE rtsp://192.168.0.1 RTSP/1.0
                   Cseq: 1
                   User-Agent: ONVIF Rtsp client
                   Accept: application/sdp
                   Require: www.onvif.org/ver20/backchannel

Server - Client:    RTSP/1.0 200 OK
                   Cseq: 1
                   Content-Type: application/sdp
                   Content-Length: xxx

                   v=0
                   o= 2890842807 IN IP4 192.168.0.1
                   s=RTSP Session with audiobackchannel
                   m=video 0 RTP/AVP 26
                   a=control:rtsp://192.168.0.1/video
                   a=recvonly
                   m=audio 0 RTP/AVP 0
                   a=control:rtsp://192.168.0.1/audio
                   a=recvonly
                   m=audio 0 RTP/AVP 0
                   a=control:rtsp://192.168.0.1/audioback
                   a=rtptime:0 PCMU/8000
                   a=sendonly
```

This SDP file completely describes the RTSP session. The server gives the client its control URLs to set up the streams.

In the next step, the client can set up the sessions:

```
Client - Server:      SETUP rtsp://192.168.0.1/video RTSP/1.0
                      Cseq: 2
                      Transport: RTP/AVP;unicast;client_port=4588-4589

Server - Client:      RTSP/1.0 200 OK
                      Cseq: 2
                      Session: 123124;timeout=60
                      Transport:RTP/AVP;unicast;client_port=4588-4589; server_port=6256-6257

Client - Server:      SETUP rtsp://192.168.0.1/audio RTSP/1.0
                      Cseq: 3
                      Session: 123124
                      Transport: RTP/AVP;unicast;client_port=4578-4579

Server - Client:      RTSP/1.0 200 OK
                      Cseq: 3
                      Session: 123124;timeout=60
                      Transport:RTP/AVP;unicast;client_port=4578-4579; server_port=6276-6277

Client - Server:      SETUP rtsp://192.168.0.1/audioback RTSP/1.0
                      Cseq: 4
                      Session: 123124
                      Transport: RTP/AVP;unicast;client_port=6296-6297
                      Require: www.onvif.org/ver20/backchannel

Server - Client:      RTSP/1.0 200 OK
                      Cseq: 4
                      Session: 123124;timeout=60
                      Transport:RTP/AVP;unicast;client_port=6296-6297; server_port=2346-2347
```

The third setup request establishes the audio backchannel connection.

In the next step, the client starts the session by sending a PLAY request.

```
Client - Server:      PLAY rtsp://192.168.0.1 RTSP/1.0
                      Cseq: 5
                      Session: 123124
                      Require: www.onvif.org/ver20/backchannel

Server - Client:      RTSP/1.0 200 OK
                      Cseq: 5
                      Session: 123124;timeout=60
```

After receiving the OK response to the PLAY request, the client can start sending audio data to the server. It shall not start sending data to the server before it has received the response.

The Require header indicates that a special interpretation of the PLAY command is necessary. The command covers both starting of the video and audio stream from NVT to the client and starting the audio connection from client to server.

To terminate the session, the client sends a TEARDOWN request.

```
Client - NVT:         TEARDOWN rtsp://192.168.0.1 RTSP/1.0
                      Cseq: 6
                      Session: 123124
                      Require: www.onvif.org/ver20/backchannel

NVT - Client:         RTSP/1.0 200 OK
                      Cseq: 6
                      Session: 123124
```

10.4.6 Multicast streaming

If the client intends to send its data via multicast, it shall use the transport parameter in the SETUP request to tell the server the multicast address and port.

Example:

```
Client - Server:      SETUP rtsp://192.168.0.1/audioback RTSP/1.0
                     Cseq: 4
                     Session: 123124
                     Transport:RTP/AVP;multicast;destination=224.2.1.1;port=60000-60001;ttl=128
                     Require: www.onvif.org/ver20/backchannel

Server - Client:     RTSP/1.0 200 OK
                     Cseq: 4
                     Session: 123124;timeout=60
                     Transport:RTP/AVP;multicast;destination=224.2.1.1;
                     port=6000060001;ttl=128;mode="PLAY"
```

10.5 Error handling

RTSP and HTTP protocol errors are classified into different categories (for example, status codes 1xx, 2xx, 3xx, 4xx and 5xx). The device and the client shall support and handle these status codes. For RTSP status code definitions, refer to 11.0 of RFC 2326. For HTTP status code definitions, refer to 10.0 (HTTP/1.1) of RFC 2616.

IECNORM.COM : Click to view the full PDF of IEC 62676-2-31:2019

Annex A (normative)

Efficient XML Interchange (EXI)

EXI encoding allows for a more compact representation of XML metadata. Provision is signalled if the CompressionType returned via GetMetadataConfigurationOptions contains "EXI". The ONVIF defined EXI configuration (see Table A.1 and Table A.2) shall be supported by all devices signalling support for EXI compression. Schema-based EXI encoding shall be used. The required schema can be obtained from any device using the GetWsdlUrl command.

The EXI header shall only be transmitted if a setting different from the ONVIF defined configuration is used. Except for the setting of the two elements "Presence Bit" and "EXI Options", the ONVIF defined EXI header settings (see Table A.1) shall always be used.

Table A.1 – ONVIF defined EXI header settings

Exi header element	Value
EXI Cookie	mandatory
Distinguishing Bits	mandatory
EXI Format Version	0 0000
Presence Bit for EXI Options	0
Exi Options	see Table A.2
Padding Bits	If present shall be "0".

Table A.2 – ONVIF defined EXI configuration settings

Exi Option	Value
alignment	default (bit-packed)
compression	default (false)
strict	default (false)
fragment	default (false)
preserve	default (all false)
selfContained	default (false)
schemaID	Insert reference to schema obtained from device here.
datatypeRepresentationMap	none
blockSize	default (1 000 000)
valueMaxLength	default (unbounded)
valuePartitionCapacity	default (unbounded)
user defined meta-data	none

Annex B (normative)

Lens description

Wide angle lenses produce a geometrically distorted image. The lens description describes where an incoming light beam hits the image sensor as depicted in Figure B.1. This allows clients to compensate the distortion. This specification describes only the mapping from incoming light beams to sensor location. The actual method of compensating the distortion can be freely chosen by the client and is outside of this specification.

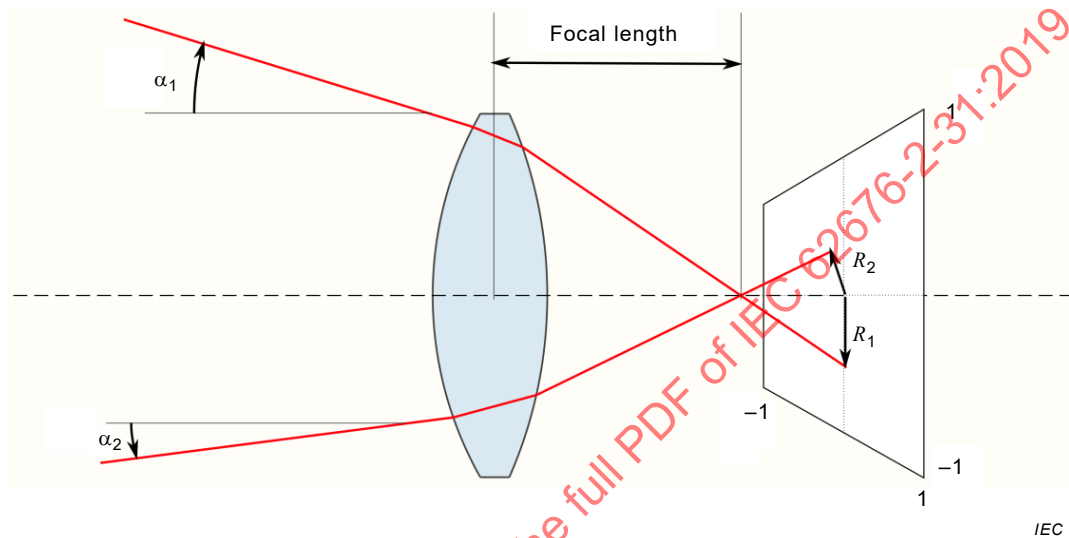


Figure B.1 – Optical mapping of angle (α) via radius (R) to normalized x/y coordinates

The lens projection function is defined by a set of points. A smooth mapping is achieved by using B-spline approximation as shown in Figure B.2.

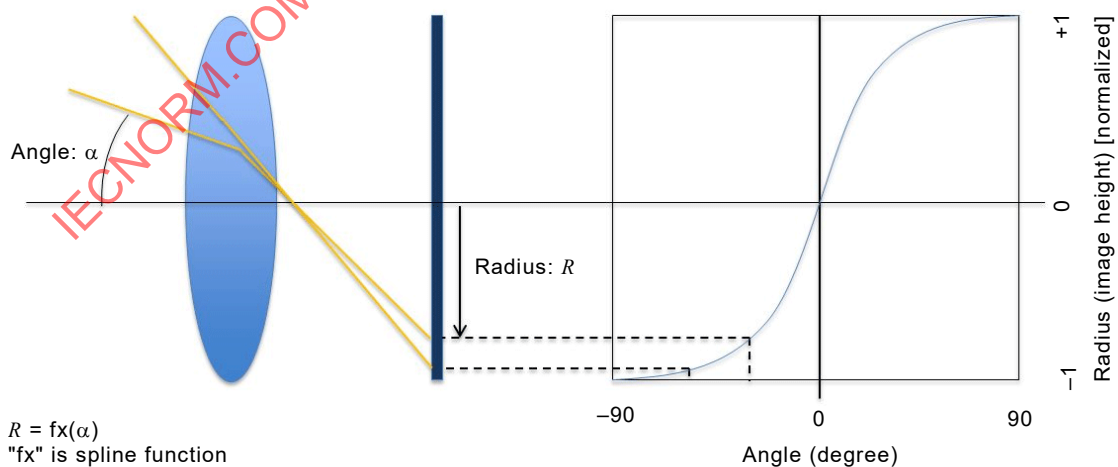
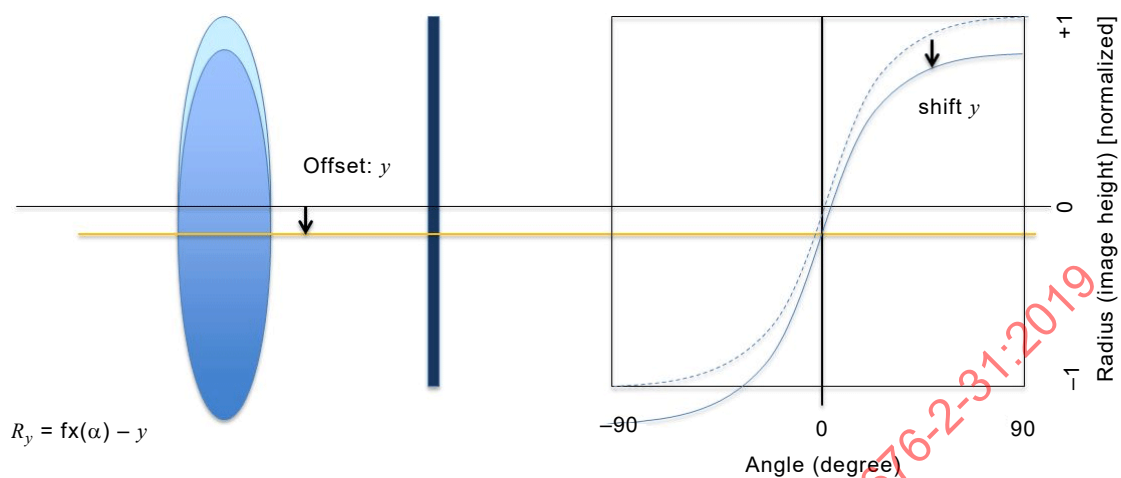


Figure B.2 – Smooth mapping using B-splines

Lens to imager axis offsets can be described via the so-called LensOffset. Figure B.3 depicts that the compensation of a vertical offset results in a shift of the spline curve.



IEC

Figure B.3 – Compensation of vertical axis offset

IECNORM.COM : Click to view the full PDF of IEC 62676-2-31:2019

Annex C (informative)

Specified rules

C.1 General

The following rules apply to static cameras. In the case of a PTZ device, image-based rules should contain an additional `ElementItem`. `ElementItem` identifies the position of the device for which the rule has been setup. The corresponding `ElementItemDescription` resembles the following:

```
<tt:ElementItemDescription Name="PTZStatus" Type="tt:PTZStatusType">
```

C.2 LineDetector

`LineDetector` is defined by a non-intersecting simple polyline. If an object crosses the polyline in the specified direction, the rule engine sends a `Crossed` event containing the name of the `LineDetector` and a reference to the object which has crossed the line. As directions, one can select between `Left`, `Right`, and `Any`, where directions `Left` and `Right` refer to the direction walking along the line from the first point to the second point and are the prohibited directions.

The `LineDetector` resembles the following code using the rule description language, detailed in the previous section:

```
<tt:RuleDescription Name="tt:LineDetector">
  <tt:Parameters>
    <tt:SimpleItemDescription Name="Direction" Type="tt:Direction"/>
    <tt:ElementItemDescription Name="Segments" Type="tt:Polyline"/>
  </tt:Parameters>
  <tt:Messages>
    <tt:Source>
      <tt:SimpleItemDescription Name="VideoSourceConfigurationToken"
        Type="tt:ReferenceToken"/>
      <tt:SimpleItemDescription Name="VideoAnalyticsConfigurationToken"
        Type="tt:ReferenceToken"/>
      <tt:SimpleItemDescription Name="Rule" Type="xs:string"/>
    </tt:Source>
    <tt>Data>
      <tt:SimpleItemDescription Name="ObjectId" Type="xs:integer"/>
    </tt>Data>
    <tt:ParentTopic>tns1:RuleEngine/LineDetector/Crossed</tt:ParentTopic>
  </tt:Messages>
</tt:RuleDescription>
```

The code above defines two parameters, `Segments` and `Direction`, and produces one event attached to the topic `tns1:RuleEngine/LineDetector/Crossed`.

C.3 FieldDetector

`FieldDetector` is defined by a simple non-intersecting polygon. The `FieldDetector` determines if each object is in the scene inside or outside the polygon. This information is put into a property.

`FieldDetector` resembles the following code, using the rule description language detailed in the previous section:

```

<tt:RuleDescription Name="tt:FieldDetector">
  <tt:Parameters>
    <tt:ElementItemDescription Name="Field" Type="tt:Polygon"/>
  </tt:Parameters>
  <tt:Messages IsProperty="true">
    <tt:Source>
      <tt:SimpleItemDescription Name="VideoSourceConfigurationToken"
                                Type="tt:ReferenceToken"/>
      <tt:SimpleItemDescription Name="VideoAnalyticsConfigurationToken"
                                Type="tt:ReferenceToken"/>
      <tt:SimpleItemDescription Name="Rule" Type="xs:string"/>
    </tt:Source>
    <tt:Key>
      <tt:SimpleItemDescription Name="ObjectId" Type="xs:integer"/>
    </tt:Key>
    <tt>Data>
      <tt:SimpleItemDescription Name="IsInside" Type="xs:boolean"/>
    </tt>Data>
    <tt:ParentTopic>
      tns1:RuleEngine/FieldDetector/ObjectsInside
    </tt:ParentTopic>
  </tt:Messages>
</tt:RuleDescription>

```

From the Inside property, a client can derive the Entering and the Leaving parameters of the detector. A client can simulate Entering and Leaving events by adding a MessageContent filter to the subscription, which lets only ObjectsInside messages pass, where the IsInside Item is set to true resp. false.

C.4 LoiteringDetector

A LoiteringDetector is defined by a simple non-intersecting polygon as an area of interest and threshold loitering interval. LoiteringDetector determines if each object in the scene inside the polygon is longer than the given time threshold value. It publishes when the object started loitering.

LoiteringDetector is defined by the following code using the rule description language:

```

<tt:RuleDescription Name="tt:LoiteringDetector">
  <tt:Parameters>
    <tt:ElementItemDescription Name="Field"
                              Type="tt:PolygonConfiguration"/>
    <tt:SimpleItemDescription Name="TimeThreshold"
                              Type="xs:nonNegativeInteger"/>
    <tt:SimpleItemDescription Name="isPeriodicNotify" Type="xs:boolean"/>
    <tt:SimpleItemDescription Name="TimeInterval" Type="xs:duration"/>
  </tt:Parameters>
  <tt:Messages IsProperty="true">
    <tt:Source>
      <tt:SimpleItemDescription Name="VideoSourceConfigurationToken"
                                Type="tt:ReferenceToken"/>
      <tt:SimpleItemDescription Name="VideoAnalyticsConfigurationToken"
                                Type="tt:ReferenceToken"/>
      <tt:SimpleItemDescription Name="Rule" Type="xs:string"/>
    </tt:Source>
    <tt:Key>
      <tt:SimpleItemDescription Name="ObjectId" Type="xs:integer"/>
    </tt:Key>
    <tt>Data>
      <tt:SimpleItemDescription Name="Since" Type="xs:dateTime"/>
      <tt:SimpleItemDescription Name="TimeStamp" Type="xs:dateTime"/>
    </tt>Data>
    <tt:ParentTopic>
      tns1:RuleEngine/LoiteringDetector/ObjectIsLoitering
    </tt:ParentTopic>
  </tt:Messages>
</tt:RuleDescription>

```

The above rule definition defines that a rule instance produces an event attached to the topic `tns1:RuleEngine/LoiteringDetector/ObjectIsLoitering`, and the configuration parameters are given in Table C.1.

Table C.1 – Loitering Detector rule configuration parameters

Parameter name	Description
Field	Area in camera field of view to detect object loitering
TimeThreshold	Maximum time threshold to judge that object is loitering
isPeriodicNotify	Whether continue to generate periodic events for loitering object
TimeInterval	Periodic time interval to continue to send loitering events for loitering object

The event contains the fields given in Table C.2.

Table C.2 – Description of loitering event fields

Parameter name	Description
ObjectId	Object identifier
Since	Since when this object is loitering
TimeStamp	Timestamp of last object sample triggering generation of rule's event

C.5 Declarative motion detector

The declarative motion detector detects the object motion against the specified motion attributes. The rule is active within an area (region of interest) and time interval. The condition is defined against the frame/object element in scene descriptor.

```
<tt:RuleDescription Name="tt:DeclarativeMotionDetector">
  <tt:Parameters>
    <tt:ElementItemDescription Name="Field" Type="tt:PolygonConfiguration"/>
    <tt:ElementItemDescription Name="MotionExpression"
      Type="tt:MotionExpressionConfiguration"/>
  </tt:Parameters>
  <tt:Messages IsProperty="true">
    <tt:Source>
      <tt:SimpleItemDescription Name="VideoSourceConfigurationToken"
        Type="tt:ReferenceToken"/>
      <tt:SimpleItemDescription Name="VideoAnalyticsConfigurationToken"
        Type="tt:ReferenceToken"/>
      <tt:SimpleItemDescription Name="Rule" Type="xs:string"/>
    </tt:Source>
    <tt:Key>
      <tt:SimpleItemDescription Name="ObjectId" Type="xs:integer"/>
    </tt:Key>
    <tt>Data>
      <tt:SimpleItemDescription Name="TimeStamp" Type="xs:dateTime"/>
    </tt>Data>
    <tt:ParentTopic>
      tns1:RuleEngine/DeclarativeMotionDetector/MotionMatched
    </tt:ParentTopic>
  </tt:Messages>
</tt:RuleDescription>
```

The above rule description defines that a rule instance produces an event attached to the topic `tns1:RuleEngine/DeclarativeMotionDetector/MotionMatched`, and the configuration parameters are given in Table C.3.

Table C.3 – Declarative motion detector rule configuration parameters

Parameter Name	Description
Field	Defines an area in camera field of view, the motion matching query is applied in this area for detected object.
MotionExpression	Motion expression for frame/object elements in Scene Descriptor schema

The event contains the fields given in Table C.4.

Table C.4 – Description of declarative motion event fields

Parameter name	Description
ObjectId	Object identifier
TimeStamp	Timestamp of detected motion

C.6 Counting rule

The counting rule counts the number of motion object passing through the set of line segments (barrier) and optional direction attribute. The configuration parameters also include the time interval to report the events and time interval to reset its counter.

```

<tt:RuleDescription Name="tt:CountAggregation">
  <tt:Parameters>
<tt:ElementItemDescription Name="LineSegments"
                                Type="tt:PolylineArrayConfiguration"/>
    <tt:SimpleItemDescription Name="ReportTimeInterval" Type="xs:duration"/>
    <tt:SimpleItemDescription Name="ResetTimeInterval" Type="xs:duration"/>
    <tt:SimpleItemDescription Name="Direction" Type="tt:Direction"/>
  </tt:Parameters>
  <tt:Messages IsProperty="true">
    <tt:Source>
      <tt:SimpleItemDescription Name="VideoSourceConfigurationToken"
                                Type="tt:ReferenceToken"/>
      <tt:SimpleItemDescription Name="VideoAnalyticsConfigurationToken"
                                Type="tt:ReferenceToken"/>
      <tt:SimpleItemDescription Name="Rule" Type="xs:string"/>
    </tt:Source>
    <tt:Key>
      <tt:SimpleItemDescription Name="ObjectId" Type="xs:integer"/>
    </tt:Key>
    <tt>Data>
      <tt:SimpleItemDescription Name="Count" Type="xs:nonNegativeInteger"/>
    </tt>Data>
    <tt:ParentTopic>
      tns1:RuleEngine/CountAggregation/Counter
    </tt:ParentTopic>
  </tt:Messages>
</tt:RuleDescription>

```

The above rule description defines that a rule instance produces a count event of topic `tns1:RuleEngine/CountAggregation/Counter` and the configuration parameters are given in Table C.5.

Table C.5 – Counting rule configuration parameters

Parameter name	Description
LineSegments	Array of lines used for detecting that object passed all of them for counting
ReportTimeInterval	Time interval to report count information
ResetTimeInterval	Periodic count reset time
Direction	Count direction

The event contains the fields given in Table C.6.

Table C.6 – Description of counting event fields

Parameter name	Description
ObjectId	Object identifier of last counted object
Count	Value of counter

C.7 Query rule

The query rule allows a client to define a query on streamed metadata of multimedia streams. The client subscribes to the topic associated with the rule to receive notifications from a device.

```

<tt:RuleDescription Name="tt:QueryRule">
  <tt:Parameters>
    <tt:ElementItemDescription Name="Query" Type="tt:MetadataFilter"/>
    <tt:SimpleItemDescription Name="VideoSource" Type="tt:ReferenceTokenList"/>
  </tt:Parameters>
  <tt:Messages>
    <tt:Source>
      <tt:SimpleItemDescription Name="VideoSourceConfigurationToken"
        Type="tt:ReferenceToken"/>
      <tt:SimpleItemDescription Name="VideoAnalyticsConfigurationToken"
        Type="tt:ReferenceToken"/>
      <tt:SimpleItemDescription Name="Rule" Type="xs:string"/>
    </tt:Source>
    <tt>Data>
      <tt:ElementItemDescription Name="Result" Type="xs:any"/>
    </tt>Data>
    <tt:ParentTopic>
      tns1:RuleEngine/QueryRule
    </tt:ParentTopic>
  </tt:Messages>
</tt:RuleDescription>

```

The topic is tns1:RuleEngine/QueryRule and the value of "Rule" in the notification event indicates which Query Rule instance generated the event. The client should subscribe the topic with the proper rule name to receive the events from the specific query rule. The configuration parameters are given in Table C.7.

Table C.7 – Query Rule configuration parameters

Parameter name	Description
Query	Contains the query definition
VideoSource	Selects which video source or sources the query will execute on

Annex D (informative)

Cell motion detection

D.1 Cell motion detector

Cell motion detector rules process the output of the cell motion analytics engine. The rule of type "tt:CellMotionDetector" consists of four parameters: "MinCount", "AlarmOnDelay", "AlarmOffDelay" and "ActiveCells" as listed in Table D.1.

The parameter "MinCount" describes the minimum number of adjacent cells that a moving object shall cover in order to generate an alarm event.

AlarmOnDelay and AlarmOffDelay are delay times in milliseconds. These delays should prevent alarm state changes within a short time. An alarm has to be active for at least AlarmOnDelay time to trigger an alarm. AlarmOffDelay will delay the transition from an activated alarm to an inactive alarm. An alarm has to be inactive for at least AlarmOffDelay time to become inactive again.

Each of the cells defined by the type "tt:CellLayout" in the AnalyticsModule can be activated or deactivated individually. To exclude particular regions from being monitored owing to continuous movements (e.g. a tree in the wind), the relevant cells can be deactivated. For a representation of the activated and deactivated cells, a bitmask is used. For instance, if there are 1 000 cells available, then 1 000 bits need to be set. To efficiently encode this bitmask, the algorithm "Packbits" from ISO 12369 (TIFF, Version 6.0), which is a kind of run length encoder, is used. Afterwards, the resulting output is stored as base64Binary.

The rule definition is as follows:

```
<tt:RuleDescription Name="tt:CellMotionDetector">
  <tt:Parameters>
    <tt:SimpleItemDescription Name="MinCount" Type="xs:integer"/>
    <tt:SimpleItemDescription Name="AlarmOnDelay" Type="xs:integer"/>
    <tt:SimpleItemDescription Name="AlarmOffDelay" Type="xs:integer"/>
    <tt:SimpleItemDescription Name="ActiveCells" Type="xs:base64Binary"/>
  </tt:Parameters>
  <tt:Messages IsProperty="true">
    <tt:Source>
      <tt:SimpleItemDescription
        Name="VideoSourceConfigurationToken"
        Type="tt:ReferenceToken"/>
      <tt:SimpleItemDescription
        Name="VideoAnalyticsConfigurationToken"
        Type="tt:ReferenceToken"/>
      <tt:SimpleItemDescription Name="Rule" Type="xs:string"/>
    </tt:Source>
    <tt>Data>
      <tt:SimpleItemDescription Name="IsMotion" Type="xs:boolean"/>
    </tt>Data>
    <tt:ParentTopic>
      tns1:RuleEngine/CellMotionDetector/Motion
    </tt:ParentTopic>
  </tt:Messages>
</tt:RuleDescription>
```

Table D.1 – Cell motion detector rule configuration parameters

Parameter name	Description
MinCount	Minimum count of adjacent activated cells to trigger motion. This parameter allows suppressing false alarms.
AlarmOnDelay	Minimum time in milliseconds that is needed to change the alarm state from off to on. This delay is intended to prevent very brief alarm events from triggering.
AlarmOffDelay	Minimum time in milliseconds that is needed to change the alarm state from on to off. This delay is intended to prevent too many alarm changes.
ActiveCells	A "1" denotes an active cell and a "0" an inactive cell. The first cell is in the upper left corner. Then the cell order goes first from left to right and then from top to bottom (see 9.2.3.7). If the number of cells is not a multiple of 8, the last byte is filled with zeros. The information is run length encoded according to Packbit coding in ISO 12369 (TIFF, Revision 6.0).

The event contains the field given in Table D.2.

Table D.2 – Description cell motion detected event fields

Parameter Name	Description
IsMotion	True if motion is detected

The following code snippet is for cell motion detector rule configuration.

```

<tt:VideoAnalyticsConfiguration>
  <tt:RuleEngineConfiguration>
    <tt:Rule Name="MyMotionDetector" Type="tt:CellMotionDetector">
      <tt:Parameters>
        <tt:SimpleItem Name="MinCount" Value="4"/>
        <tt:SimpleItem Name="AlarmOnDelay" Value="1000"/>
        <tt:SimpleItem Name="AlarmOffDelay" Value="1000"/>
        <tt:SimpleItem Name="ActiveCells" Value="/v/+8A==" />
      </tt:Parameters>
    </tt:Rule>
  </tt:RuleEngineConfiguration>
</tt:VideoAnalyticsConfiguration>

```

The activated cells are filled with grey and the deactivated cells are white. The active cells in hex coding are: "ff ff ff f0 f0 f0". This is encoded with the Packbit algorithm: "fe ff fe f0". Finally, this packed data is base64Binary encoded: "/v/+8A==".

D.2 Cell motion analytics engine

D.2.1 General

A cell motion detector determines the motion in an image. For this, the image is subdivided into square or rectangle sensor fields, denoted as cells. A cell is based on a number of pixels, for example, a 5 × 5 cell is a cell of 25 pixels. The detector tries simply to estimate motion in each cell.

A cell motion detector consists of two units: AnalyticsModule and Rule. AnalyticsModule consists of two parameters: "CellLayout" and "Sensitivity" as listed in Table D.3. The latter can adjust the motion detector to the environment to which the camera is subjected. For lower sensitivity values, higher variations of the luminance values are needed to be identified as motion. Hence, a low sensitivity can be used to suppress estimated motion owing to noise.

The subdivision of the sensor fields depends on the analytics engine and cannot be modified by the user. The type "tt:CellLayout", which describes the cell structure, consists of the following three elements: "Columns", "Rows", and "Transformation". The first two parameters describe the number of cells in the X and Y directions. The first cell is located in the upper left-hand corner and the last cell in the lower right-hand corner. The parameter "Transformation" maps the frame coordinate system (see 9.2.2.3) to the cell grid. The origin of the new coordinate system is the upper left-hand corner of the upper left-hand cell. The unit dimension in the X direction corresponds to the cell width, and the unit dimension in the Y direction corresponds to the cell height.

D.2.2 Module configuration

Cell motion analytics engine module configuration is described by the analytics module description language:

```
<tt:AnalyticsModuleDescription Name="tt:CellMotionEngine">
  <tt:Parameters>
    <tt:SimpleItemDescription Name="Sensitivity"
                              Type="xs:integer"/>
    <tt:ElementItemDescription Name="Layout" Type="tt:CellLayout"/>
  </tt:Parameters>
</tt:AnalyticsModuleDescription>
```

Table D.3 – Module configuration parameters

Parameter name	Description
Sensitivity	The lower the value of the sensitivity is set, the higher the variations of the luminance values need to be in order to be identified as motion. Hence, a low sensitivity can be used to suppress estimated motion due to noise. Range is 0 to 100
Layout	The layout is typically defined by the hardware of the motion detector. It is fixed and cannot be altered. The layout structure defines the size of the cell array as well as their mapping to the video image.

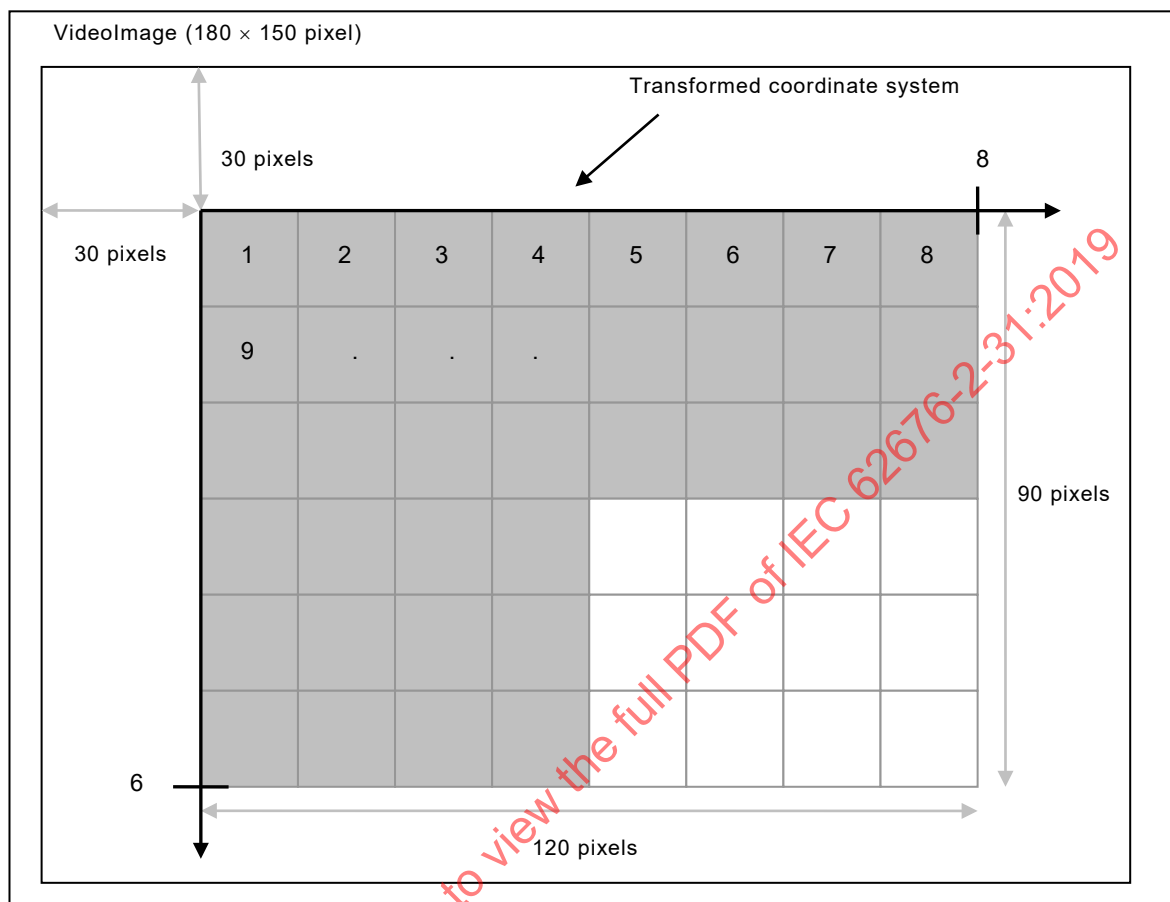
The definition uses tt:CellLayout data structure (see also Table D.4) defined as:

```
<xs:complexType name="CellLayout">
  <xs:sequence>
    <xs:element name="Transformation" type="tt:Transformation"/>
    <xs:any namespace="##any" processContents="lax"
              minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="Columns" type="xs:integer" use="required"/>
  <xs:attribute name="Rows" type="xs:integer" use="required"/>
  <xs:anyAttribute processContents="lax"/>
</xs:complexType>
```

Table D.4 – Description of CellLayout fields

Parameter name	Description
Columns	Number of columns of the cell grid (x dimension)
Rows	Number of rows of the cell grid (y dimension).
Transformation	Mapping of the cell grid to the video frame. The cell grid is starting from the upper left corner and x dimension is going from left to right and the y dimension from top to bottom.

An example of a configuration of a CellMotionEngine is shown in Figure D.1 where an 8 × 6-cell motion detector is depicted. The video image is assumed to have the size 180 pixels × 150 pixels. The cells have a dimension of 15 pixels × 15 pixels and the size of the working area of the cell motion detector is 120 pixels × 90 pixels.



IEC

Figure D.1 – CellLayout of an 8 × 6 CellMotionEngine

```
<tt:VideoAnalyticsConfiguration>
  <tt:AnalyticsEngineConfiguration>
    <tt:AnalyticsModule Name="MyCellMotion" Type="tt:CellMotionEngine">
      <tt:Parameters>
        <tt:SimpleItem Name="Sensitivity" Value="90"/>
        <tt:ElementItem Name="Layout">
          <tt:CellLayout Columns="8" Rows="6">
            <tt:Transformation>
              <tt:Translate x="-0.66666" y="0.6" />
              <tt:Scale x="0.1666666" y="-0.2" />
            </tt:Transformation>
          </tt:CellLayout>
        </tt:ElementItem>
      </tt:Parameters>
    </tt:AnalyticsModule>
  </tt:AnalyticsEngineConfiguration>
```

Annex E (normative)

Motion detection

The region motion detector detects any motion against the specified motion region. The rule is configured for an area (region of interest on the image source), which can be armed or disarmed. The region may be defined by a polygon structure, but cannot be defined by both. Although this rule is defined within the context of the analytics service, it is intended to allow configuration of hardware motion detection as opposed to motion detection from scene description data. For a description of the parameters, see Table E.1.

```
<tt:RuleDescription Name="tt:MotionRegionDetector">
  <tt:Parameters>
    <tt:ElementItemDescription Name="MotionRegion"
Type="axt:MotionRegionConfig"/>
  </tt:Parameters>
  <tt:MessageDescription IsProperty="true">
    <tt:Source>
      <tt:SimpleItemDescription Name="VideoSource" Type="tt:ReferenceToken"/>
      <tt:SimpleItemDescription Name="RuleName" Type="xs:string"/>
    </tt:Source>
    <tt:Key/>
    <tt>Data>
      <tt:SimpleItemDescription Name="State" Type="xs:boolean"/>
    </tt>Data>
  </tt:MessageDescription>
</tt:RuleDescription>
```

<tt:ParentTopic>tns1:RuleEngine/MotionRegionDetector/Motion</tt:ParentTopic>
</tt:MessageDescription>
</tt:RuleDescription>

The above rule description defines that a rule instance produces an event attached to the topic tns1:RuleEngine/MotionRegionDetector/Motion Motion with the state field defined in Table E.3. Calling the GetRuleOptions operation with the tt:MotionRegionDetector type will return a tt:MotionRegionConfigOptions with parameters defined in Table E.2.

Table E.1 – Motion Region Detector Rule configuration parameters

Parameter Name	Description
Armed	Indicates if the Motion Region is armed (detecting motion) or disarmed (motion is not being detected).
Rectangle	Provides the points of a rectangle that is specified within a region defined by the Bounds element of VideoSourceConfiguration.
Polygon	Provides the points of a polygon that is specified within a region defined by the Bounds element of VideoSourceConfiguration. If the device does not support polygons, the points in the polygon shall represent a rectangle.
Sensitivity	Indicates the sensitivity level of the motion detector for this region. The sensitivity value is normalized where 0 represents the lower sensitivity where significant motion is required to trigger an alarm and 1 represents the higher sensitivity where very little motion is required to trigger an alarm.

Table E.2 – Motion region detector rule configuration options

Parameter name	Description
MaxRegions	The total number of motion region detector rules that can be created on the device.
DisarmSupport	True if the device supports disarming a motion region detector rule.
PolygonSupport	True if the device supports defining a region using a polygon instead of a rectangle. The rectangle points are still passed using a Polygon element if the device does not support polygon regions. In this case, the points provided in the Polygon element shall represent a rectangle.
PolygonLimits	For devices that support polygons with limitations on the number of points, provides the minimum and maximum number of points that can be defined in Polygon.
RuleNotification	Indicates the device will supply the name of the rule that triggered the motion.
SingleSensitivitySupport	Indicates the device can only support one sensitivity level for all defined motion detection regions. Changing the sensitivity for one region would be applied to all regions.

Table E.3 – Description of the motion region detector event fields

Parameter name	Description
State	True (motion is detected) or False (motion is not detected)

IECNORM.COM : Click to view the full PDF of IEC 62676-2-31:2019

Annex F (normative)

Schema files

F.1 Device IO

```
<?xml version="1.0" encoding="utf-8"?><?xml-stylesheet type="text/xsl"
href="../../ver20/util/onvif-wsdl-viewer.xsl"?>
<wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap12/"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:tds="http://www.onvif.org/ver10/device/wsdl"
  xmlns:tmd="http://www.onvif.org/ver10/deviceIO/wsdl"
  targetNamespace="http://www.onvif.org/ver10/deviceIO/wsdl">
  <wsdl:import namespace="http://www.onvif.org/ver10/device/wsdl"
    location="../../ver10/device/wsdl/devicemgmt.wsdl"/>
  <wsdl:types>
    <xs:schema xmlns:tt="http://www.onvif.org/ver10/schema"
      targetNamespace="http://www.onvif.org/ver10/deviceIO/wsdl"
      elementFormDefault="qualified" version="16.12">
      <xs:import namespace="http://www.onvif.org/ver10/schema"
        schemaLocation="../../ver10/schema/onvif.xsd"/>
      <xs:element name="GetServiceCapabilities">
        <xs:complexType>
          <xs:sequence/>
        </xs:complexType>
      </xs:element>
      <xs:element name="GetServiceCapabilitiesResponse">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Capabilities" type="tmd:Capabilities"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="Capabilities">
        <xs:sequence>
          <xs:any namespace="##any" processContents="lax" minOccurs="0"
            maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="VideoSources" type="xs:int" default="0"/>
        <xs:attribute name="VideoOutputs" type="xs:int" default="0"/>
        <xs:attribute name="AudioSources" type="xs:int" default="0"/>
        <xs:attribute name="AudioOutputs" type="xs:int" default="0"/>
        <xs:attribute name="RelayOutputs" type="xs:int" default="0"/>
        <xs:attribute name="SerialPorts" type="xs:int" default="0"/>
        <xs:attribute name="DigitalInputs" type="xs:int" default="0"/>
        <xs:attribute name="DigitalInputOptions" type="xs:boolean" default="false"/>
        <xs:anyAttribute processContents="lax"/>
      </xs:complexType>
      <xs:element name="Capabilities" type="tmd:Capabilities"/>
      <xs:element name="GetRelayOutputOptions">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="RelayOutputToken" type="tt:ReferenceToken"
              minOccurs="0"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="GetRelayOutputOptionsResponse">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="RelayOutputOptions" type="tmd:RelayOutputOptions"
              minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="RelayOutputOptions">
        <xs:sequence>
          <xs:element name="Mode" type="tt:RelayMode" maxOccurs="unbounded"/>
          <xs:element name="DelayTimes" type="tmd:DelayTimes" minOccurs="0"/>
          <xs:element name="Discrete" type="xs:boolean" minOccurs="0"/>
          <xs:element name="Extension" type="tmd:RelayOutputOptionsExtension"
            minOccurs="0"/>
        </xs:sequence>
      </xs:complexType>
    </xs:schema>
  </wsdl:types>
</wsdl:definitions>
```

```

        <xs:attribute name="token" type="tt:ReferenceToken" use="required"/>
        <xs:anyAttribute processContents="lax"/>
    </xs:complexType>
    <xs:complexType name="RelayOutputOptionsExtension">
        <xs:sequence>
            <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax"/>
        </xs:sequence>
    </xs:complexType>
    <xs:simpleType name="DelayTimes">
        <xs:list itemType="xs:float"/>
    </xs:simpleType>
    <xs:complexType name="Get">
        <xs:sequence/>
    </xs:complexType>
    <xs:complexType name="GetResponse">
        <xs:sequence>
            <xs:element name="Token" type="tt:ReferenceToken" minOccurs="0"
maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
    <xs:element name="GetVideoSources" type="tmd:Get"/>
    <xs:element name="GetVideoSourcesResponse" type="tmd:GetResponse"/>
    <xs:element name="GetAudioSources" type="tmd:Get"/>
    <xs:element name="GetAudioSourcesResponse" type="tmd:GetResponse"/>
    <xs:element name="GetAudioOutputs" type="tmd:Get"/>
    <xs:element name="GetAudioOutputsResponse" type="tmd:GetResponse"/>
    <xs:element name="GetVideoOutputs">
        <xs:complexType>
            <xs:sequence/>
        </xs:complexType>
    </xs:element>
    <xs:element name="GetVideoOutputsResponse">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="VideoOutputs" type="tt:VideoOutput" minOccurs="0"
maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="GetAudioSourceConfiguration">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="AudioSourceToken" type="tt:ReferenceToken"/>
                <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="GetAudioSourceConfigurationResponse">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="AudioSourceConfiguration"
type="tt:AudioSourceConfiguration"/>
                <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="GetAudioOutputConfiguration">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="AudioOutputToken" type="tt:ReferenceToken"/>
                <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="GetAudioOutputConfigurationResponse">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="AudioOutputConfiguration"
type="tt:AudioOutputConfiguration"/>
                <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>

```



```

<xs:element name="GetVideoSourceConfiguration">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="VideoSourceToken" type="tt:ReferenceToken"/>
      <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="GetVideoSourceConfigurationResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="VideoSourceConfiguration"
type="tt:VideoSourceConfiguration"/>
      <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="GetVideoOutputConfiguration">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="VideoOutputToken" type="tt:ReferenceToken"/>
      <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="GetVideoOutputConfigurationResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="VideoOutputConfiguration"
type="tt:VideoOutputConfiguration"/>
      <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="SetAudioSourceConfiguration">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Configuration" type="tt:AudioSourceConfiguration"/>
      <xs:element name="ForcePersistence" type="xs:boolean"/>
      <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="SetAudioSourceConfigurationResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="SetAudioOutputConfiguration">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Configuration" type="tt:AudioOutputConfiguration"/>
      <xs:element name="ForcePersistence" type="xs:boolean"/>
      <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="SetAudioOutputConfigurationResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="SetVideoSourceConfiguration">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Configuration" type="tt:VideoSourceConfiguration"/>

```

```

        <xs:element name="ForcePersistence" type="xs:boolean"/>
        <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax"/>
    </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="SetVideoSourceConfigurationResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="SetVideoOutputConfiguration">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Configuration" type="tt:VideoOutputConfiguration"/>
            <xs:element name="ForcePersistence" type="xs:boolean"/>
            <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="SetVideoOutputConfigurationResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="GetVideoSourceConfigurationOptions">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="VideoSourceToken" type="tt:ReferenceToken"/>
            <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="GetVideoSourceConfigurationOptionsResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="VideoSourceConfigurationOptions"
type="tt:VideoSourceConfigurationOptions"/>
            <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="GetVideoOutputConfigurationOptions">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="VideoOutputToken" type="tt:ReferenceToken"/>
            <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="GetVideoOutputConfigurationOptionsResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="VideoOutputConfigurationOptions"
type="tt:VideoOutputConfigurationOptions"/>
            <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="GetAudioSourceConfigurationOptions">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="AudioSourceToken" type="tt:ReferenceToken"/>
            <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax"/>
        </xs:sequence>
    </xs:complexType>

```

```

</xs:element>
<xs:element name="GetAudioSourceConfigurationOptionsResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="AudioSourceOptions"
type="tt:AudioSourceConfigurationOptions"/>
      <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="GetAudioOutputConfigurationOptions">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="AudioOutputToken" type="tt:ReferenceToken"/>
      <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="GetAudioOutputConfigurationOptionsResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="AudioOutputOptions"
type="tt:AudioOutputConfigurationOptions"/>
      <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="SetRelayOutputSettings">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="RelayOutput" type="tt:RelayOutput"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="SetRelayOutputSettingsResponse">
  <xs:complexType>
    <xs:sequence/>
  </xs:complexType>
</xs:element>
<xs:element name="GetDigitalInputs">
  <xs:complexType>
    <xs:sequence/>
  </xs:complexType>
</xs:element>
<xs:element name="GetDigitalInputsResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="DigitalInputs" type="tt:DigitalInput" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:complexType name="DigitalInputConfigurationInputOptions">
  <xs:sequence>
    <xs:element name="IdleState" type="tt:DigitalIdleState"
maxOccurs="unbounded"/>
    <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:element name="GetDigitalInputConfigurationOptions">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Token" type="tt:ReferenceToken" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="GetDigitalInputConfigurationOptionsResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="DigitalInputOptions"
type="tmd:DigitalInputConfigurationInputOptions"/>
    </xs:sequence>
  </xs:complexType>

```

```

</xs:element>
<xs:element name="SetDigitalInputConfigurations">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="DigitalInputs" type="tt:DigitalInput"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="SetDigitalInputConfigurationsResponse">
  <xs:complexType>
    <xs:sequence/>
  </xs:complexType>
</xs:element>
<xs:element name="GetSerialPorts">
  <xs:complexType>
    <xs:sequence/>
  </xs:complexType>
</xs:element>
<xs:element name="GetSerialPortsResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="SerialPort" type="tmd:SerialPort" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="GetSerialPortConfiguration">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="SerialPortToken" type="tt:ReferenceToken"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="GetSerialPortConfigurationResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="SerialPortConfiguration"
type="tmd:SerialPortConfiguration"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="SetSerialPortConfiguration">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="SerialPortConfiguration"
type="tmd:SerialPortConfiguration"/>
      <xs:element name="ForcePersistence" type="xs:boolean"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="SetSerialPortConfigurationResponse">
  <xs:complexType>
    <xs:sequence/>
  </xs:complexType>
</xs:element>
<xs:element name="GetSerialPortConfigurationOptions">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="SerialPortToken" type="tt:ReferenceToken"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="GetSerialPortConfigurationOptionsResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="SerialPortOptions"
type="tmd:SerialPortConfigurationOptions"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="SendReceiveSerialCommand">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="SerialData" type="tmd:SerialData" minOccurs="0"/>
      <xs:element name="TimeOut" type="xs:duration" minOccurs="0"/>
      <xs:element name="DataLength" type="xs:integer" minOccurs="0"/>
      <xs:element name="Delimiter" type="xs:string" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

```

        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="SendReceiveSerialCommandResponse">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="SerialData" type="tmd:SerialData" minOccurs="0"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:complexType name="SerialData">
      <xs:choice>
        <xs:element name="Binary" type="xs:base64Binary"/>
        <xs:element name="String" type="xs:string"/>
      </xs:choice>
      <xs:anyAttribute processContents="lax"/>
    </xs:complexType>
    <xs:complexType name="SerialPort">
      <xs:complexContent>
        <xs:extension base="tt:DeviceEntity">
          <xs:sequence>
            <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:anyAttribute processContents="lax"/>
        </xs:extension>
      </xs:complexContent>
    </xs:complexType>
    <xs:simpleType name="SerialPortType">
      <xs:restriction base="xs:string">
        <xs:enumeration value="RS232"/>
        <xs:enumeration value="RS422HalfDuplex"/>
        <xs:enumeration value="RS422FullDuplex"/>
        <xs:enumeration value="RS485HalfDuplex"/>
        <xs:enumeration value="RS485FullDuplex"/>
        <xs:enumeration value="Generic"/>
      </xs:restriction>
    </xs:simpleType>
    <xs:complexType name="SerialPortConfiguration">
      <xs:sequence>
        <xs:element name="BaudRate" type="xs:int"/>
        <xs:element name="ParityBit" type="tmd:ParityBit"/>
        <xs:element name="CharacterLength" type="xs:int"/>
        <xs:element name="StopBit" type="xs:float"/>
        <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attribute name="token" type="tt:ReferenceToken" use="required"/>
      <xs:attribute name="type" type="tmd:SerialPortType" use="required"/>
      <xs:anyAttribute processContents="lax"/>
    </xs:complexType>
    <xs:simpleType name="ParityBit">
      <xs:restriction base="xs:string">
        <xs:enumeration value="None"/>
        <xs:enumeration value="Even"/>
        <xs:enumeration value="Odd"/>
        <xs:enumeration value="Mark"/>
        <xs:enumeration value="Space"/>
        <xs:enumeration value="Extended"/>
      </xs:restriction>
    </xs:simpleType>
    <xs:complexType name="SerialPortConfigurationOptions">
      <xs:sequence>
        <xs:element name="BaudRateList" type="tt:IntList"/>
        <xs:element name="ParityBitList" type="tmd:ParityBitList"/>
        <xs:element name="CharacterLengthList" type="tt:IntList"/>
        <xs:element name="StopBitList" type="tt:FloatList"/>
        <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attribute name="token" type="tt:ReferenceToken" use="required"/>
      <xs:anyAttribute processContents="lax"/>
    </xs:complexType>
    <xs:complexType name="ParityBitList">
      <xs:sequence>
        <xs:element name="Items" type="tmd:ParityBit" minOccurs="0"
maxOccurs="unbounded"/>
      </xs:sequence>

```

```

    </xs:complexType>
  </xs:schema>
</wsdl:types>
<wsdl:message name="GetServiceCapabilitiesRequest">
  <wsdl:part name="parameters" element="tmd:GetServiceCapabilities"/>
</wsdl:message>
<wsdl:message name="GetServiceCapabilitiesResponse">
  <wsdl:part name="parameters" element="tmd:GetServiceCapabilitiesResponse"/>
</wsdl:message>
<wsdl:message name="GetRelayOutputOptionsRequest">
  <wsdl:part name="parameters" element="tmd:GetRelayOutputOptions"/>
</wsdl:message>
<wsdl:message name="GetRelayOutputOptionsResponse">
  <wsdl:part name="parameters" element="tmd:GetRelayOutputOptionsResponse"/>
</wsdl:message>
<wsdl:message name="GetVideoOutputsRequest">
  <wsdl:part name="parameters" element="tmd:GetVideoOutputs"/>
</wsdl:message>
<wsdl:message name="GetVideoOutputsResponse">
  <wsdl:part name="parameters" element="tmd:GetVideoOutputsResponse"/>
</wsdl:message>
<wsdl:message name="GetAudioOutputsRequest">
  <wsdl:part name="parameters" element="tmd:GetAudioOutputs"/>
</wsdl:message>
<wsdl:message name="GetAudioOutputsResponse">
  <wsdl:part name="parameters" element="tmd:GetAudioOutputsResponse"/>
</wsdl:message>
<wsdl:message name="GetVideoSourcesRequest">
  <wsdl:part name="parameters" element="tmd:GetVideoSources"/>
</wsdl:message>
<wsdl:message name="GetVideoSourcesResponse">
  <wsdl:part name="parameters" element="tmd:GetVideoSourcesResponse"/>
</wsdl:message>
<wsdl:message name="GetAudioSourcesRequest">
  <wsdl:part name="parameters" element="tmd:GetAudioSources"/>
</wsdl:message>
<wsdl:message name="GetAudioSourcesResponse">
  <wsdl:part name="parameters" element="tmd:GetAudioSourcesResponse"/>
</wsdl:message>
<wsdl:message name="GetVideoSourceConfigurationRequest">
  <wsdl:part name="parameters" element="tmd:GetVideoSourceConfiguration"/>
</wsdl:message>
<wsdl:message name="GetVideoSourceConfigurationResponse">
  <wsdl:part name="parameters" element="tmd:GetVideoSourceConfigurationResponse"/>
</wsdl:message>
<wsdl:message name="GetVideoOutputConfigurationRequest">
  <wsdl:part name="parameters" element="tmd:GetVideoOutputConfiguration"/>
</wsdl:message>
<wsdl:message name="GetVideoOutputConfigurationResponse">
  <wsdl:part name="parameters" element="tmd:GetVideoOutputConfigurationResponse"/>
</wsdl:message>
<wsdl:message name="GetAudioSourceConfigurationRequest">
  <wsdl:part name="parameters" element="tmd:GetAudioSourceConfiguration"/>
</wsdl:message>
<wsdl:message name="GetAudioSourceConfigurationResponse">
  <wsdl:part name="parameters" element="tmd:GetAudioSourceConfigurationResponse"/>
</wsdl:message>
<wsdl:message name="GetAudioOutputConfigurationRequest">
  <wsdl:part name="parameters" element="tmd:GetAudioOutputConfiguration"/>
</wsdl:message>
<wsdl:message name="GetAudioOutputConfigurationResponse">
  <wsdl:part name="parameters" element="tmd:GetAudioOutputConfigurationResponse"/>
</wsdl:message>
<wsdl:message name="SetVideoSourceConfigurationRequest">
  <wsdl:part name="parameters" element="tmd:SetVideoSourceConfiguration"/>
</wsdl:message>
<wsdl:message name="SetVideoSourceConfigurationResponse">
  <wsdl:part name="parameters" element="tmd:SetVideoSourceConfigurationResponse"/>
</wsdl:message>
<wsdl:message name="SetVideoOutputConfigurationRequest">
  <wsdl:part name="parameters" element="tmd:SetVideoOutputConfiguration"/>
</wsdl:message>
<wsdl:message name="SetVideoOutputConfigurationResponse">
  <wsdl:part name="parameters" element="tmd:SetVideoOutputConfigurationResponse"/>
</wsdl:message>
<wsdl:message name="SetAudioSourceConfigurationRequest">
  <wsdl:part name="parameters" element="tmd:SetAudioSourceConfiguration"/>
</wsdl:message>

```

```
<wsdl:message name="SetAudioSourceConfigurationResponse">
  <wsdl:part name="parameters" element="tmd:SetAudioSourceConfigurationResponse"/>
</wsdl:message>
<wsdl:message name="SetAudioOutputConfigurationRequest">
  <wsdl:part name="parameters" element="tmd:SetAudioOutputConfiguration"/>
</wsdl:message>
<wsdl:message name="SetAudioOutputConfigurationResponse">
  <wsdl:part name="parameters" element="tmd:SetAudioOutputConfigurationResponse"/>
</wsdl:message>
<wsdl:message name="GetVideoSourceConfigurationOptionsRequest">
  <wsdl:part name="parameters" element="tmd:GetVideoSourceConfigurationOptions"/>
</wsdl:message>
<wsdl:message name="GetVideoSourceConfigurationOptionsResponse">
  <wsdl:part name="parameters"
element="tmd:GetVideoSourceConfigurationOptionsResponse"/>
</wsdl:message>
<wsdl:message name="GetVideoOutputConfigurationOptionsRequest">
  <wsdl:part name="parameters" element="tmd:GetVideoOutputConfigurationOptions"/>
</wsdl:message>
<wsdl:message name="GetVideoOutputConfigurationOptionsResponse">
  <wsdl:part name="parameters"
element="tmd:GetVideoOutputConfigurationOptionsResponse"/>
</wsdl:message>
<wsdl:message name="GetAudioSourceConfigurationOptionsRequest">
  <wsdl:part name="parameters" element="tmd:GetAudioSourceConfigurationOptions"/>
</wsdl:message>
<wsdl:message name="GetAudioSourceConfigurationOptionsResponse">
  <wsdl:part name="parameters"
element="tmd:GetAudioSourceConfigurationOptionsResponse"/>
</wsdl:message>
<wsdl:message name="GetAudioOutputConfigurationOptionsRequest">
  <wsdl:part name="parameters" element="tmd:GetAudioOutputConfigurationOptions"/>
</wsdl:message>
<wsdl:message name="GetAudioOutputConfigurationOptionsResponse">
  <wsdl:part name="parameters"
element="tmd:GetAudioOutputConfigurationOptionsResponse"/>
</wsdl:message>
<wsdl:message name="GetRelayOutputsRequest">
  <wsdl:part name="parameters" element="tds:GetRelayOutputs"/>
</wsdl:message>
<wsdl:message name="GetRelayOutputsResponse">
  <wsdl:part name="parameters" element="tds:GetRelayOutputsResponse"/>
</wsdl:message>
<wsdl:message name="SetRelayOutputSettingsRequest">
  <wsdl:part name="parameters" element="tmd:SetRelayOutputSettings"/>
</wsdl:message>
<wsdl:message name="SetRelayOutputSettingsResponse">
  <wsdl:part name="parameters" element="tmd:SetRelayOutputSettingsResponse"/>
</wsdl:message>
<wsdl:message name="SetRelayOutputStateRequest">
  <wsdl:part name="parameters" element="tds:SetRelayOutputState"/>
</wsdl:message>
<wsdl:message name="SetRelayOutputStateResponse">
  <wsdl:part name="parameters" element="tds:SetRelayOutputStateResponse"/>
</wsdl:message>
<wsdl:message name="GetDigitalInputsRequest">
  <wsdl:part name="parameters" element="tmd:GetDigitalInputs"/>
</wsdl:message>
<wsdl:message name="GetDigitalInputsResponse">
  <wsdl:part name="parameters" element="tmd:GetDigitalInputsResponse"/>
</wsdl:message>
<wsdl:message name="GetDigitalInputConfigurationOptionsRequest">
  <wsdl:part name="parameters" element="tmd:GetDigitalInputConfigurationOptions"/>
</wsdl:message>
<wsdl:message name="GetDigitalInputConfigurationOptionsResponse">
  <wsdl:part name="parameters"
element="tmd:GetDigitalInputConfigurationOptionsResponse"/>
</wsdl:message>
<wsdl:message name="SetDigitalInputConfigurationsRequest">
  <wsdl:part name="parameters" element="tmd:SetDigitalInputConfigurations"/>
</wsdl:message>
<wsdl:message name="SetDigitalInputConfigurationsResponse">
  <wsdl:part name="parameters" element="tmd:SetDigitalInputConfigurationsResponse"/>
</wsdl:message>
<wsdl:message name="GetSerialPortsRequest">
  <wsdl:part name="parameters" element="tmd:GetSerialPorts"/>
</wsdl:message>
<wsdl:message name="GetSerialPortsResponse">
```



```

    <wsdl:part name="parameters" element="tmd:GetSerialPortsResponse"/>
  </wsdl:message>
  <wsdl:message name="GetSerialPortConfigurationRequest">
    <wsdl:part name="parameters" element="tmd:GetSerialPortConfiguration"/>
  </wsdl:message>
  <wsdl:message name="GetSerialPortConfigurationResponse">
    <wsdl:part name="parameters" element="tmd:GetSerialPortConfigurationResponse"/>
  </wsdl:message>
  <wsdl:message name="SetSerialPortConfigurationRequest">
    <wsdl:part name="parameters" element="tmd:SetSerialPortConfiguration"/>
  </wsdl:message>
  <wsdl:message name="SetSerialPortConfigurationResponse">
    <wsdl:part name="parameters" element="tmd:SetSerialPortConfigurationResponse"/>
  </wsdl:message>
  <wsdl:message name="GetSerialPortConfigurationOptionsRequest">
    <wsdl:part name="parameters" element="tmd:GetSerialPortConfigurationOptions"/>
  </wsdl:message>
  <wsdl:message name="GetSerialPortConfigurationOptionsResponse">
    <wsdl:part name="parameters"
element="tmd:GetSerialPortConfigurationOptionsResponse"/>
  </wsdl:message>
  <wsdl:message name="SendReceiveSerialCommandRequest">
    <wsdl:part name="parameters" element="tmd:SendReceiveSerialCommand"/>
  </wsdl:message>
  <wsdl:message name="SendReceiveSerialCommandResponse">
    <wsdl:part name="parameters" element="tmd:SendReceiveSerialCommandResponse"/>
  </wsdl:message>
  <wsdl:portType name="DeviceIOPort">
    <wsdl:operation name="GetServiceCapabilities">
      <wsdl:input message="tmd:GetServiceCapabilitiesRequest"/>
      <wsdl:output message="tmd:GetServiceCapabilitiesResponse"/>
    </wsdl:operation>
    <wsdl:operation name="GetRelayOutputOptions">
      <wsdl:input message="tmd:GetRelayOutputOptionsRequest"/>
      <wsdl:output message="tmd:GetRelayOutputOptionsResponse"/>
    </wsdl:operation>
    <wsdl:operation name="GetAudioSources">
      <wsdl:input message="tmd:GetAudioSourcesRequest"/>
      <wsdl:output message="tmd:GetAudioSourcesResponse"/>
    </wsdl:operation>
    <wsdl:operation name="GetAudioOutputs">
      <wsdl:input message="tmd:GetAudioOutputsRequest"/>
      <wsdl:output message="tmd:GetAudioOutputsResponse"/>
    </wsdl:operation>
    <wsdl:operation name="GetVideoSources">
      <wsdl:input message="tmd:GetVideoSourcesRequest"/>
      <wsdl:output message="tmd:GetVideoSourcesResponse"/>
    </wsdl:operation>
    <wsdl:operation name="GetVideoOutputs">
      <wsdl:input message="tmd:GetVideoOutputsRequest"/>
      <wsdl:output message="tmd:GetVideoOutputsResponse"/>
    </wsdl:operation>
    <wsdl:operation name="GetVideoSourceConfiguration">
      <wsdl:input message="tmd:GetVideoSourceConfigurationRequest"/>
      <wsdl:output message="tmd:GetVideoSourceConfigurationResponse"/>
    </wsdl:operation>
    <wsdl:operation name="GetVideoOutputConfiguration">
      <wsdl:input message="tmd:GetVideoOutputConfigurationRequest"/>
      <wsdl:output message="tmd:GetVideoOutputConfigurationResponse"/>
    </wsdl:operation>
    <wsdl:operation name="GetAudioSourceConfiguration">
      <wsdl:input message="tmd:GetAudioSourceConfigurationRequest"/>
      <wsdl:output message="tmd:GetAudioSourceConfigurationResponse"/>
    </wsdl:operation>
    <wsdl:operation name="GetAudioOutputConfiguration">
      <wsdl:input message="tmd:GetAudioOutputConfigurationRequest"/>
      <wsdl:output message="tmd:GetAudioOutputConfigurationResponse"/>
    </wsdl:operation>
    <wsdl:operation name="SetVideoSourceConfiguration">
      <wsdl:input message="tmd:SetVideoSourceConfigurationRequest"/>
      <wsdl:output message="tmd:SetVideoSourceConfigurationResponse"/>
    </wsdl:operation>
    <wsdl:operation name="SetVideoOutputConfiguration">
      <wsdl:input message="tmd:SetVideoOutputConfigurationRequest"/>
      <wsdl:output message="tmd:SetVideoOutputConfigurationResponse"/>
    </wsdl:operation>
    <wsdl:operation name="SetAudioSourceConfiguration">
      <wsdl:input message="tmd:SetAudioSourceConfigurationRequest"/>

```



```

        <wsdl:output message="tmd:SetAudioSourceConfigurationResponse"/>
    </wsdl:operation>
    <wsdl:operation name="SetAudioOutputConfiguration">
        <wsdl:input message="tmd:SetAudioOutputConfigurationRequest"/>
        <wsdl:output message="tmd:SetAudioOutputConfigurationResponse"/>
    </wsdl:operation>
    <wsdl:operation name="GetVideoSourceConfigurationOptions">
        <wsdl:input message="tmd:GetVideoSourceConfigurationOptionsRequest"/>
        <wsdl:output message="tmd:GetVideoSourceConfigurationOptionsResponse"/>
    </wsdl:operation>
    <wsdl:operation name="GetVideoOutputConfigurationOptions">
        <wsdl:input message="tmd:GetVideoOutputConfigurationOptionsRequest"/>
        <wsdl:output message="tmd:GetVideoOutputConfigurationOptionsResponse"/>
    </wsdl:operation>
    <wsdl:operation name="GetAudioSourceConfigurationOptions">
        <wsdl:input message="tmd:GetAudioSourceConfigurationOptionsRequest"/>
        <wsdl:output message="tmd:GetAudioSourceConfigurationOptionsResponse"/>
    </wsdl:operation>
    <wsdl:operation name="GetAudioOutputConfigurationOptions">
        <wsdl:input message="tmd:GetAudioOutputConfigurationOptionsRequest"/>
        <wsdl:output message="tmd:GetAudioOutputConfigurationOptionsResponse"/>
    </wsdl:operation>
    <wsdl:operation name="GetRelayOutputs">
        <wsdl:input message="tmd:GetRelayOutputsRequest"/>
        <wsdl:output message="tmd:GetRelayOutputsResponse"/>
    </wsdl:operation>
    <wsdl:operation name="SetRelayOutputSettings">
        <wsdl:input message="tmd:SetRelayOutputSettingsRequest"/>
        <wsdl:output message="tmd:SetRelayOutputSettingsResponse"/>
    </wsdl:operation>
    <wsdl:operation name="SetRelayOutputState">
        <wsdl:input message="tmd:SetRelayOutputStateRequest"/>
        <wsdl:output message="tmd:SetRelayOutputStateResponse"/>
    </wsdl:operation>
    <wsdl:operation name="GetDigitalInputs">
        <wsdl:input message="tmd:GetDigitalInputsRequest"/>
        <wsdl:output message="tmd:GetDigitalInputsResponse"/>
    </wsdl:operation>
    <wsdl:operation name="GetDigitalInputConfigurationOptions">
        <wsdl:input message="tmd:GetDigitalInputConfigurationOptionsRequest"/>
        <wsdl:output message="tmd:GetDigitalInputConfigurationOptionsResponse"/>
    </wsdl:operation>
    <wsdl:operation name="SetDigitalInputConfigurations">
        <wsdl:input message="tmd:SetDigitalInputConfigurationsRequest"/>
        <wsdl:output message="tmd:SetDigitalInputConfigurationsResponse"/>
    </wsdl:operation>
    <wsdl:operation name="GetSerialPorts">
        <wsdl:input message="tmd:GetSerialPortsRequest"/>
        <wsdl:output message="tmd:GetSerialPortsResponse"/>
    </wsdl:operation>
    <wsdl:operation name="GetSerialPortConfiguration">
        <wsdl:input message="tmd:GetSerialPortConfigurationRequest"/>
        <wsdl:output message="tmd:GetSerialPortConfigurationResponse"/>
    </wsdl:operation>
    <wsdl:operation name="SetSerialPortConfiguration">
        <wsdl:input message="tmd:SetSerialPortConfigurationRequest"/>
        <wsdl:output message="tmd:SetSerialPortConfigurationResponse"/>
    </wsdl:operation>
    <wsdl:operation name="GetSerialPortConfigurationOptions">
        <wsdl:input message="tmd:GetSerialPortConfigurationOptionsRequest"/>
        <wsdl:output message="tmd:GetSerialPortConfigurationOptionsResponse"/>
    </wsdl:operation>
    <wsdl:operation name="SendReceiveSerialCommand">
        <wsdl:input message="tmd:SendReceiveSerialCommandRequest"/>
        <wsdl:output message="tmd:SendReceiveSerialCommandResponse"/>
    </wsdl:operation>
</wsdl:portType>
<wsdl:binding name="DeviceIOBinding" type="tmd:DeviceIOPort">
    <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
    <wsdl:operation name="GetServiceCapabilities">
        <soap:operation
soapAction="http://www.onvif.org/ver10/deviceio/wsdl/GetServiceCapabilities"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>

```

```

</wsdl:operation>
<wsdl:operation name="GetRelayOutputOptions">
  <soap:operation
soapAction="http://www.onvif.org/ver10/deviceio/wsdl/GetRelayOutputOptions"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
<wsdl:operation name="GetAudioSources">
  <soap:operation
soapAction="http://www.onvif.org/ver10/deviceio/wsdl/GetAudioSources"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
<wsdl:operation name="GetAudioOutputs">
  <soap:operation
soapAction="http://www.onvif.org/ver10/deviceio/wsdl/GetAudioOutputs"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
<wsdl:operation name="GetVideoSources">
  <soap:operation
soapAction="http://www.onvif.org/ver10/deviceio/wsdl/GetVideoSources"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
<wsdl:operation name="GetVideoOutputs">
  <soap:operation
soapAction="http://www.onvif.org/ver10/deviceio/wsdl/GetVideoOutputs"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
<wsdl:operation name="GetVideoSourceConfiguration">
  <soap:operation
soapAction="http://www.onvif.org/ver10/deviceio/wsdl/GetVideoSourceConfiguration"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
<wsdl:operation name="GetVideoOutputConfiguration">
  <soap:operation
soapAction="http://www.onvif.org/ver10/deviceio/wsdl/GetVideoOutputConfiguration"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
<wsdl:operation name="GetAudioSourceConfiguration">
  <soap:operation
soapAction="http://www.onvif.org/ver10/deviceio/wsdl/GetAudioSourceConfiguration"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>

```

```

        <soap:body use="literal"/>
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetAudioOutputConfiguration">
    <soap:operation
soapAction="http://www.onvif.org/ver10/deviceio/wsdl/GetAudioOutputConfiguration"/>
    <wsdl:input>
        <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
        <soap:body use="literal"/>
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="SetVideoSourceConfiguration">
    <soap:operation
soapAction="http://www.onvif.org/ver10/deviceio/wsdl/SetVideoSourceConfiguration"/>
    <wsdl:input>
        <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
        <soap:body use="literal"/>
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="SetVideoOutputConfiguration">
    <soap:operation
soapAction="http://www.onvif.org/ver10/deviceio/wsdl/SetVideoOutputConfiguration"/>
    <wsdl:input>
        <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
        <soap:body use="literal"/>
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="SetAudioSourceConfiguration">
    <soap:operation
soapAction="http://www.onvif.org/ver10/deviceio/wsdl/SetAudioSourceConfiguration"/>
    <wsdl:input>
        <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
        <soap:body use="literal"/>
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="SetAudioOutputConfiguration">
    <soap:operation
soapAction="http://www.onvif.org/ver10/deviceio/wsdl/SetAudioOutputConfiguration"/>
    <wsdl:input>
        <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
        <soap:body use="literal"/>
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetVideoSourceConfigurationOptions">
    <soap:operation
soapAction="http://www.onvif.org/ver10/deviceio/wsdl/GetVideoSourceConfigurationOptions
"/>
    <wsdl:input>
        <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
        <soap:body use="literal"/>
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetVideoOutputConfigurationOptions">
    <soap:operation
soapAction="http://www.onvif.org/ver10/deviceio/wsdl/GetVideoOutputConfigurationOptions
"/>
    <wsdl:input>
        <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
        <soap:body use="literal"/>
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetAudioSourceConfigurationOptions">
    <soap:operation
soapAction="http://www.onvif.org/ver10/deviceio/wsdl/GetAudioSourceConfigurationOptions

```

```

"/>
    <wsdl:input>
        <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
        <soap:body use="literal"/>
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetAudioOutputConfigurationOptions">
    <soap:operation
soapAction="http://www.onvif.org/ver10/deviceio/wsdl/GetAudioOutputConfigurationOptions
"/>
    <wsdl:input>
        <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
        <soap:body use="literal"/>
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetRelayOutputs">
    <soap:operation
soapAction="http://www.onvif.org/ver10/deviceio/wsdl/GetRelayOutputs"/>
    <wsdl:input>
        <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
        <soap:body use="literal"/>
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="SetRelayOutputSettings">
    <soap:operation
soapAction="http://www.onvif.org/ver10/deviceio/wsdl/SetRelayOutputSettings"/>
    <wsdl:input>
        <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
        <soap:body use="literal"/>
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="SetRelayOutputState">
    <soap:operation
soapAction="http://www.onvif.org/ver10/deviceio/wsdl/SetRelayOutputState"/>
    <wsdl:input>
        <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
        <soap:body use="literal"/>
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetDigitalInputs">
    <soap:operation
soapAction="http://www.onvif.org/ver10/deviceio/wsdl/GetDigitalInputs"/>
    <wsdl:input>
        <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
        <soap:body use="literal"/>
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetDigitalInputConfigurationOptions">
    <soap:operation
soapAction="http://www.onvif.org/ver10/deviceio/wsdl/GetDigitalInputConfigurationOption
s"/>
    <wsdl:input>
        <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
        <soap:body use="literal"/>
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="SetDigitalInputConfigurations">
    <soap:operation
soapAction="http://www.onvif.org/ver10/deviceio/wsdl/SetDigitalInputConfigurations"/>
    <wsdl:input>
        <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
        <soap:body use="literal"/>
    </wsdl:output>
    <soap:body use="literal"/>

```

```
</wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetSerialPorts">
  <soap:operation
soapAction="http://www.onvif.org/ver10/deviceio/wsdl/GetSerialPorts"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetSerialPortConfiguration">
  <soap:operation
soapAction="http://www.onvif.org/ver10/deviceio/wsdl/GetSerialPortConfigurations"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="SetSerialPortConfiguration">
  <soap:operation
soapAction="http://www.onvif.org/ver10/deviceio/wsdl/SetSerialPortConfiguration"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetSerialPortConfigurationOptions">
  <soap:operation
soapAction="http://www.onvif.org/ver10/deviceio/wsdl/GetSerialPortConfigurationOptions"
/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="SendReceiveSerialCommand">
  <soap:operation
soapAction="http://www.onvif.org/ver10/deviceio/wsdl/SendReceiveSerialCommand"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
</wsdl:binding>
</wsdl:definitions>
```

F.2 Imaging

```
<?xml version="1.0" encoding="utf-8"?><?xml-stylesheet type="text/xsl"
href="../../../ver20/util/onvif-wsdl-viewer.xsl"?><wsdl:definitions
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:timg="http://www.onvif.org/ver20/imaging/wsdl"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap12/"
xmlns:tt="http://www.onvif.org/ver10/schema" name="ImagingService"
targetNamespace="http://www.onvif.org/ver20/imaging/wsdl">
  <wsdl:types>
    <xs:schema targetNamespace="http://www.onvif.org/ver20/imaging/wsdl"
elementFormDefault="qualified" version="16.06">
      <xs:import namespace="http://www.onvif.org/ver10/schema"
schemaLocation="../../../ver10/schema/onvif.xsd"/>
      <xs:element name="GetServiceCapabilities">
        <xs:complexType>
          <xs:sequence/>
        </xs:complexType>
      </xs:element>
      <xs:element name="GetServiceCapabilitiesResponse">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Capabilities" type="timg:Capabilities"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="Capabilities">
        <xs:sequence>
          <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="ImageStabilization" type="xs:boolean"/>
        <xs:attribute name="Presets" type="xs:boolean"/>
        <xs:anyAttribute processContents="lax"/>
      </xs:complexType>
      <xs:element name="Capabilities" type="timg:Capabilities"/>
      <xs:element name="GetImagingSettings">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="VideoSourceToken" type="tt:ReferenceToken"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="GetImagingSettingsResponse">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="ImagingSettings" type="tt:ImagingSettings20"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="SetImagingSettings">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="VideoSourceToken" type="tt:ReferenceToken"/>
            <xs:element name="ImagingSettings" type="tt:ImagingSettings20"/>
            <xs:element name="ForcePersistence" type="xs:boolean" maxOccurs="1"
minOccurs="0"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="SetImagingSettingsResponse">
        <xs:complexType/>
      </xs:element>
      <xs:element name="GetOptions">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="VideoSourceToken" type="tt:ReferenceToken"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="GetOptionsResponse">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="ImagingOptions" type="tt:ImagingOptions20"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:schema>
  </wsdl:types>
</wsdl:definitions>
```

```

    </xs:complexType>
  </xs:element>
  <xs:element name="Move">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="VideoSourceToken" type="tt:ReferenceToken"/>
        <xs:element name="Focus" type="tt:FocusMove"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="MoveResponse">
    <xs:complexType>
      <xs:sequence/>
    </xs:complexType>
  </xs:element>
  <xs:element name="GetMoveOptions">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="VideoSourceToken" type="tt:ReferenceToken"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="GetMoveOptionsResponse">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="MoveOptions" type="tt:MoveOptions20"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="Stop">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="VideoSourceToken" type="tt:ReferenceToken"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="StopResponse">
    <xs:complexType>
      <xs:sequence/>
    </xs:complexType>
  </xs:element>
  <xs:element name="GetStatus">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="VideoSourceToken" type="tt:ReferenceToken"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="GetStatusResponse">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Status" type="tt:ImagingStatus20"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:simpleType name="ImagingPresetType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="Custom"/>
      <xs:enumeration value="ClearWeather"/>
      <xs:enumeration value="Cloudy"/>
      <xs:enumeration value="Fog"/>
      <xs:enumeration value="Rain"/>
      <xs:enumeration value="Snowing"/>
      <xs:enumeration value="Snow"/>
      <xs:enumeration value="WDR"/>
      <xs:enumeration value="Shade"/>
      <xs:enumeration value="Night"/>
      <xs:enumeration value="Indoor"/>
      <xs:enumeration value="Fluorescent"/>
      <xs:enumeration value="Incandescent"/>
      <xs:enumeration value="Sodium (Natrium)"/>
      <xs:enumeration value="Sunrise (Horizon)"/>
      <xs:enumeration value="Sunset (Rear)"/>
      <xs:enumeration value="ExtremeHot"/>
      <xs:enumeration value="ExtremeCold"/>
      <xs:enumeration value="Underwater"/>
      <xs:enumeration value="CloseUp"/>
      <xs:enumeration value="Motion"/>
    </xs:restriction>
  </xs:simpleType>

```

```

        <xs:enumeration value="FlickerFree50"/>
        <xs:enumeration value="FlickerFree60"/>
    </xs:restriction>
</xs:simpleType>
<xs:complexType name="ImagingPreset">
    <xs:sequence>
        <xs:element name="Name" type="tt:Name"/>
    </xs:sequence>
    <xs:attribute name="token" type="tt:ReferenceToken" use="required"/>
    <xs:attribute name="type" type="xs:string" use="required"/>
    <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:element name="GetPresets">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="VideoSourceToken" type="tt:ReferenceToken"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="GetPresetsResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Preset" type="timg:ImagingPreset"
maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="GetCurrentPreset">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="VideoSourceToken" type="tt:ReferenceToken"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="GetCurrentPresetResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Preset" type="timg:ImagingPreset" minOccurs="0"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="SetCurrentPreset">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="VideoSourceToken" type="tt:ReferenceToken"/>
            <xs:element name="PresetToken" type="tt:ReferenceToken"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="SetCurrentPresetResponse">
    <xs:complexType>
        <xs:sequence/>
    </xs:complexType>
</xs:element>
</xs:schema>
</wsdl:types>
<wsdl:message name="GetServiceCapabilitiesRequest">
    <wsdl:part name="parameters" element="timg:GetServiceCapabilities"/>
</wsdl:message>
<wsdl:message name="GetServiceCapabilitiesResponse">
    <wsdl:part name="parameters" element="timg:GetServiceCapabilitiesResponse"/>
</wsdl:message>
<wsdl:message name="GetImagingSettingsRequest">
    <wsdl:part name="parameters" element="timg:GetImagingSettings"/>
</wsdl:message>
<wsdl:message name="GetImagingSettingsResponse">
    <wsdl:part name="parameters" element="timg:GetImagingSettingsResponse"/>
</wsdl:message>
<wsdl:message name="SetImagingSettingsRequest">
    <wsdl:part name="parameters" element="timg:SetImagingSettings"/>
</wsdl:message>
<wsdl:message name="SetImagingSettingsResponse">
    <wsdl:part name="parameters" element="timg:SetImagingSettingsResponse"/>
</wsdl:message>
<wsdl:message name="GetOptionsRequest">
    <wsdl:part name="parameters" element="timg:GetOptions"/>
</wsdl:message>
<wsdl:message name="GetOptionsResponse">

```



```
<wsdl:part name="parameters" element="timg:GetOptionsResponse"/>
</wsdl:message>
<wsdl:message name="MoveRequest">
  <wsdl:part name="parameters" element="timg:Move"/>
</wsdl:message>
<wsdl:message name="MoveResponse">
  <wsdl:part name="parameters" element="timg:MoveResponse"/>
</wsdl:message>
<wsdl:message name="GetMoveOptionsRequest">
  <wsdl:part name="parameters" element="timg:GetMoveOptions"/>
</wsdl:message>
<wsdl:message name="GetMoveOptionsResponse">
  <wsdl:part name="parameters" element="timg:GetMoveOptionsResponse"/>
</wsdl:message>
<wsdl:message name="StopRequest">
  <wsdl:part name="parameters" element="timg:Stop"/>
</wsdl:message>
<wsdl:message name="StopResponse">
  <wsdl:part name="parameters" element="timg:StopResponse"/>
</wsdl:message>
<wsdl:message name="GetStatusRequest">
  <wsdl:part name="parameters" element="timg:GetStatus"/>
</wsdl:message>
<wsdl:message name="GetStatusResponse">
  <wsdl:part name="parameters" element="timg:GetStatusResponse"/>
</wsdl:message>
<wsdl:message name="GetPresetsRequest">
  <wsdl:part name="parameters" element="timg:GetPresets"/>
</wsdl:message>
<wsdl:message name="GetPresetsResponse">
  <wsdl:part name="parameters" element="timg:GetPresetsResponse"/>
</wsdl:message>
<wsdl:message name="GetCurrentPresetRequest">
  <wsdl:part name="parameters" element="timg:GetCurrentPreset"/>
</wsdl:message>
<wsdl:message name="GetCurrentPresetResponse">
  <wsdl:part name="parameters" element="timg:GetCurrentPresetResponse"/>
</wsdl:message>
<wsdl:message name="SetCurrentPresetRequest">
  <wsdl:part name="parameters" element="timg:SetCurrentPreset"/>
</wsdl:message>
<wsdl:message name="SetCurrentPresetResponse">
  <wsdl:part name="parameters" element="timg:SetCurrentPresetResponse"/>
</wsdl:message>
<wsdl:portType name="ImagingPort">
  <wsdl:operation name="GetServiceCapabilities">
    <wsdl:input message="timg:GetServiceCapabilitiesRequest"/>
    <wsdl:output message="timg:GetServiceCapabilitiesResponse"/>
  </wsdl:operation>
  <wsdl:operation name="GetImagingSettings">
    <wsdl:input message="timg:GetImagingSettingsRequest"/>
    <wsdl:output message="timg:GetImagingSettingsResponse"/>
  </wsdl:operation>
  <wsdl:operation name="SetImagingSettings">
    <wsdl:input message="timg:SetImagingSettingsRequest"/>
    <wsdl:output message="timg:SetImagingSettingsResponse"/>
  </wsdl:operation>
  <wsdl:operation name="GetOptions">
    <wsdl:input message="timg:GetOptionsRequest"/>
    <wsdl:output message="timg:GetOptionsResponse"/>
  </wsdl:operation>
  <wsdl:operation name="Move">
    <wsdl:input message="timg:MoveRequest"/>
    <wsdl:output message="timg:MoveResponse"/>
  </wsdl:operation>
  <wsdl:operation name="GetMoveOptions">
    <wsdl:input message="timg:GetMoveOptionsRequest"/>
    <wsdl:output message="timg:GetMoveOptionsResponse"/>
  </wsdl:operation>
  <wsdl:operation name="Stop">
    <wsdl:input message="timg:StopRequest"/>
    <wsdl:output message="timg:StopResponse"/>
  </wsdl:operation>
  <wsdl:operation name="GetStatus">
    <wsdl:input message="timg:GetStatusRequest"/>
    <wsdl:output message="timg:GetStatusResponse"/>
  </wsdl:operation>
  <wsdl:operation name="GetPresets">
```

```

        <wsdl:input message="timg:GetPresetsRequest"/>
        <wsdl:output message="timg:GetPresetsResponse"/>
    </wsdl:operation>
    <wsdl:operation name="GetCurrentPreset">
        <wsdl:input message="timg:GetCurrentPresetRequest"/>
        <wsdl:output message="timg:GetCurrentPresetResponse"/>
    </wsdl:operation>
    <wsdl:operation name="SetCurrentPreset">
        <wsdl:input message="timg:SetCurrentPresetRequest"/>
        <wsdl:output message="timg:SetCurrentPresetResponse"/>
    </wsdl:operation>
</wsdl:portType>
<wsdl:binding name="ImagingBinding" type="timg:ImagingPort">
    <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
    <wsdl:operation name="GetServiceCapabilities">
        <soap:operation
soapAction="http://www.onvif.org/ver20/imaging/wsdl/GetServiceCapabilities"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetImagingSettings">
        <soap:operation
soapAction="http://www.onvif.org/ver20/imaging/wsdl/GetImagingSettings"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="SetImagingSettings">
        <soap:operation
soapAction="http://www.onvif.org/ver20/imaging/wsdl/SetImagingSettings"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetOptions">
        <soap:operation
soapAction="http://www.onvif.org/ver20/imaging/wsdl/GetOptions"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="Move">
        <soap:operation soapAction="http://www.onvif.org/ver20/imaging/wsdl/Move"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="Stop">
        <soap:operation
soapAction="http://www.onvif.org/ver20/imaging/wsdl/FocusStop"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetStatus">
        <soap:operation
soapAction="http://www.onvif.org/ver20/imaging/wsdl/GetStatus"/>
        <wsdl:input>
            <soap:body use="literal"/>

```

```
</wsdl:input>
<wsdl:output>
  <soap:body use="literal"/>
</wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetMoveOptions">
  <soap:operation
soapAction="http://www.onvif.org/ver20/imaging/wsdl/GetMoveOptions"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetPresets">
  <soap:operation
soapAction="http://www.onvif.org/ver20/imaging/wsdl/GetPresets"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetCurrentPreset">
  <soap:operation
soapAction="http://www.onvif.org/ver20/imaging/wsdl/GetCurrentPreset"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="SetCurrentPreset">
  <soap:operation
soapAction="http://www.onvif.org/ver20/imaging/wsdl/SetCurrentPreset"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
</wsdl:binding>
</wsdl:definitions>
```

IECNORM.COM : Click to view the full PDF of IEC 62676-2-31:2019

F.3 Media

```
<?xml version="1.0" encoding="utf-8"?><?xml-stylesheet type="text/xsl"
href="../../../ver20/util/onvif-wsdl-viewer.xsl"?><wsdl:definitions
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap12/"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:trt="http://www.onvif.org/ver10/media/wsdl"
targetNamespace="http://www.onvif.org/ver10/media/wsdl">
  <wsdl:types>
    <xs:schema xmlns:tt="http://www.onvif.org/ver10/schema"
targetNamespace="http://www.onvif.org/ver10/media/wsdl" elementFormDefault="qualified"
version="2.6">
      <xs:import namespace="http://www.onvif.org/ver10/schema"
schemaLocation="../../../ver10/schema/onvif.xsd"/>
      <xs:element name="GetServiceCapabilities">
        <xs:complexType>
          <xs:sequence/>
        </xs:complexType>
      </xs:element>
      <xs:element name="GetServiceCapabilitiesResponse">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Capabilities" type="trt:Capabilities"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="Capabilities">
        <xs:sequence>
          <xs:element name="ProfileCapabilities" type="trt:ProfileCapabilities"/>
          <xs:element name="StreamingCapabilities"
type="trt:StreamingCapabilities"/>
          <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="SnapshotUri" type="xs:boolean"/>
        <xs:attribute name="Rotation" type="xs:boolean"/>
        <xs:attribute name="VideoSourceMode" type="xs:boolean"/>
        <xs:attribute name="OSD" type="xs:boolean"/>
        <xs:attribute name="EXICompression" type="xs:boolean"/>
        <xs:anyAttribute processContents="lax"/>
      </xs:complexType>
      <xs:element name="Capabilities" type="trt:Capabilities"/>
      <xs:complexType name="ProfileCapabilities">
        <xs:sequence>
          <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="MaximumNumberOfProfiles" type="xs:int"/>
        <xs:anyAttribute processContents="lax"/>
      </xs:complexType>
      <xs:complexType name="StreamingCapabilities">
        <xs:sequence>
          <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="RTPMulticast" type="xs:boolean"/>
        <xs:attribute name="RTP_TCP" type="xs:boolean"/>
        <xs:attribute name="RTP_RTSP_TCP" type="xs:boolean"/>
        <xs:attribute name="NonAggregateControl" type="xs:boolean"/>
        <xs:attribute name="NoRTSPStreaming" type="xs:boolean"/>
        <xs:anyAttribute processContents="lax"/>
      </xs:complexType>
      <xs:element name="GetVideoSources">
        <xs:complexType>
          <xs:sequence/>
        </xs:complexType>
      </xs:element>
      <xs:element name="GetVideoSourcesResponse">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="VideoSources" type="tt:VideoSource" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="GetAudioSources">
```

```

        <xs:complexType>
          <xs:sequence/>
        </xs:complexType>
      </xs:element>
      <xs:element name="GetAudioSourcesResponse">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="AudioSources" type="tt:AudioSource" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="GetAudioOutputs">
        <xs:complexType>
          <xs:sequence/>
        </xs:complexType>
      </xs:element>
      <xs:element name="GetAudioOutputsResponse">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="AudioOutputs" type="tt:AudioOutput" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="CreateProfile">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Name" type="tt:Name"/>
            <xs:element name="Token" type="tt:ReferenceToken" minOccurs="0"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="CreateProfileResponse">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Profile" type="tt:Profile"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="GetProfile">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="ProfileToken" type="tt:ReferenceToken"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="GetProfileResponse">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Profile" type="tt:Profile"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="GetProfiles">
        <xs:complexType>
          <xs:sequence/>
        </xs:complexType>
      </xs:element>
      <xs:element name="GetProfilesResponse">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Profiles" type="tt:Profile" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="AddVideoEncoderConfiguration">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="ProfileToken" type="tt:ReferenceToken"/>
            <xs:element name="ConfigurationToken" type="tt:ReferenceToken"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="AddVideoEncoderConfigurationResponse">
        <xs:complexType>
          <xs:sequence/>

```

```

    </xs:complexType>
  </xs:element>
  <xs:element name="RemoveVideoEncoderConfiguration">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="ProfileToken" type="tt:ReferenceToken"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="RemoveVideoEncoderConfigurationResponse">
    <xs:complexType>
      <xs:sequence/>
    </xs:complexType>
  </xs:element>
  <xs:element name="AddVideoSourceConfiguration">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="ProfileToken" type="tt:ReferenceToken"/>
        <xs:element name="ConfigurationToken" type="tt:ReferenceToken"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="AddVideoSourceConfigurationResponse">
    <xs:complexType>
      <xs:sequence/>
    </xs:complexType>
  </xs:element>
  <xs:element name="RemoveVideoSourceConfiguration">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="ProfileToken" type="tt:ReferenceToken"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="RemoveVideoSourceConfigurationResponse">
    <xs:complexType>
      <xs:sequence/>
    </xs:complexType>
  </xs:element>
  <xs:element name="AddAudioEncoderConfiguration">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="ProfileToken" type="tt:ReferenceToken"/>
        <xs:element name="ConfigurationToken" type="tt:ReferenceToken"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="AddAudioEncoderConfigurationResponse">
    <xs:complexType>
      <xs:sequence/>
    </xs:complexType>
  </xs:element>
  <xs:element name="RemoveAudioEncoderConfiguration">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="ProfileToken" type="tt:ReferenceToken"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="RemoveAudioEncoderConfigurationResponse">
    <xs:complexType>
      <xs:sequence/>
    </xs:complexType>
  </xs:element>
  <xs:element name="AddAudioSourceConfiguration">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="ProfileToken" type="tt:ReferenceToken"/>
        <xs:element name="ConfigurationToken" type="tt:ReferenceToken"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="AddAudioSourceConfigurationResponse">
    <xs:complexType>
      <xs:sequence/>
    </xs:complexType>
  </xs:element>
  <xs:element name="RemoveAudioSourceConfiguration">

```

```

        <xs:complexType>
          <xs:sequence>
            <xs:element name="ProfileToken" type="tt:ReferenceToken"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="RemoveAudioSourceConfigurationResponse">
        <xs:complexType>
          <xs:sequence/>
        </xs:complexType>
      </xs:element>
      <xs:element name="AddPTZConfiguration">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="ProfileToken" type="tt:ReferenceToken"/>
            <xs:element name="ConfigurationToken" type="tt:ReferenceToken"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="AddPTZConfigurationResponse">
        <xs:complexType>
          <xs:sequence/>
        </xs:complexType>
      </xs:element>
      <xs:element name="RemovePTZConfiguration">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="ProfileToken" type="tt:ReferenceToken"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="RemovePTZConfigurationResponse">
        <xs:complexType>
          <xs:sequence/>
        </xs:complexType>
      </xs:element>
      <xs:element name="AddVideoAnalyticsConfiguration">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="ProfileToken" type="tt:ReferenceToken"/>
            <xs:element name="ConfigurationToken" type="tt:ReferenceToken"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="AddVideoAnalyticsConfigurationResponse">
        <xs:complexType>
          <xs:sequence/>
        </xs:complexType>
      </xs:element>
      <xs:element name="RemoveVideoAnalyticsConfiguration">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="ProfileToken" type="tt:ReferenceToken"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="RemoveVideoAnalyticsConfigurationResponse">
        <xs:complexType>
          <xs:sequence/>
        </xs:complexType>
      </xs:element>
      <xs:element name="AddMetadataConfiguration">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="ProfileToken" type="tt:ReferenceToken"/>
            <xs:element name="ConfigurationToken" type="tt:ReferenceToken"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="AddMetadataConfigurationResponse">
        <xs:complexType>
          <xs:sequence/>
        </xs:complexType>
      </xs:element>
      <xs:element name="RemoveMetadataConfiguration">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="ProfileToken" type="tt:ReferenceToken"/>

```

```

        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="RemoveMetadataConfigurationResponse">
      <xs:complexType>
        <xs:sequence/>
      </xs:complexType>
    </xs:element>
    <xs:element name="AddAudioOutputConfiguration">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="ProfileToken" type="tt:ReferenceToken"/>
          <xs:element name="ConfigurationToken" type="tt:ReferenceToken"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="AddAudioOutputConfigurationResponse">
      <xs:complexType>
        <xs:sequence/>
      </xs:complexType>
    </xs:element>
    <xs:element name="RemoveAudioOutputConfiguration">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="ProfileToken" type="tt:ReferenceToken"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="RemoveAudioOutputConfigurationResponse">
      <xs:complexType>
        <xs:sequence/>
      </xs:complexType>
    </xs:element>
    <xs:element name="AddAudioDecoderConfiguration">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="ProfileToken" type="tt:ReferenceToken"/>
          <xs:element name="ConfigurationToken" type="tt:ReferenceToken"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="AddAudioDecoderConfigurationResponse">
      <xs:complexType>
        <xs:sequence/>
      </xs:complexType>
    </xs:element>
    <xs:element name="RemoveAudioDecoderConfiguration">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="ProfileToken" type="tt:ReferenceToken"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="RemoveAudioDecoderConfigurationResponse">
      <xs:complexType>
        <xs:sequence/>
      </xs:complexType>
    </xs:element>
    <xs:element name="DeleteProfile">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="ProfileToken" type="tt:ReferenceToken"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="DeleteProfileResponse">
      <xs:complexType>
        <xs:sequence/>
      </xs:complexType>
    </xs:element>
    <xs:element name="GetVideoEncoderConfigurations">
      <xs:complexType>
        <xs:sequence/>
      </xs:complexType>
    </xs:element>
    <xs:element name="GetVideoEncoderConfigurationsResponse">
      <xs:complexType>
        <xs:sequence>

```



```

        <xs:element name="Configurations" type="tt:VideoEncoderConfiguration"
minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="GetVideoSourceConfigurations">
    <xs:complexType>
        <xs:sequence/>
    </xs:complexType>
</xs:element>
<xs:element name="GetVideoSourceConfigurationsResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Configurations" type="tt:VideoSourceConfiguration"
minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
<xs:element name="GetAudioEncoderConfigurations">
    <xs:complexType>
        <xs:sequence/>
    </xs:complexType>
</xs:element>
<xs:element name="GetAudioEncoderConfigurationsResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Configurations" type="tt:AudioEncoderConfiguration"
minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
<xs:element name="GetAudioSourceConfigurations">
    <xs:complexType>
        <xs:sequence/>
    </xs:complexType>
</xs:element>
<xs:element name="GetAudioSourceConfigurationsResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Configurations" type="tt:AudioSourceConfiguration"
minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
<xs:element name="GetVideoAnalyticsConfigurations">
    <xs:complexType>
        <xs:sequence/>
    </xs:complexType>
</xs:element>
<xs:element name="GetVideoAnalyticsConfigurationsResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Configurations"
type="tt:VideoAnalyticsConfiguration" minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
<xs:element name="GetMetadataConfigurations">
    <xs:complexType>
        <xs:sequence/>
    </xs:complexType>
</xs:element>
<xs:element name="GetMetadataConfigurationsResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Configurations" type="tt:MetadataConfiguration"
minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
<xs:element name="GetAudioOutputConfigurations">
    <xs:complexType>
        <xs:sequence/>
    </xs:complexType>
</xs:element>
<xs:element name="GetAudioOutputConfigurationsResponse">
    <xs:complexType>
        <xs:sequence>

```

```

        <xs:element name="Configurations" type="tt:AudioOutputConfiguration"
minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="GetAudioDecoderConfigurations">
    <xs:complexType>
        <xs:sequence/>
    </xs:complexType>
</xs:element>
<xs:element name="GetAudioDecoderConfigurationsResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Configurations" type="tt:AudioDecoderConfiguration"
minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="GetVideoSourceConfiguration">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="ConfigurationToken" type="tt:ReferenceToken"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="GetVideoSourceConfigurationResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Configuration" type="tt:VideoSourceConfiguration"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="GetVideoEncoderConfiguration">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="ConfigurationToken" type="tt:ReferenceToken"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="GetVideoEncoderConfigurationResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Configuration"
type="tt:VideoEncoderConfiguration"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="GetAudioSourceConfiguration">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="ConfigurationToken" type="tt:ReferenceToken"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="GetAudioSourceConfigurationResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Configuration" type="tt:AudioSourceConfiguration"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="GetAudioEncoderConfiguration">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="ConfigurationToken" type="tt:ReferenceToken"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="GetAudioEncoderConfigurationResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Configuration"
type="tt:AudioEncoderConfiguration"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="GetVideoAnalyticsConfiguration">
    <xs:complexType>

```

```

        <xs:sequence>
          <xs:element name="ConfigurationToken" type="tt:ReferenceToken"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="GetVideoAnalyticsConfigurationResponse">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="Configuration"
type="tt:VideoAnalyticsConfiguration"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="GetMetadataConfiguration">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="ConfigurationToken" type="tt:ReferenceToken"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="GetMetadataConfigurationResponse">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="Configuration" type="tt:MetadataConfiguration"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="GetAudioOutputConfiguration">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="ConfigurationToken" type="tt:ReferenceToken"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="GetAudioOutputConfigurationResponse">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="Configuration" type="tt:AudioOutputConfiguration"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="GetAudioDecoderConfiguration">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="ConfigurationToken" type="tt:ReferenceToken"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="GetAudioDecoderConfigurationResponse">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="Configuration"
type="tt:AudioDecoderConfiguration"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="GetCompatibleVideoEncoderConfigurations">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="ProfileToken" type="tt:ReferenceToken"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="GetCompatibleVideoEncoderConfigurationsResponse">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="Configurations" type="tt:VideoEncoderConfiguration"
minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="GetCompatibleVideoSourceConfigurations">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="ProfileToken" type="tt:ReferenceToken"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>

```

```

<xs:element name="GetCompatibleVideoSourceConfigurationsResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Configurations" type="tt:VideoSourceConfiguration"
minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="GetCompatibleAudioEncoderConfigurations">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="ProfileToken" type="tt:ReferenceToken"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="GetCompatibleAudioEncoderConfigurationsResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Configurations" type="tt:AudioEncoderConfiguration"
minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="GetCompatibleAudioSourceConfigurations">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="ProfileToken" type="tt:ReferenceToken"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="GetCompatibleAudioSourceConfigurationsResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Configurations" type="tt:AudioSourceConfiguration"
minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="GetCompatibleVideoAnalyticsConfigurations">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="ProfileToken" type="tt:ReferenceToken"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="GetCompatibleVideoAnalyticsConfigurationsResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Configurations"
type="tt:VideoAnalyticsConfiguration" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="GetCompatibleMetadataConfigurations">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="ProfileToken" type="tt:ReferenceToken"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="GetCompatibleMetadataConfigurationsResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Configurations" type="tt:MetadataConfiguration"
minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="GetCompatibleAudioOutputConfigurations">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="ProfileToken" type="tt:ReferenceToken"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="GetCompatibleAudioOutputConfigurationsResponse">
  <xs:complexType>
    <xs:sequence>

```

```

        <xs:element name="Configurations" type="tt:AudioOutputConfiguration"
minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="GetCompatibleAudioDecoderConfigurations">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="ProfileToken" type="tt:ReferenceToken"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="GetCompatibleAudioDecoderConfigurationsResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Configurations" type="tt:AudioDecoderConfiguration"
minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="SetVideoEncoderConfiguration">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Configuration"
type="tt:VideoEncoderConfiguration"/>
            <xs:element name="ForcePersistence" type="xs:boolean"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="SetVideoEncoderConfigurationResponse">
    <xs:complexType>
        <xs:sequence/>
    </xs:complexType>
</xs:element>
<xs:element name="SetVideoSourceConfiguration">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Configuration" type="tt:VideoSourceConfiguration"/>
            <xs:element name="ForcePersistence" type="xs:boolean"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="SetVideoSourceConfigurationResponse">
    <xs:complexType>
        <xs:sequence/>
    </xs:complexType>
</xs:element>
<xs:element name="SetAudioEncoderConfiguration">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Configuration"
type="tt:AudioEncoderConfiguration"/>
            <xs:element name="ForcePersistence" type="xs:boolean"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="SetAudioEncoderConfigurationResponse">
    <xs:complexType>
        <xs:sequence/>
    </xs:complexType>
</xs:element>
<xs:element name="SetAudioSourceConfiguration">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Configuration" type="tt:AudioSourceConfiguration"/>
            <xs:element name="ForcePersistence" type="xs:boolean"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="SetAudioSourceConfigurationResponse">
    <xs:complexType>
        <xs:sequence/>
    </xs:complexType>
</xs:element>
<xs:element name="SetVideoAnalyticsConfiguration">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Configuration"

```

```

type="tt:VideoAnalyticsConfiguration"/>
    <xs:element name="ForcePersistence" type="xs:boolean"/>
  </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="SetVideoAnalyticsConfigurationResponse">
  <xs:complexType>
    <xs:sequence/>
  </xs:complexType>
</xs:element>
<xs:element name="SetMetadataConfiguration">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Configuration" type="tt:MetadataConfiguration"/>
      <xs:element name="ForcePersistence" type="xs:boolean"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="SetMetadataConfigurationResponse">
  <xs:complexType>
    <xs:sequence/>
  </xs:complexType>
</xs:element>
<xs:element name="SetAudioOutputConfiguration">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Configuration" type="tt:AudioOutputConfiguration"/>
      <xs:element name="ForcePersistence" type="xs:boolean"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="SetAudioOutputConfigurationResponse">
  <xs:complexType>
    <xs:sequence/>
  </xs:complexType>
</xs:element>
<xs:element name="SetAudioDecoderConfiguration">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Configuration"
type="tt:AudioDecoderConfiguration"/>
      <xs:element name="ForcePersistence" type="xs:boolean"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="SetAudioDecoderConfigurationResponse">
  <xs:complexType>
    <xs:sequence/>
  </xs:complexType>
</xs:element>
<xs:element name="GetVideoSourceConfigurationOptions">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="ConfigurationToken" type="tt:ReferenceToken"
minOccurs="0"/>
      <xs:element name="ProfileToken" type="tt:ReferenceToken"
minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="GetVideoSourceConfigurationOptionsResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Options"
type="tt:VideoSourceConfigurationOptions"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="GetVideoEncoderConfigurationOptions">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="ConfigurationToken" type="tt:ReferenceToken"
minOccurs="0"/>
      <xs:element name="ProfileToken" type="tt:ReferenceToken"
minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

```

<xs:element name="GetVideoEncoderConfigurationOptionsResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Options"
type="tt:VideoEncoderConfigurationOptions"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="GetAudioSourceConfigurationOptions">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="ConfigurationToken" type="tt:ReferenceToken"
minOccurs="0"/>
      <xs:element name="ProfileToken" type="tt:ReferenceToken"
minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="GetAudioSourceConfigurationOptionsResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Options"
type="tt:AudioSourceConfigurationOptions"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="GetAudioEncoderConfigurationOptions">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="ConfigurationToken" type="tt:ReferenceToken"
minOccurs="0"/>
      <xs:element name="ProfileToken" type="tt:ReferenceToken"
minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="GetAudioEncoderConfigurationOptionsResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Options"
type="tt:AudioEncoderConfigurationOptions"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="GetMetadataConfigurationOptions">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="ConfigurationToken" type="tt:ReferenceToken"
minOccurs="0"/>
      <xs:element name="ProfileToken" type="tt:ReferenceToken"
minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="GetMetadataConfigurationOptionsResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Options" type="tt:MetadataConfigurationOptions"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="GetAudioOutputConfigurationOptions">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="ConfigurationToken" type="tt:ReferenceToken"
minOccurs="0"/>
      <xs:element name="ProfileToken" type="tt:ReferenceToken"
minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="GetAudioOutputConfigurationOptionsResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Options"
type="tt:AudioOutputConfigurationOptions"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

```

</xs:element>
<xs:element name="GetAudioDecoderConfigurationOptions">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="ConfigurationToken" type="tt:ReferenceToken"
minOccurs="0"/>
      <xs:element name="ProfileToken" type="tt:ReferenceToken"
minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="GetAudioDecoderConfigurationOptionsResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Options"
type="tt:AudioDecoderConfigurationOptions"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="GetGuaranteedNumberOfVideoEncoderInstances">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="ConfigurationToken" type="tt:ReferenceToken"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="GetGuaranteedNumberOfVideoEncoderInstancesResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="TotalNumber" type="xs:int"/>
      <xs:element name="JPEG" type="xs:int" minOccurs="0"/>
      <xs:element name="H264" type="xs:int" minOccurs="0"/>
      <xs:element name="MPEG4" type="xs:int" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="GetStreamUri">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="StreamSetup" type="tt:StreamSetup"/>
      <xs:element name="ProfileToken" type="tt:ReferenceToken"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="GetStreamUriResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="MediaUri" type="tt:MediaUri"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="StartMulticastStreaming">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="ProfileToken" type="tt:ReferenceToken"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="StartMulticastStreamingResponse">
  <xs:complexType>
    <xs:sequence/>
  </xs:complexType>
</xs:element>
<xs:element name="StopMulticastStreaming">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="ProfileToken" type="tt:ReferenceToken"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="StopMulticastStreamingResponse">
  <xs:complexType>
    <xs:sequence/>
  </xs:complexType>
</xs:element>
<xs:element name="SetSynchronizationPoint">
  <xs:complexType>
    <xs:sequence>

```



```

        <xs:element name="ProfileToken" type="tt:ReferenceToken"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="SetSynchronizationPointResponse">
    <xs:complexType>
      <xs:sequence/>
    </xs:complexType>
  </xs:element>
  <xs:element name="GetSnapshotUri">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="ProfileToken" type="tt:ReferenceToken"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="GetSnapshotUriResponse">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="MediaUri" type="tt:MediaUri"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="GetVideoSourceModes">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="VideoSourceToken" type="tt:ReferenceToken"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="GetVideoSourceModesResponse">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="VideoSourceModes" type="trt:VideoSourceMode"
maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="SetVideoSourceMode">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="VideoSourceToken" type="tt:ReferenceToken"/>
        <xs:element name="VideoSourceModeToken" type="tt:ReferenceToken"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="SetVideoSourceModeResponse">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Reboot" type="xs:boolean"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:simpleType name="EncodingTypes">
    <xs:list itemType="xs:string"/>
  </xs:simpleType>
  <xs:complexType name="VideoSourceMode">
    <xs:sequence>
      <xs:element name="MaxFramerate" type="xs:float"/>
      <xs:element name="MaxResolution" type="tt:VideoResolution"/>
      <xs:element name="Encodings" type="trt:EncodingTypes"/>
      <xs:element name="Reboot" type="xs:boolean"/>
      <xs:element name="Description" type="tt:Description" minOccurs="0"/>
      <xs:element name="Extension" type="trt:VideoSourceModeExtension"
minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="token" type="tt:ReferenceToken" use="required"/>
    <xs:attribute name="Enabled" type="xs:boolean"/>
    <xs:anyAttribute processContents="lax"/>
  </xs:complexType>
  <xs:complexType name="VideoSourceModeExtension">
    <xs:sequence>
      <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="GetOSDs">
    <xs:complexType>

```

```

        <xs:sequence>
            <xs:element name="ConfigurationToken" type="tt:ReferenceToken"
minOccurs="0" maxOccurs="1"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="GetOSDsResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="OSDs" type="tt:OSDConfiguration" minOccurs="0"
maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="GetOSD">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="OSDToken" type="tt:ReferenceToken"/>
            <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="GetOSDResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="OSD" type="tt:OSDConfiguration"/>
            <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="SetOSD">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="OSD" type="tt:OSDConfiguration"/>
            <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="SetOSDResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="GetOSDOptions">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="ConfigurationToken" type="tt:ReferenceToken"/>
            <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="GetOSDOptionsResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="OSDOptions" type="tt:OSDConfigurationOptions"/>
            <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="CreateOSD">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="OSD" type="tt:OSDConfiguration"/>
            <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="CreateOSDResponse">
    <xs:complexType>

```

```

        <xs:sequence>
            <xs:element name="OSDTOKEN" type="tt:ReferenceToken"/>
            <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="DeleteOSD">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="OSDTOKEN" type="tt:ReferenceToken"/>
            <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="DeleteOSDResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
</xs:schema>
</wsdl:types>
<wsdl:message name="GetServiceCapabilitiesRequest">
    <wsdl:part name="parameters" element="trt:GetServiceCapabilities"/>
</wsdl:message>
<wsdl:message name="GetServiceCapabilitiesResponse">
    <wsdl:part name="parameters" element="trt:GetServiceCapabilitiesResponse"/>
</wsdl:message>
<wsdl:message name="GetVideoSourcesRequest">
    <wsdl:part name="parameters" element="trt:GetVideoSources"/>
</wsdl:message>
<wsdl:message name="GetVideoSourcesResponse">
    <wsdl:part name="parameters" element="trt:GetVideoSourcesResponse"/>
</wsdl:message>
<wsdl:message name="GetAudioSourcesRequest">
    <wsdl:part name="parameters" element="trt:GetAudioSources"/>
</wsdl:message>
<wsdl:message name="GetAudioSourcesResponse">
    <wsdl:part name="parameters" element="trt:GetAudioSourcesResponse"/>
</wsdl:message>
<wsdl:message name="GetAudioOutputsRequest">
    <wsdl:part name="parameters" element="trt:GetAudioOutputs"/>
</wsdl:message>
<wsdl:message name="GetAudioOutputsResponse">
    <wsdl:part name="parameters" element="trt:GetAudioOutputsResponse"/>
</wsdl:message>
<wsdl:message name="CreateProfileRequest">
    <wsdl:part name="parameters" element="trt:CreateProfile"/>
</wsdl:message>
<wsdl:message name="CreateProfileResponse">
    <wsdl:part name="parameters" element="trt:CreateProfileResponse"/>
</wsdl:message>
<wsdl:message name="GetProfileRequest">
    <wsdl:part name="parameters" element="trt:GetProfile"/>
</wsdl:message>
<wsdl:message name="GetProfileResponse">
    <wsdl:part name="parameters" element="trt:GetProfileResponse"/>
</wsdl:message>
<wsdl:message name="GetProfilesRequest">
    <wsdl:part name="parameters" element="trt:GetProfiles"/>
</wsdl:message>
<wsdl:message name="GetProfilesResponse">
    <wsdl:part name="parameters" element="trt:GetProfilesResponse"/>
</wsdl:message>
<wsdl:message name="AddVideoEncoderConfigurationRequest">
    <wsdl:part name="parameters" element="trt:AddVideoEncoderConfiguration"/>
</wsdl:message>
<wsdl:message name="AddVideoEncoderConfigurationResponse">
    <wsdl:part name="parameters" element="trt:AddVideoEncoderConfigurationResponse"/>
</wsdl:message>
<wsdl:message name="RemoveVideoEncoderConfigurationRequest">
    <wsdl:part name="parameters" element="trt:RemoveVideoEncoderConfiguration"/>
</wsdl:message>
<wsdl:message name="RemoveVideoEncoderConfigurationResponse">

```

```

        <wsdl:part name="parameters"
element="trt:RemoveVideoEncoderConfigurationResponse"/>
    </wsdl:message>
    <wsdl:message name="AddVideoSourceConfigurationRequest">
        <wsdl:part name="parameters" element="trt:AddVideoSourceConfiguration"/>
    </wsdl:message>
    <wsdl:message name="AddVideoSourceConfigurationResponse">
        <wsdl:part name="parameters" element="trt:AddVideoSourceConfigurationResponse"/>
    </wsdl:message>
    <wsdl:message name="RemoveVideoSourceConfigurationRequest">
        <wsdl:part name="parameters" element="trt:RemoveVideoSourceConfiguration"/>
    </wsdl:message>
    <wsdl:message name="RemoveVideoSourceConfigurationResponse">
        <wsdl:part name="parameters"
element="trt:RemoveVideoSourceConfigurationResponse"/>
    </wsdl:message>
    <wsdl:message name="AddAudioEncoderConfigurationRequest">
        <wsdl:part name="parameters" element="trt:AddAudioEncoderConfiguration"/>
    </wsdl:message>
    <wsdl:message name="AddAudioEncoderConfigurationResponse">
        <wsdl:part name="parameters" element="trt:AddAudioEncoderConfigurationResponse"/>
    </wsdl:message>
    <wsdl:message name="RemoveAudioEncoderConfigurationRequest">
        <wsdl:part name="parameters" element="trt:RemoveAudioEncoderConfiguration"/>
    </wsdl:message>
    <wsdl:message name="RemoveAudioEncoderConfigurationResponse">
        <wsdl:part name="parameters"
element="trt:RemoveAudioEncoderConfigurationResponse"/>
    </wsdl:message>
    <wsdl:message name="AddAudioSourceConfigurationRequest">
        <wsdl:part name="parameters" element="trt:AddAudioSourceConfiguration"/>
    </wsdl:message>
    <wsdl:message name="AddAudioSourceConfigurationResponse">
        <wsdl:part name="parameters" element="trt:AddAudioSourceConfigurationResponse"/>
    </wsdl:message>
    <wsdl:message name="RemoveAudioSourceConfigurationRequest">
        <wsdl:part name="parameters" element="trt:RemoveAudioSourceConfiguration"/>
    </wsdl:message>
    <wsdl:message name="RemoveAudioSourceConfigurationResponse">
        <wsdl:part name="parameters"
element="trt:RemoveAudioSourceConfigurationResponse"/>
    </wsdl:message>
    <wsdl:message name="AddPTZConfigurationRequest">
        <wsdl:part name="parameters" element="trt:AddPTZConfiguration"/>
    </wsdl:message>
    <wsdl:message name="AddPTZConfigurationResponse">
        <wsdl:part name="parameters" element="trt:AddPTZConfigurationResponse"/>
    </wsdl:message>
    <wsdl:message name="RemovePTZConfigurationRequest">
        <wsdl:part name="parameters" element="trt:RemovePTZConfiguration"/>
    </wsdl:message>
    <wsdl:message name="RemovePTZConfigurationResponse">
        <wsdl:part name="parameters" element="trt:RemovePTZConfigurationResponse"/>
    </wsdl:message>
    <wsdl:message name="AddVideoAnalyticsConfigurationRequest">
        <wsdl:part name="parameters" element="trt:AddVideoAnalyticsConfiguration"/>
    </wsdl:message>
    <wsdl:message name="AddVideoAnalyticsConfigurationResponse">
        <wsdl:part name="parameters"
element="trt:AddVideoAnalyticsConfigurationResponse"/>
    </wsdl:message>
    <wsdl:message name="RemoveVideoAnalyticsConfigurationRequest">
        <wsdl:part name="parameters" element="trt:RemoveVideoAnalyticsConfiguration"/>
    </wsdl:message>
    <wsdl:message name="RemoveVideoAnalyticsConfigurationResponse">
        <wsdl:part name="parameters"
element="trt:RemoveVideoAnalyticsConfigurationResponse"/>
    </wsdl:message>
    <wsdl:message name="AddMetadataConfigurationRequest">
        <wsdl:part name="parameters" element="trt:AddMetadataConfiguration"/>
    </wsdl:message>
    <wsdl:message name="AddMetadataConfigurationResponse">
        <wsdl:part name="parameters" element="trt:AddMetadataConfigurationResponse"/>
    </wsdl:message>
    <wsdl:message name="RemoveMetadataConfigurationRequest">
        <wsdl:part name="parameters" element="trt:RemoveMetadataConfiguration"/>
    </wsdl:message>
    <wsdl:message name="RemoveMetadataConfigurationResponse">

```

```

    <wsdl:part name="parameters" element="trt:RemoveMetadataConfigurationResponse"/>
  </wsdl:message>
  <wsdl:message name="AddAudioOutputConfigurationRequest">
    <wsdl:part name="parameters" element="trt:AddAudioOutputConfiguration"/>
  </wsdl:message>
  <wsdl:message name="AddAudioOutputConfigurationResponse">
    <wsdl:part name="parameters" element="trt:AddAudioOutputConfigurationResponse"/>
  </wsdl:message>
  <wsdl:message name="RemoveAudioOutputConfigurationRequest">
    <wsdl:part name="parameters" element="trt:RemoveAudioOutputConfiguration"/>
  </wsdl:message>
  <wsdl:message name="RemoveAudioOutputConfigurationResponse">
    <wsdl:part name="parameters"
element="trt:RemoveAudioOutputConfigurationResponse"/>
  </wsdl:message>
  <wsdl:message name="AddAudioDecoderConfigurationRequest">
    <wsdl:part name="parameters" element="trt:AddAudioDecoderConfiguration"/>
  </wsdl:message>
  <wsdl:message name="AddAudioDecoderConfigurationResponse">
    <wsdl:part name="parameters" element="trt:AddAudioDecoderConfigurationResponse"/>
  </wsdl:message>
  <wsdl:message name="RemoveAudioDecoderConfigurationRequest">
    <wsdl:part name="parameters" element="trt:RemoveAudioDecoderConfiguration"/>
  </wsdl:message>
  <wsdl:message name="RemoveAudioDecoderConfigurationResponse">
    <wsdl:part name="parameters"
element="trt:RemoveAudioDecoderConfigurationResponse"/>
  </wsdl:message>
  <wsdl:message name="DeleteProfileRequest">
    <wsdl:part name="parameters" element="trt:DeleteProfile"/>
  </wsdl:message>
  <wsdl:message name="DeleteProfileResponse">
    <wsdl:part name="parameters" element="trt:DeleteProfileResponse"/>
  </wsdl:message>
  <wsdl:message name="GetVideoSourceConfigurationsRequest">
    <wsdl:part name="parameters" element="trt:GetVideoSourceConfigurations"/>
  </wsdl:message>
  <wsdl:message name="GetVideoSourceConfigurationsResponse">
    <wsdl:part name="parameters" element="trt:GetVideoSourceConfigurationsResponse"/>
  </wsdl:message>
  <wsdl:message name="GetVideoEncoderConfigurationsRequest">
    <wsdl:part name="parameters" element="trt:GetVideoEncoderConfigurations"/>
  </wsdl:message>
  <wsdl:message name="GetVideoEncoderConfigurationsResponse">
    <wsdl:part name="parameters"
element="trt:GetVideoEncoderConfigurationsResponse"/>
  </wsdl:message>
  <wsdl:message name="GetAudioSourceConfigurationsRequest">
    <wsdl:part name="parameters" element="trt:GetAudioSourceConfigurations"/>
  </wsdl:message>
  <wsdl:message name="GetAudioSourceConfigurationsResponse">
    <wsdl:part name="parameters" element="trt:GetAudioSourceConfigurationsResponse"/>
  </wsdl:message>
  <wsdl:message name="GetAudioEncoderConfigurationsRequest">
    <wsdl:part name="parameters" element="trt:GetAudioEncoderConfigurations"/>
  </wsdl:message>
  <wsdl:message name="GetAudioEncoderConfigurationsResponse">
    <wsdl:part name="parameters"
element="trt:GetAudioEncoderConfigurationsResponse"/>
  </wsdl:message>
  <wsdl:message name="GetVideoAnalyticsConfigurationsRequest">
    <wsdl:part name="parameters" element="trt:GetVideoAnalyticsConfigurations"/>
  </wsdl:message>
  <wsdl:message name="GetVideoAnalyticsConfigurationsResponse">
    <wsdl:part name="parameters"
element="trt:GetVideoAnalyticsConfigurationsResponse"/>
  </wsdl:message>
  <wsdl:message name="GetMetadataConfigurationsRequest">
    <wsdl:part name="parameters" element="trt:GetMetadataConfigurations"/>
  </wsdl:message>
  <wsdl:message name="GetMetadataConfigurationsResponse">
    <wsdl:part name="parameters" element="trt:GetMetadataConfigurationsResponse"/>
  </wsdl:message>
  <wsdl:message name="GetAudioOutputConfigurationsRequest">
    <wsdl:part name="parameters" element="trt:GetAudioOutputConfigurations"/>
  </wsdl:message>
  <wsdl:message name="GetAudioOutputConfigurationsResponse">
    <wsdl:part name="parameters" element="trt:GetAudioOutputConfigurationsResponse"/>
  </wsdl:message>

```

```

</wsdl:message>
<wsdl:message name="GetAudioDecoderConfigurationsRequest">
  <wsdl:part name="parameters" element="trt:GetAudioDecoderConfigurations"/>
</wsdl:message>
<wsdl:message name="GetAudioDecoderConfigurationsResponse">
  <wsdl:part name="parameters"
element="trt:GetAudioDecoderConfigurationsResponse"/>
</wsdl:message>
<wsdl:message name="GetVideoSourceConfigurationRequest">
  <wsdl:part name="parameters" element="trt:GetVideoSourceConfiguration"/>
</wsdl:message>
<wsdl:message name="GetVideoSourceConfigurationResponse">
  <wsdl:part name="parameters" element="trt:GetVideoSourceConfigurationResponse"/>
</wsdl:message>
<wsdl:message name="GetVideoEncoderConfigurationRequest">
  <wsdl:part name="parameters" element="trt:GetVideoEncoderConfiguration"/>
</wsdl:message>
<wsdl:message name="GetVideoEncoderConfigurationResponse">
  <wsdl:part name="parameters" element="trt:GetVideoEncoderConfigurationResponse"/>
</wsdl:message>
<wsdl:message name="GetAudioSourceConfigurationRequest">
  <wsdl:part name="parameters" element="trt:GetAudioSourceConfiguration"/>
</wsdl:message>
<wsdl:message name="GetAudioSourceConfigurationResponse">
  <wsdl:part name="parameters" element="trt:GetAudioSourceConfigurationResponse"/>
</wsdl:message>
<wsdl:message name="GetAudioEncoderConfigurationRequest">
  <wsdl:part name="parameters" element="trt:GetAudioEncoderConfiguration"/>
</wsdl:message>
<wsdl:message name="GetAudioEncoderConfigurationResponse">
  <wsdl:part name="parameters" element="trt:GetAudioEncoderConfigurationResponse"/>
</wsdl:message>
<wsdl:message name="GetVideoAnalyticsConfigurationRequest">
  <wsdl:part name="parameters" element="trt:GetVideoAnalyticsConfiguration"/>
</wsdl:message>
<wsdl:message name="GetVideoAnalyticsConfigurationResponse">
  <wsdl:part name="parameters"
element="trt:GetVideoAnalyticsConfigurationResponse"/>
</wsdl:message>
<wsdl:message name="GetMetadataConfigurationRequest">
  <wsdl:part name="parameters" element="trt:GetMetadataConfiguration"/>
</wsdl:message>
<wsdl:message name="GetMetadataConfigurationResponse">
  <wsdl:part name="parameters" element="trt:GetMetadataConfigurationResponse"/>
</wsdl:message>
<wsdl:message name="GetAudioOutputConfigurationRequest">
  <wsdl:part name="parameters" element="trt:GetAudioOutputConfiguration"/>
</wsdl:message>
<wsdl:message name="GetAudioOutputConfigurationResponse">
  <wsdl:part name="parameters" element="trt:GetAudioOutputConfigurationResponse"/>
</wsdl:message>
<wsdl:message name="GetAudioDecoderConfigurationRequest">
  <wsdl:part name="parameters" element="trt:GetAudioDecoderConfiguration"/>
</wsdl:message>
<wsdl:message name="GetAudioDecoderConfigurationResponse">
  <wsdl:part name="parameters" element="trt:GetAudioDecoderConfigurationResponse"/>
</wsdl:message>
<wsdl:message name="GetCompatibleVideoEncoderConfigurationsRequest">
  <wsdl:part name="parameters"
element="trt:GetCompatibleVideoEncoderConfigurations"/>
</wsdl:message>
<wsdl:message name="GetCompatibleVideoEncoderConfigurationsResponse">
  <wsdl:part name="parameters"
element="trt:GetCompatibleVideoEncoderConfigurationsResponse"/>
</wsdl:message>
<wsdl:message name="GetCompatibleVideoSourceConfigurationsRequest">
  <wsdl:part name="parameters"
element="trt:GetCompatibleVideoSourceConfigurations"/>
</wsdl:message>
<wsdl:message name="GetCompatibleVideoSourceConfigurationsResponse">
  <wsdl:part name="parameters"
element="trt:GetCompatibleVideoSourceConfigurationsResponse"/>
</wsdl:message>
<wsdl:message name="GetCompatibleAudioEncoderConfigurationsRequest">
  <wsdl:part name="parameters"
element="trt:GetCompatibleAudioEncoderConfigurations"/>
</wsdl:message>
<wsdl:message name="GetCompatibleAudioEncoderConfigurationsResponse">

```



```

        <wsdl:part name="parameters"
element="trt:GetCompatibleAudioEncoderConfigurationsResponse"/>
    </wsdl:message>
    <wsdl:message name="GetCompatibleAudioSourceConfigurationsRequest">
        <wsdl:part name="parameters"
element="trt:GetCompatibleAudioSourceConfigurations"/>
    </wsdl:message>
    <wsdl:message name="GetCompatibleAudioSourceConfigurationsResponse">
        <wsdl:part name="parameters"
element="trt:GetCompatibleAudioSourceConfigurationsResponse"/>
    </wsdl:message>
    <wsdl:message name="GetCompatibleVideoAnalyticsConfigurationsRequest">
        <wsdl:part name="parameters"
element="trt:GetCompatibleVideoAnalyticsConfigurations"/>
    </wsdl:message>
    <wsdl:message name="GetCompatibleVideoAnalyticsConfigurationsResponse">
        <wsdl:part name="parameters"
element="trt:GetCompatibleVideoAnalyticsConfigurationsResponse"/>
    </wsdl:message>
    <wsdl:message name="GetCompatibleMetadataConfigurationsRequest">
        <wsdl:part name="parameters" element="trt:GetCompatibleMetadataConfigurations"/>
    </wsdl:message>
    <wsdl:message name="GetCompatibleMetadataConfigurationsResponse">
        <wsdl:part name="parameters"
element="trt:GetCompatibleMetadataConfigurationsResponse"/>
    </wsdl:message>
    <wsdl:message name="GetCompatibleAudioOutputConfigurationsRequest">
        <wsdl:part name="parameters"
element="trt:GetCompatibleAudioOutputConfigurations"/>
    </wsdl:message>
    <wsdl:message name="GetCompatibleAudioOutputConfigurationsResponse">
        <wsdl:part name="parameters"
element="trt:GetCompatibleAudioOutputConfigurationsResponse"/>
    </wsdl:message>
    <wsdl:message name="GetCompatibleAudioDecoderConfigurationsRequest">
        <wsdl:part name="parameters"
element="trt:GetCompatibleAudioDecoderConfigurations"/>
    </wsdl:message>
    <wsdl:message name="GetCompatibleAudioDecoderConfigurationsResponse">
        <wsdl:part name="parameters"
element="trt:GetCompatibleAudioDecoderConfigurationsResponse"/>
    </wsdl:message>
    <wsdl:message name="SetVideoSourceConfigurationRequest">
        <wsdl:part name="parameters" element="trt:SetVideoSourceConfiguration"/>
    </wsdl:message>
    <wsdl:message name="SetVideoSourceConfigurationResponse">
        <wsdl:part name="parameters" element="trt:SetVideoSourceConfigurationResponse"/>
    </wsdl:message>
    <wsdl:message name="SetVideoEncoderConfigurationRequest">
        <wsdl:part name="parameters" element="trt:SetVideoEncoderConfiguration"/>
    </wsdl:message>
    <wsdl:message name="SetVideoEncoderConfigurationResponse">
        <wsdl:part name="parameters" element="trt:SetVideoEncoderConfigurationResponse"/>
    </wsdl:message>
    <wsdl:message name="SetAudioSourceConfigurationRequest">
        <wsdl:part name="parameters" element="trt:SetAudioSourceConfiguration"/>
    </wsdl:message>
    <wsdl:message name="SetAudioSourceConfigurationResponse">
        <wsdl:part name="parameters" element="trt:SetAudioSourceConfigurationResponse"/>
    </wsdl:message>
    <wsdl:message name="SetAudioEncoderConfigurationRequest">
        <wsdl:part name="parameters" element="trt:SetAudioEncoderConfiguration"/>
    </wsdl:message>
    <wsdl:message name="SetAudioEncoderConfigurationResponse">
        <wsdl:part name="parameters" element="trt:SetAudioEncoderConfigurationResponse"/>
    </wsdl:message>
    <wsdl:message name="SetVideoAnalyticsConfigurationRequest">
        <wsdl:part name="parameters" element="trt:SetVideoAnalyticsConfiguration"/>
    </wsdl:message>
    <wsdl:message name="SetVideoAnalyticsConfigurationResponse">
        <wsdl:part name="parameters"
element="trt:SetVideoAnalyticsConfigurationResponse"/>
    </wsdl:message>
    <wsdl:message name="SetMetadataConfigurationRequest">
        <wsdl:part name="parameters" element="trt:SetMetadataConfiguration"/>
    </wsdl:message>
    <wsdl:message name="SetMetadataConfigurationResponse">
        <wsdl:part name="parameters" element="trt:SetMetadataConfigurationResponse"/>
    </wsdl:message>

```

```

</wsdl:message>
<wsdl:message name="SetAudioOutputConfigurationRequest">
  <wsdl:part name="parameters" element="trt:SetAudioOutputConfiguration"/>
</wsdl:message>
<wsdl:message name="SetAudioOutputConfigurationResponse">
  <wsdl:part name="parameters" element="trt:SetAudioOutputConfigurationResponse"/>
</wsdl:message>
<wsdl:message name="SetAudioDecoderConfigurationRequest">
  <wsdl:part name="parameters" element="trt:SetAudioDecoderConfiguration"/>
</wsdl:message>
<wsdl:message name="SetAudioDecoderConfigurationResponse">
  <wsdl:part name="parameters" element="trt:SetAudioDecoderConfigurationResponse"/>
</wsdl:message>
<wsdl:message name="GetVideoSourceConfigurationOptionsRequest">
  <wsdl:part name="parameters" element="trt:GetVideoSourceConfigurationOptions"/>
</wsdl:message>
<wsdl:message name="GetVideoSourceConfigurationOptionsResponse">
  <wsdl:part name="parameters"
element="trt:GetVideoSourceConfigurationOptionsResponse"/>
</wsdl:message>
<wsdl:message name="GetVideoEncoderConfigurationOptionsRequest">
  <wsdl:part name="parameters" element="trt:GetVideoEncoderConfigurationOptions"/>
</wsdl:message>
<wsdl:message name="GetVideoEncoderConfigurationOptionsResponse">
  <wsdl:part name="parameters"
element="trt:GetVideoEncoderConfigurationOptionsResponse"/>
</wsdl:message>
<wsdl:message name="GetAudioSourceConfigurationOptionsRequest">
  <wsdl:part name="parameters" element="trt:GetAudioSourceConfigurationOptions"/>
</wsdl:message>
<wsdl:message name="GetAudioSourceConfigurationOptionsResponse">
  <wsdl:part name="parameters"
element="trt:GetAudioSourceConfigurationOptionsResponse"/>
</wsdl:message>
<wsdl:message name="GetAudioEncoderConfigurationOptionsRequest">
  <wsdl:part name="parameters" element="trt:GetAudioEncoderConfigurationOptions"/>
</wsdl:message>
<wsdl:message name="GetAudioEncoderConfigurationOptionsResponse">
  <wsdl:part name="parameters"
element="trt:GetAudioEncoderConfigurationOptionsResponse"/>
</wsdl:message>
<wsdl:message name="GetMetadataConfigurationOptionsRequest">
  <wsdl:part name="parameters" element="trt:GetMetadataConfigurationOptions"/>
</wsdl:message>
<wsdl:message name="GetMetadataConfigurationOptionsResponse">
  <wsdl:part name="parameters"
element="trt:GetMetadataConfigurationOptionsResponse"/>
</wsdl:message>
<wsdl:message name="GetAudioOutputConfigurationOptionsRequest">
  <wsdl:part name="parameters" element="trt:GetAudioOutputConfigurationOptions"/>
</wsdl:message>
<wsdl:message name="GetAudioOutputConfigurationOptionsResponse">
  <wsdl:part name="parameters"
element="trt:GetAudioOutputConfigurationOptionsResponse"/>
</wsdl:message>
<wsdl:message name="GetAudioDecoderConfigurationOptionsRequest">
  <wsdl:part name="parameters" element="trt:GetAudioDecoderConfigurationOptions"/>
</wsdl:message>
<wsdl:message name="GetAudioDecoderConfigurationOptionsResponse">
  <wsdl:part name="parameters"
element="trt:GetAudioDecoderConfigurationOptionsResponse"/>
</wsdl:message>
<wsdl:message name="GetGuaranteedNumberOfVideoEncoderInstancesRequest">
  <wsdl:part name="parameters"
element="trt:GetGuaranteedNumberOfVideoEncoderInstances"/>
</wsdl:message>
<wsdl:message name="GetGuaranteedNumberOfVideoEncoderInstancesResponse">
  <wsdl:part name="parameters"
element="trt:GetGuaranteedNumberOfVideoEncoderInstancesResponse"/>
</wsdl:message>
<wsdl:message name="GetStreamUriRequest">
  <wsdl:part name="parameters" element="trt:GetStreamUri"/>
</wsdl:message>
<wsdl:message name="GetStreamUriResponse">
  <wsdl:part name="parameters" element="trt:GetStreamUriResponse"/>
</wsdl:message>
<wsdl:message name="StartMulticastStreamingRequest">
  <wsdl:part name="parameters" element="trt:StartMulticastStreaming"/>

```



```

</wsdl:message>
<wsdl:message name="StartMulticastStreamingResponse">
  <wsdl:part name="parameters" element="trt:StartMulticastStreamingResponse"/>
</wsdl:message>
<wsdl:message name="StopMulticastStreamingRequest">
  <wsdl:part name="parameters" element="trt:StopMulticastStreaming"/>
</wsdl:message>
<wsdl:message name="StopMulticastStreamingResponse">
  <wsdl:part name="parameters" element="trt:StopMulticastStreamingResponse"/>
</wsdl:message>
<wsdl:message name="SetSynchronizationPointRequest">
  <wsdl:part name="parameters" element="trt:SetSynchronizationPoint"/>
</wsdl:message>
<wsdl:message name="SetSynchronizationPointResponse">
  <wsdl:part name="parameters" element="trt:SetSynchronizationPointResponse"/>
</wsdl:message>
<wsdl:message name="GetSnapshotUriRequest">
  <wsdl:part name="parameters" element="trt:GetSnapshotUri"/>
</wsdl:message>
<wsdl:message name="GetSnapshotUriResponse">
  <wsdl:part name="parameters" element="trt:GetSnapshotUriResponse"/>
</wsdl:message>
<wsdl:message name="GetVideoSourceModesRequest">
  <wsdl:part name="parameters" element="trt:GetVideoSourceModes"/>
</wsdl:message>
<wsdl:message name="GetVideoSourceModesResponse">
  <wsdl:part name="parameters" element="trt:GetVideoSourceModesResponse"/>
</wsdl:message>
<wsdl:message name="SetVideoSourceModeRequest">
  <wsdl:part name="parameters" element="trt:SetVideoSourceMode"/>
</wsdl:message>
<wsdl:message name="SetVideoSourceModeResponse">
  <wsdl:part name="parameters" element="trt:SetVideoSourceModeResponse"/>
</wsdl:message>
<wsdl:message name="GetOSDsRequest">
  <wsdl:part name="parameters" element="trt:GetOSDs"/>
</wsdl:message>
<wsdl:message name="GetOSDsResponse">
  <wsdl:part name="parameters" element="trt:GetOSDsResponse"/>
</wsdl:message>
<wsdl:message name="GetOSDRequest">
  <wsdl:part name="parameters" element="trt:GetOSD"/>
</wsdl:message>
<wsdl:message name="GetOSDResponse">
  <wsdl:part name="parameters" element="trt:GetOSDResponse"/>
</wsdl:message>
<wsdl:message name="GetOSDOptionsRequest">
  <wsdl:part name="parameters" element="trt:GetOSDOptions"/>
</wsdl:message>
<wsdl:message name="GetOSDOptionsResponse">
  <wsdl:part name="parameters" element="trt:GetOSDOptionsResponse"/>
</wsdl:message>
<wsdl:message name="SetOSDRequest">
  <wsdl:part name="parameters" element="trt:SetOSD"/>
</wsdl:message>
<wsdl:message name="SetOSDResponse">
  <wsdl:part name="parameters" element="trt:SetOSDResponse"/>
</wsdl:message>
<wsdl:message name="CreateOSDRequest">
  <wsdl:part name="parameters" element="trt:CreateOSD"/>
</wsdl:message>
<wsdl:message name="CreateOSDResponse">
  <wsdl:part name="parameters" element="trt:CreateOSDResponse"/>
</wsdl:message>
<wsdl:message name="DeleteOSDRequest">
  <wsdl:part name="parameters" element="trt>DeleteOSD"/>
</wsdl:message>
<wsdl:message name="DeleteOSDResponse">
  <wsdl:part name="parameters" element="trt>DeleteOSDResponse"/>
</wsdl:message>
<wsdl:portType name="Media">
  <wsdl:operation name="GetServiceCapabilities">
    <wsdl:input message="trt:GetServiceCapabilitiesRequest"/>
    <wsdl:output message="trt:GetServiceCapabilitiesResponse"/>
  </wsdl:operation>
  <wsdl:operation name="GetVideoSources">
    <wsdl:input message="trt:GetVideoSourcesRequest"/>
    <wsdl:output message="trt:GetVideoSourcesResponse"/>
  </wsdl:operation>

```

```

</wsdl:operation>
<wsdl:operation name="GetAudioSources">
  <wsdl:input message="trt:GetAudioSourcesRequest"/>
  <wsdl:output message="trt:GetAudioSourcesResponse"/>
</wsdl:operation>
<wsdl:operation name="GetAudioOutputs">
  <wsdl:input message="trt:GetAudioOutputsRequest"/>
  <wsdl:output message="trt:GetAudioOutputsResponse"/>
</wsdl:operation>
<wsdl:operation name="CreateProfile">
  <wsdl:input message="trt:CreateProfileRequest"/>
  <wsdl:output message="trt:CreateProfileResponse"/>
</wsdl:operation>
<wsdl:operation name="GetProfile">
  <wsdl:input message="trt:GetProfileRequest"/>
  <wsdl:output message="trt:GetProfileResponse"/>
</wsdl:operation>
<wsdl:operation name="GetProfiles">
  <wsdl:input message="trt:GetProfilesRequest"/>
  <wsdl:output message="trt:GetProfilesResponse"/>
</wsdl:operation>
<wsdl:operation name="AddVideoEncoderConfiguration">
  <wsdl:input message="trt:AddVideoEncoderConfigurationRequest"/>
  <wsdl:output message="trt:AddVideoEncoderConfigurationResponse"/>
</wsdl:operation>
<wsdl:operation name="RemoveVideoEncoderConfiguration">
  <wsdl:input message="trt:RemoveVideoEncoderConfigurationRequest"/>
  <wsdl:output message="trt:RemoveVideoEncoderConfigurationResponse"/>
</wsdl:operation>
<wsdl:operation name="AddVideoSourceConfiguration">
  <wsdl:input message="trt:AddVideoSourceConfigurationRequest"/>
  <wsdl:output message="trt:AddVideoSourceConfigurationResponse"/>
</wsdl:operation>
<wsdl:operation name="RemoveVideoSourceConfiguration">
  <wsdl:input message="trt:RemoveVideoSourceConfigurationRequest"/>
  <wsdl:output message="trt:RemoveVideoSourceConfigurationResponse"/>
</wsdl:operation>
<wsdl:operation name="AddAudioEncoderConfiguration">
  <wsdl:input message="trt:AddAudioEncoderConfigurationRequest"/>
  <wsdl:output message="trt:AddAudioEncoderConfigurationResponse"/>
</wsdl:operation>
<wsdl:operation name="RemoveAudioEncoderConfiguration">
  <wsdl:input message="trt:RemoveAudioEncoderConfigurationRequest"/>
  <wsdl:output message="trt:RemoveAudioEncoderConfigurationResponse"/>
</wsdl:operation>
<wsdl:operation name="AddAudioSourceConfiguration">
  <wsdl:input message="trt:AddAudioSourceConfigurationRequest"/>
  <wsdl:output message="trt:AddAudioSourceConfigurationResponse"/>
</wsdl:operation>
<wsdl:operation name="RemoveAudioSourceConfiguration">
  <wsdl:input message="trt:RemoveAudioSourceConfigurationRequest"/>
  <wsdl:output message="trt:RemoveAudioSourceConfigurationResponse"/>
</wsdl:operation>
<wsdl:operation name="AddPTZConfiguration">
  <wsdl:input message="trt:AddPTZConfigurationRequest"/>
  <wsdl:output message="trt:AddPTZConfigurationResponse"/>
</wsdl:operation>
<wsdl:operation name="RemovePTZConfiguration">
  <wsdl:input message="trt:RemovePTZConfigurationRequest"/>
  <wsdl:output message="trt:RemovePTZConfigurationResponse"/>
</wsdl:operation>
<wsdl:operation name="AddVideoAnalyticsConfiguration">
  <wsdl:input message="trt:AddVideoAnalyticsConfigurationRequest"/>
  <wsdl:output message="trt:AddVideoAnalyticsConfigurationResponse"/>
</wsdl:operation>
<wsdl:operation name="RemoveVideoAnalyticsConfiguration">
  <wsdl:input message="trt:RemoveVideoAnalyticsConfigurationRequest"/>
  <wsdl:output message="trt:RemoveVideoAnalyticsConfigurationResponse"/>
</wsdl:operation>
<wsdl:operation name="AddMetadataConfiguration">
  <wsdl:input message="trt:AddMetadataConfigurationRequest"/>
  <wsdl:output message="trt:AddMetadataConfigurationResponse"/>
</wsdl:operation>
<wsdl:operation name="RemoveMetadataConfiguration">
  <wsdl:input message="trt:RemoveMetadataConfigurationRequest"/>
  <wsdl:output message="trt:RemoveMetadataConfigurationResponse"/>
</wsdl:operation>
<wsdl:operation name="AddAudioOutputConfiguration">

```

```
<wsdl:input message="trt:AddAudioOutputConfigurationRequest"/>
<wsdl:output message="trt:AddAudioOutputConfigurationResponse"/>
</wsdl:operation>
<wsdl:operation name="RemoveAudioOutputConfiguration">
  <wsdl:input message="trt:RemoveAudioOutputConfigurationRequest"/>
  <wsdl:output message="trt:RemoveAudioOutputConfigurationResponse"/>
</wsdl:operation>
<wsdl:operation name="AddAudioDecoderConfiguration">
  <wsdl:input message="trt:AddAudioDecoderConfigurationRequest"/>
  <wsdl:output message="trt:AddAudioDecoderConfigurationResponse"/>
</wsdl:operation>
<wsdl:operation name="RemoveAudioDecoderConfiguration">
  <wsdl:input message="trt:RemoveAudioDecoderConfigurationRequest"/>
  <wsdl:output message="trt:RemoveAudioDecoderConfigurationResponse"/>
</wsdl:operation>
<wsdl:operation name="DeleteProfile">
  <wsdl:input message="trt:DeleteProfileRequest"/>
  <wsdl:output message="trt:DeleteProfileResponse"/>
</wsdl:operation>
<wsdl:operation name="GetVideoSourceConfigurations">
  <wsdl:input message="trt:GetVideoSourceConfigurationsRequest"/>
  <wsdl:output message="trt:GetVideoSourceConfigurationsResponse"/>
</wsdl:operation>
<wsdl:operation name="GetVideoEncoderConfigurations">
  <wsdl:input message="trt:GetVideoEncoderConfigurationsRequest"/>
  <wsdl:output message="trt:GetVideoEncoderConfigurationsResponse"/>
</wsdl:operation>
<wsdl:operation name="GetAudioSourceConfigurations">
  <wsdl:input message="trt:GetAudioSourceConfigurationsRequest"/>
  <wsdl:output message="trt:GetAudioSourceConfigurationsResponse"/>
</wsdl:operation>
<wsdl:operation name="GetAudioEncoderConfigurations">
  <wsdl:input message="trt:GetAudioEncoderConfigurationsRequest"/>
  <wsdl:output message="trt:GetAudioEncoderConfigurationsResponse"/>
</wsdl:operation>
<wsdl:operation name="GetVideoAnalyticsConfigurations">
  <wsdl:input message="trt:GetVideoAnalyticsConfigurationsRequest"/>
  <wsdl:output message="trt:GetVideoAnalyticsConfigurationsResponse"/>
</wsdl:operation>
<wsdl:operation name="GetMetadataConfigurations">
  <wsdl:input message="trt:GetMetadataConfigurationsRequest"/>
  <wsdl:output message="trt:GetMetadataConfigurationsResponse"/>
</wsdl:operation>
<wsdl:operation name="GetAudioOutputConfigurations">
  <wsdl:input message="trt:GetAudioOutputConfigurationsRequest"/>
  <wsdl:output message="trt:GetAudioOutputConfigurationsResponse"/>
</wsdl:operation>
<wsdl:operation name="GetAudioDecoderConfigurations">
  <wsdl:input message="trt:GetAudioDecoderConfigurationsRequest"/>
  <wsdl:output message="trt:GetAudioDecoderConfigurationsResponse"/>
</wsdl:operation>
<wsdl:operation name="GetVideoSourceConfiguration">
  <wsdl:input message="trt:GetVideoSourceConfigurationRequest"/>
  <wsdl:output message="trt:GetVideoSourceConfigurationResponse"/>
</wsdl:operation>
<wsdl:operation name="GetVideoEncoderConfiguration">
  <wsdl:input message="trt:GetVideoEncoderConfigurationRequest"/>
  <wsdl:output message="trt:GetVideoEncoderConfigurationResponse"/>
</wsdl:operation>
<wsdl:operation name="GetAudioSourceConfiguration">
  <wsdl:input message="trt:GetAudioSourceConfigurationRequest"/>
  <wsdl:output message="trt:GetAudioSourceConfigurationResponse"/>
</wsdl:operation>
<wsdl:operation name="GetAudioEncoderConfiguration">
  <wsdl:input message="trt:GetAudioEncoderConfigurationRequest"/>
  <wsdl:output message="trt:GetAudioEncoderConfigurationResponse"/>
</wsdl:operation>
<wsdl:operation name="GetVideoAnalyticsConfiguration">
  <wsdl:input message="trt:GetVideoAnalyticsConfigurationRequest"/>
  <wsdl:output message="trt:GetVideoAnalyticsConfigurationResponse"/>
</wsdl:operation>
<wsdl:operation name="GetMetadataConfiguration">
  <wsdl:input message="trt:GetMetadataConfigurationRequest"/>
  <wsdl:output message="trt:GetMetadataConfigurationResponse"/>
</wsdl:operation>
<wsdl:operation name="GetAudioOutputConfiguration">
  <wsdl:input message="trt:GetAudioOutputConfigurationRequest"/>
  <wsdl:output message="trt:GetAudioOutputConfigurationResponse"/>
</wsdl:operation>
```

```

</wsdl:operation>
<wsdl:operation name="GetAudioDecoderConfiguration">
  <wsdl:input message="trt:GetAudioDecoderConfigurationRequest"/>
  <wsdl:output message="trt:GetAudioDecoderConfigurationResponse"/>
</wsdl:operation>
<wsdl:operation name="GetCompatibleVideoEncoderConfigurations">
  <wsdl:input message="trt:GetCompatibleVideoEncoderConfigurationsRequest"/>
  <wsdl:output message="trt:GetCompatibleVideoEncoderConfigurationsResponse"/>
</wsdl:operation>
<wsdl:operation name="GetCompatibleVideoSourceConfigurations">
  <wsdl:input message="trt:GetCompatibleVideoSourceConfigurationsRequest"/>
  <wsdl:output message="trt:GetCompatibleVideoSourceConfigurationsResponse"/>
</wsdl:operation>
<wsdl:operation name="GetCompatibleAudioEncoderConfigurations">
  <wsdl:input message="trt:GetCompatibleAudioEncoderConfigurationsRequest"/>
  <wsdl:output message="trt:GetCompatibleAudioEncoderConfigurationsResponse"/>
</wsdl:operation>
<wsdl:operation name="GetCompatibleAudioSourceConfigurations">
  <wsdl:input message="trt:GetCompatibleAudioSourceConfigurationsRequest"/>
  <wsdl:output message="trt:GetCompatibleAudioSourceConfigurationsResponse"/>
</wsdl:operation>
<wsdl:operation name="GetCompatibleVideoAnalyticsConfigurations">
  <wsdl:input message="trt:GetCompatibleVideoAnalyticsConfigurationsRequest"/>
  <wsdl:output message="trt:GetCompatibleVideoAnalyticsConfigurationsResponse"/>
</wsdl:operation>
<wsdl:operation name="GetCompatibleMetadataConfigurations">
  <wsdl:input message="trt:GetCompatibleMetadataConfigurationsRequest"/>
  <wsdl:output message="trt:GetCompatibleMetadataConfigurationsResponse"/>
</wsdl:operation>
<wsdl:operation name="GetCompatibleAudioOutputConfigurations">
  <wsdl:input message="trt:GetCompatibleAudioOutputConfigurationsRequest"/>
  <wsdl:output message="trt:GetCompatibleAudioOutputConfigurationsResponse"/>
</wsdl:operation>
<wsdl:operation name="GetCompatibleAudioDecoderConfigurations">
  <wsdl:input message="trt:GetCompatibleAudioDecoderConfigurationsRequest"/>
  <wsdl:output message="trt:GetCompatibleAudioDecoderConfigurationsResponse"/>
</wsdl:operation>
<wsdl:operation name="SetVideoSourceConfiguration">
  <wsdl:input message="trt:SetVideoSourceConfigurationRequest"/>
  <wsdl:output message="trt:SetVideoSourceConfigurationResponse"/>
</wsdl:operation>
<wsdl:operation name="SetVideoEncoderConfiguration">
  <wsdl:input message="trt:SetVideoEncoderConfigurationRequest"/>
  <wsdl:output message="trt:SetVideoEncoderConfigurationResponse"/>
</wsdl:operation>
<wsdl:operation name="SetAudioSourceConfiguration">
  <wsdl:input message="trt:SetAudioSourceConfigurationRequest"/>
  <wsdl:output message="trt:SetAudioSourceConfigurationResponse"/>
</wsdl:operation>
<wsdl:operation name="SetAudioEncoderConfiguration">
  <wsdl:input message="trt:SetAudioEncoderConfigurationRequest"/>
  <wsdl:output message="trt:SetAudioEncoderConfigurationResponse"/>
</wsdl:operation>
<wsdl:operation name="SetVideoAnalyticsConfiguration">
  <wsdl:input message="trt:SetVideoAnalyticsConfigurationRequest"/>
  <wsdl:output message="trt:SetVideoAnalyticsConfigurationResponse"/>
</wsdl:operation>
<wsdl:operation name="SetMetadataConfiguration">
  <wsdl:input message="trt:SetMetadataConfigurationRequest"/>
  <wsdl:output message="trt:SetMetadataConfigurationResponse"/>
</wsdl:operation>
<wsdl:operation name="SetAudioOutputConfiguration">
  <wsdl:input message="trt:SetAudioOutputConfigurationRequest"/>
  <wsdl:output message="trt:SetAudioOutputConfigurationResponse"/>
</wsdl:operation>
<wsdl:operation name="SetAudioDecoderConfiguration">
  <wsdl:input message="trt:SetAudioDecoderConfigurationRequest"/>
  <wsdl:output message="trt:SetAudioDecoderConfigurationResponse"/>
</wsdl:operation>
<wsdl:operation name="GetVideoSourceConfigurationOptions">
  <wsdl:input message="trt:GetVideoSourceConfigurationOptionsRequest"/>
  <wsdl:output message="trt:GetVideoSourceConfigurationOptionsResponse"/>
</wsdl:operation>
<wsdl:operation name="GetVideoEncoderConfigurationOptions">
  <wsdl:input message="trt:GetVideoEncoderConfigurationOptionsRequest"/>
  <wsdl:output message="trt:GetVideoEncoderConfigurationOptionsResponse"/>
</wsdl:operation>
<wsdl:operation name="GetAudioSourceConfigurationOptions">

```

```

        <wsdl:input message="trt:GetAudioSourceConfigurationOptionsRequest"/>
        <wsdl:output message="trt:GetAudioSourceConfigurationOptionsResponse"/>
    </wsdl:operation>
    <wsdl:operation name="GetAudioEncoderConfigurationOptions">
        <wsdl:input message="trt:GetAudioEncoderConfigurationOptionsRequest"/>
        <wsdl:output message="trt:GetAudioEncoderConfigurationOptionsResponse"/>
    </wsdl:operation>
    <wsdl:operation name="GetMetadataConfigurationOptions">
        <wsdl:input message="trt:GetMetadataConfigurationOptionsRequest"/>
        <wsdl:output message="trt:GetMetadataConfigurationOptionsResponse"/>
    </wsdl:operation>
    <wsdl:operation name="GetAudioOutputConfigurationOptions">
        <wsdl:input message="trt:GetAudioOutputConfigurationOptionsRequest"/>
        <wsdl:output message="trt:GetAudioOutputConfigurationOptionsResponse"/>
    </wsdl:operation>
    <wsdl:operation name="GetAudioDecoderConfigurationOptions">
        <wsdl:input message="trt:GetAudioDecoderConfigurationOptionsRequest"/>
        <wsdl:output message="trt:GetAudioDecoderConfigurationOptionsResponse"/>
    </wsdl:operation>
    <wsdl:operation name="GetGuaranteedNumberOfVideoEncoderInstances">
        <wsdl:input message="trt:GetGuaranteedNumberOfVideoEncoderInstancesRequest"/>
        <wsdl:output
message="trt:GetGuaranteedNumberOfVideoEncoderInstancesResponse"/>
    </wsdl:operation>
    <wsdl:operation name="GetStreamUri">
        <wsdl:input message="trt:GetStreamUriRequest"/>
        <wsdl:output message="trt:GetStreamUriResponse"/>
    </wsdl:operation>
    <wsdl:operation name="StartMulticastStreaming">
        <wsdl:input message="trt:StartMulticastStreamingRequest"/>
        <wsdl:output message="trt:StartMulticastStreamingResponse"/>
    </wsdl:operation>
    <wsdl:operation name="StopMulticastStreaming">
        <wsdl:input message="trt:StopMulticastStreamingRequest"/>
        <wsdl:output message="trt:StopMulticastStreamingResponse"/>
    </wsdl:operation>
    <wsdl:operation name="SetSynchronizationPoint">
        <wsdl:input message="trt:SetSynchronizationPointRequest"/>
        <wsdl:output message="trt:SetSynchronizationPointResponse"/>
    </wsdl:operation>
    <wsdl:operation name="GetSnapshotUri">
        <wsdl:input message="trt:GetSnapshotUriRequest"/>
        <wsdl:output message="trt:GetSnapshotUriResponse"/>
    </wsdl:operation>
    <wsdl:operation name="GetVideoSourceModes">
        <wsdl:input message="trt:GetVideoSourceModesRequest"/>
        <wsdl:output message="trt:GetVideoSourceModesResponse"/>
    </wsdl:operation>
    <wsdl:operation name="SetVideoSourceMode">
        <wsdl:input message="trt:SetVideoSourceModeRequest"/>
        <wsdl:output message="trt:SetVideoSourceModeResponse"/>
    </wsdl:operation>
    <wsdl:operation name="GetOSDs">
        <wsdl:input message="trt:GetOSDsRequest"/>
        <wsdl:output message="trt:GetOSDsResponse"/>
    </wsdl:operation>
    <wsdl:operation name="GetOSD">
        <wsdl:input message="trt:GetOSDRequest"/>
        <wsdl:output message="trt:GetOSDResponse"/>
    </wsdl:operation>
    <wsdl:operation name="GetOSDOptions">
        <wsdl:input message="trt:GetOSDOptionsRequest"/>
        <wsdl:output message="trt:GetOSDOptionsResponse"/>
    </wsdl:operation>
    <wsdl:operation name="SetOSD">
        <wsdl:input message="trt:SetOSDRequest"/>
        <wsdl:output message="trt:SetOSDResponse"/>
    </wsdl:operation>
    <wsdl:operation name="CreateOSD">
        <wsdl:input message="trt:CreateOSDRequest"/>
        <wsdl:output message="trt:CreateOSDResponse"/>
    </wsdl:operation>
    <wsdl:operation name="DeleteOSD">
        <wsdl:input message="trt>DeleteOSDRequest"/>
        <wsdl:output message="trt>DeleteOSDResponse"/>
    </wsdl:operation>
</wsdl:portType>
<wsdl:binding name="MediaBinding" type="trt:Media">

```



```

<soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
<wsdl:operation name="GetServiceCapabilities">
  <soap:operation
soapAction="http://www.onvif.org/ver10/media/wsdl/GetServiceCapabilities"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
<wsdl:operation name="GetVideoSources">
  <soap:operation
soapAction="http://www.onvif.org/ver10/media/wsdl/GetVideoSources"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
<wsdl:operation name="GetAudioSources">
  <soap:operation
soapAction="http://www.onvif.org/ver10/media/wsdl/GetAudioSources"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
<wsdl:operation name="GetAudioOutputs">
  <soap:operation
soapAction="http://www.onvif.org/ver10/media/wsdl/GetAudioOutputs"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
<wsdl:operation name="CreateProfile">
  <soap:operation
soapAction="http://www.onvif.org/ver10/media/wsdl/CreateProfile"/>
    <wsdl:input>
      <soap:body parts="parameters" use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body parts="parameters" use="literal"/>
    </wsdl:output>
  </wsdl:operation>
<wsdl:operation name="GetProfile">
  <soap:operation
soapAction="http://www.onvif.org/ver10/media/wsdl/GetProfile"/>
    <wsdl:input>
      <soap:body parts="parameters" use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body parts="parameters" use="literal"/>
    </wsdl:output>
  </wsdl:operation>
<wsdl:operation name="GetProfiles">
  <soap:operation
soapAction="http://www.onvif.org/ver10/media/wsdl/GetProfiles"/>
    <wsdl:input>
      <soap:body parts="parameters" use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body parts="parameters" use="literal"/>
    </wsdl:output>
  </wsdl:operation>
<wsdl:operation name="AddVideoEncoderConfiguration">
  <soap:operation
soapAction="http://www.onvif.org/ver10/media/wsdl/AddVideoEncoderConfiguration"/>
    <wsdl:input>
      <soap:body parts="parameters" use="literal"/>
    </wsdl:input>
    <wsdl:output>

```

```
<soap:body parts="parameters" use="literal"/>
</wsdl:output>
</wsdl:operation>
<wsdl:operation name="AddVideoSourceConfiguration">
  <soap:operation
soapAction="http://www.onvif.org/ver10/media/wsdl/AddVideoSourceConfiguration"/>
  <wsdl:input>
    <soap:body parts="parameters" use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body parts="parameters" use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="AddAudioEncoderConfiguration">
  <soap:operation
soapAction="http://www.onvif.org/ver10/media/wsdl/AddAudioEncoderConfiguration"/>
  <wsdl:input>
    <soap:body parts="parameters" use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body parts="parameters" use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="AddAudioSourceConfiguration">
  <soap:operation
soapAction="http://www.onvif.org/ver10/media/wsdl/AddAudioSourceConfiguration"/>
  <wsdl:input>
    <soap:body parts="parameters" use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body parts="parameters" use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="AddPTZConfiguration">
  <soap:operation
soapAction="http://www.onvif.org/ver10/media/wsdl/AddPTZConfiguration"/>
  <wsdl:input>
    <soap:body parts="parameters" use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body parts="parameters" use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="AddVideoAnalyticsConfiguration">
  <soap:operation
soapAction="http://www.onvif.org/ver10/media/wsdl/AddVideoAnalyticsConfiguration"/>
  <wsdl:input>
    <soap:body parts="parameters" use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body parts="parameters" use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="AddMetadataConfiguration">
  <soap:operation
soapAction="http://www.onvif.org/ver10/media/wsdl/AddMetadataConfiguration"/>
  <wsdl:input>
    <soap:body parts="parameters" use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body parts="parameters" use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="AddAudioOutputConfiguration">
  <soap:operation
soapAction="http://www.onvif.org/ver10/media/wsdl/AddAudioOutputConfiguration"/>
  <wsdl:input>
    <soap:body parts="parameters" use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body parts="parameters" use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="AddAudioDecoderConfiguration">
  <soap:operation
soapAction="http://www.onvif.org/ver10/media/wsdl/AddAudioDecoderConfiguration"/>
  <wsdl:input>
    <soap:body parts="parameters" use="literal"/>
```

```

        </wsdl:input>
        <wsdl:output>
            <soap:body parts="parameters" use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="RemoveVideoEncoderConfiguration">
        <soap:operation
soapAction="http://www.onvif.org/ver10/media/wsdl/RemoveVideoEncoderConfiguration"/>
        <wsdl:input>
            <soap:body parts="parameters" use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body parts="parameters" use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="RemoveVideoSourceConfiguration">
        <soap:operation
soapAction="http://www.onvif.org/ver10/media/wsdl/RemoveVideoSourceConfiguration"/>
        <wsdl:input>
            <soap:body parts="parameters" use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body parts="parameters" use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="RemoveAudioEncoderConfiguration">
        <soap:operation
soapAction="http://www.onvif.org/ver10/media/wsdl/RemoveAudioEncoderConfiguration"/>
        <wsdl:input>
            <soap:body parts="parameters" use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body parts="parameters" use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="RemoveAudioSourceConfiguration">
        <soap:operation
soapAction="http://www.onvif.org/ver10/media/wsdl/RemoveAudioSourceConfiguration"/>
        <wsdl:input>
            <soap:body parts="parameters" use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body parts="parameters" use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="RemovePTZConfiguration">
        <soap:operation
soapAction="http://www.onvif.org/ver10/media/wsdl/RemovePTZConfiguration"/>
        <wsdl:input>
            <soap:body parts="parameters" use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body parts="parameters" use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="RemoveVideoAnalyticsConfiguration">
        <soap:operation
soapAction="http://www.onvif.org/ver10/media/wsdl/RemoveVideoAnalyticsConfiguration"/>
        <wsdl:input>
            <soap:body parts="parameters" use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body parts="parameters" use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="RemoveMetadataConfiguration">
        <soap:operation
soapAction="http://www.onvif.org/ver10/media/wsdl/RemoveMetadataConfiguration"/>
        <wsdl:input>
            <soap:body parts="parameters" use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body parts="parameters" use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="RemoveAudioOutputConfiguration">
        <soap:operation
soapAction="http://www.onvif.org/ver10/media/wsdl/RemoveAudioOutputConfiguration"/>

```



```
<wsdl:input>
  <soap:body parts="parameters" use="literal"/>
</wsdl:input>
<wsdl:output>
  <soap:body parts="parameters" use="literal"/>
</wsdl:output>
</wsdl:operation>
<wsdl:operation name="RemoveAudioDecoderConfiguration">
  <soap:operation
soapAction="http://www.onvif.org/ver10/media/wsdl/RemoveAudioDecoderConfiguration"/>
  <wsdl:input>
    <soap:body parts="parameters" use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body parts="parameters" use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="DeleteProfile">
  <soap:operation
soapAction="http://www.onvif.org/ver10/media/wsdl/DeleteProfile"/>
  <wsdl:input>
    <soap:body parts="parameters" use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body parts="parameters" use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetVideoSourceConfigurations">
  <soap:operation
soapAction="http://www.onvif.org/ver10/media/wsdl/GetVideoSourceConfigurations"/>
  <wsdl:input>
    <soap:body parts="parameters" use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body parts="parameters" use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetVideoEncoderConfigurations">
  <soap:operation
soapAction="http://www.onvif.org/ver10/media/wsdl/GetVideoEncoderConfigurations"/>
  <wsdl:input>
    <soap:body parts="parameters" use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body parts="parameters" use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetAudioSourceConfigurations">
  <soap:operation
soapAction="http://www.onvif.org/ver10/media/wsdl/GetAudioSourceConfigurations"/>
  <wsdl:input>
    <soap:body parts="parameters" use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body parts="parameters" use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetAudioEncoderConfigurations">
  <soap:operation
soapAction="http://www.onvif.org/ver10/media/wsdl/GetAudioEncoderConfigurations"/>
  <wsdl:input>
    <soap:body parts="parameters" use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body parts="parameters" use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetVideoAnalyticsConfigurations">
  <soap:operation
soapAction="http://www.onvif.org/ver10/media/wsdl/GetVideoAnalyticsConfigurations"/>
  <wsdl:input>
    <soap:body parts="parameters" use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body parts="parameters" use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetMetadataConfigurations">
```

```

        <soap:operation
soapAction="http://www.onvif.org/ver10/media/wsd/GetMetadataConfigurations"/>
        <wsdl:input>
            <soap:body parts="parameters" use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body parts="parameters" use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetAudioOutputConfigurations">
        <soap:operation
soapAction="http://www.onvif.org/ver10/media/wsd/GetAudioOutputConfigurations"/>
        <wsdl:input>
            <soap:body parts="parameters" use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body parts="parameters" use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetAudioDecoderConfigurations">
        <soap:operation
soapAction="http://www.onvif.org/ver10/media/wsd/GetAudioDecoderConfigurations"/>
        <wsdl:input>
            <soap:body parts="parameters" use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body parts="parameters" use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetVideoSourceConfiguration">
        <soap:operation
soapAction="http://www.onvif.org/ver10/media/wsd/GetVideoSourceConfiguration"/>
        <wsdl:input>
            <soap:body parts="parameters" use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body parts="parameters" use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetVideoEncoderConfiguration">
        <soap:operation
soapAction="http://www.onvif.org/ver10/media/wsd/GetVideoEncoderConfiguration"/>
        <wsdl:input>
            <soap:body parts="parameters" use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body parts="parameters" use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetAudioSourceConfiguration">
        <soap:operation
soapAction="http://www.onvif.org/ver10/media/wsd/GetAudioSourceConfiguration"/>
        <wsdl:input>
            <soap:body parts="parameters" use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body parts="parameters" use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetAudioEncoderConfiguration">
        <soap:operation
soapAction="http://www.onvif.org/ver10/media/wsd/GetAudioEncoderConfiguration"/>
        <wsdl:input>
            <soap:body parts="parameters" use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body parts="parameters" use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetVideoAnalyticsConfiguration">
        <soap:operation
soapAction="http://www.onvif.org/ver10/media/wsd/GetVideoAnalyticsConfiguration"/>
        <wsdl:input>
            <soap:body parts="parameters" use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body parts="parameters" use="literal"/>
        </wsdl:output>
    </wsdl:operation>

```

```

</wsdl:operation>
<wsdl:operation name="GetMetadataConfiguration">
  <soap:operation
soapAction="http://www.onvif.org/ver10/media/wsdl/GetMetadataConfiguration"/>
    <wsdl:input>
      <soap:body parts="parameters" use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body parts="parameters" use="literal"/>
    </wsdl:output>
  </wsdl:operation>
<wsdl:operation name="GetAudioOutputConfiguration">
  <soap:operation
soapAction="http://www.onvif.org/ver10/media/wsdl/GetAudioOutputConfiguration"/>
    <wsdl:input>
      <soap:body parts="parameters" use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body parts="parameters" use="literal"/>
    </wsdl:output>
  </wsdl:operation>
<wsdl:operation name="GetAudioDecoderConfiguration">
  <soap:operation
soapAction="http://www.onvif.org/ver10/media/wsdl/GetAudioDecoderConfiguration"/>
    <wsdl:input>
      <soap:body parts="parameters" use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body parts="parameters" use="literal"/>
    </wsdl:output>
  </wsdl:operation>
<wsdl:operation name="GetCompatibleVideoEncoderConfigurations">
  <soap:operation
soapAction="http://www.onvif.org/ver10/media/wsdl/GetCompatibleVideoEncoderConfigurations"/>
    <wsdl:input>
      <soap:body parts="parameters" use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body parts="parameters" use="literal"/>
    </wsdl:output>
  </wsdl:operation>
<wsdl:operation name="GetCompatibleVideoSourceConfigurations">
  <soap:operation
soapAction="http://www.onvif.org/ver10/media/wsdl/GetCompatibleVideoSourceConfigurations"/>
    <wsdl:input>
      <soap:body parts="parameters" use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body parts="parameters" use="literal"/>
    </wsdl:output>
  </wsdl:operation>
<wsdl:operation name="GetCompatibleAudioEncoderConfigurations">
  <soap:operation
soapAction="http://www.onvif.org/ver10/media/wsdl/GetCompatibleAudioEncoderConfigurations"/>
    <wsdl:input>
      <soap:body parts="parameters" use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body parts="parameters" use="literal"/>
    </wsdl:output>
  </wsdl:operation>
<wsdl:operation name="GetCompatibleAudioSourceConfigurations">
  <soap:operation
soapAction="http://www.onvif.org/ver10/media/wsdl/GetCompatibleAudioSourceConfigurations"/>
    <wsdl:input>
      <soap:body parts="parameters" use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body parts="parameters" use="literal"/>
    </wsdl:output>
  </wsdl:operation>
<wsdl:operation name="GetCompatibleVideoAnalyticsConfigurations">
  <soap:operation
soapAction="http://www.onvif.org/ver10/media/wsdl/GetCompatibleVideoAnalyticsConfigurations"/>
    <wsdl:input>
      <soap:body parts="parameters" use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body parts="parameters" use="literal"/>
    </wsdl:output>
  </wsdl:operation>

```

```

ions"/>
    <wsdl:input>
        <soap:body parts="parameters" use="literal"/>
    </wsdl:input>
    <wsdl:output>
        <soap:body parts="parameters" use="literal"/>
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetCompatibleMetadataConfigurations">
    <soap:operation
soapAction="http://www.onvif.org/ver10/media/wsdl/GetCompatibleMetadataConfigurations"/>
    <wsdl:input>
        <soap:body parts="parameters" use="literal"/>
    </wsdl:input>
    <wsdl:output>
        <soap:body parts="parameters" use="literal"/>
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetCompatibleAudioOutputConfigurations">
    <soap:operation
soapAction="http://www.onvif.org/ver10/media/wsdl/GetCompatibleAudioOutputConfiguration
s"/>
    <wsdl:input>
        <soap:body parts="parameters" use="literal"/>
    </wsdl:input>
    <wsdl:output>
        <soap:body parts="parameters" use="literal"/>
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetCompatibleAudioDecoderConfigurations">
    <soap:operation
soapAction="http://www.onvif.org/ver10/media/wsdl/GetCompatibleAudioDecoderConfiguratio
ns"/>
    <wsdl:input>
        <soap:body parts="parameters" use="literal"/>
    </wsdl:input>
    <wsdl:output>
        <soap:body parts="parameters" use="literal"/>
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="SetVideoSourceConfiguration">
    <soap:operation
soapAction="http://www.onvif.org/ver10/media/wsdl/SetVideoSourceConfiguration"/>
    <wsdl:input>
        <soap:body parts="parameters" use="literal"/>
    </wsdl:input>
    <wsdl:output>
        <soap:body parts="parameters" use="literal"/>
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="SetVideoEncoderConfiguration">
    <soap:operation
soapAction="http://www.onvif.org/ver10/media/wsdl/SetVideoEncoderConfiguration"/>
    <wsdl:input>
        <soap:body parts="parameters" use="literal"/>
    </wsdl:input>
    <wsdl:output>
        <soap:body parts="parameters" use="literal"/>
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="SetAudioSourceConfiguration">
    <soap:operation
soapAction="http://www.onvif.org/ver10/media/wsdl/SetAudioSourceConfiguration"/>
    <wsdl:input>
        <soap:body parts="parameters" use="literal"/>
    </wsdl:input>
    <wsdl:output>
        <soap:body parts="parameters" use="literal"/>
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="SetAudioEncoderConfiguration">
    <soap:operation
soapAction="http://www.onvif.org/ver10/media/wsdl/SetAudioEncoderConfiguration"/>
    <wsdl:input>
        <soap:body parts="parameters" use="literal"/>
    </wsdl:input>
    <wsdl:output>

```

```

        <soap:body parts="parameters" use="literal"/>
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="SetVideoAnalyticsConfiguration">
    <soap:operation
soapAction="http://www.onvif.org/ver10/media/wsdl/SetVideoAnalyticsConfiguration"/>
    <wsdl:input>
        <soap:body parts="parameters" use="literal"/>
    </wsdl:input>
    <wsdl:output>
        <soap:body parts="parameters" use="literal"/>
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="SetMetadataConfiguration">
    <soap:operation
soapAction="http://www.onvif.org/ver10/media/wsdl/SetMetadataConfiguration"/>
    <wsdl:input>
        <soap:body parts="parameters" use="literal"/>
    </wsdl:input>
    <wsdl:output>
        <soap:body parts="parameters" use="literal"/>
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="SetAudioOutputConfiguration">
    <soap:operation
soapAction="http://www.onvif.org/ver10/media/wsdl/SetAudioOutputConfiguration"/>
    <wsdl:input>
        <soap:body parts="parameters" use="literal"/>
    </wsdl:input>
    <wsdl:output>
        <soap:body parts="parameters" use="literal"/>
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="SetAudioDecoderConfiguration">
    <soap:operation
soapAction="http://www.onvif.org/ver10/media/wsdl/SetAudioDecoderConfiguration"/>
    <wsdl:input>
        <soap:body parts="parameters" use="literal"/>
    </wsdl:input>
    <wsdl:output>
        <soap:body parts="parameters" use="literal"/>
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetVideoSourceConfigurationOptions">
    <soap:operation
soapAction="http://www.onvif.org/ver10/media/wsdl/GetVideoSourceConfigurationOptions"/>
    <wsdl:input>
        <soap:body parts="parameters" use="literal"/>
    </wsdl:input>
    <wsdl:output>
        <soap:body parts="parameters" use="literal"/>
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetVideoEncoderConfigurationOptions">
    <soap:operation
soapAction="http://www.onvif.org/ver10/media/wsdl/GetVideoEncoderConfigurationOptions"/>
>
    <wsdl:input>
        <soap:body parts="parameters" use="literal"/>
    </wsdl:input>
    <wsdl:output>
        <soap:body parts="parameters" use="literal"/>
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetAudioSourceConfigurationOptions">
    <soap:operation
soapAction="http://www.onvif.org/ver10/media/wsdl/GetAudioSourceConfigurationOptions"/>
    <wsdl:input>
        <soap:body parts="parameters" use="literal"/>
    </wsdl:input>
    <wsdl:output>
        <soap:body parts="parameters" use="literal"/>
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetAudioEncoderConfigurationOptions">
    <soap:operation
soapAction="http://www.onvif.org/ver10/media/wsdl/GetAudioEncoderConfigurationOptions"/>
>

```

```

        <wsdl:input>
            <soap:body parts="parameters" use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body parts="parameters" use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetMetadataConfigurationOptions">
        <soap:operation
soapAction="http://www.onvif.org/ver10/media/wsdl/GetMetadataConfigurationOptions"/>
        <wsdl:input>
            <soap:body parts="parameters" use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body parts="parameters" use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetAudioOutputConfigurationOptions">
        <soap:operation
soapAction="http://www.onvif.org/ver10/media/wsdl/GetAudioOutputConfigurationOptions"/>
        <wsdl:input>
            <soap:body parts="parameters" use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body parts="parameters" use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetAudioDecoderConfigurationOptions">
        <soap:operation
soapAction="http://www.onvif.org/ver10/media/wsdl/GetAudioDecoderConfigurationOptions"/
>
        <wsdl:input>
            <soap:body parts="parameters" use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body parts="parameters" use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetGuaranteedNumberOfVideoEncoderInstances">
        <soap:operation
soapAction="http://www.onvif.org/ver10/media/wsdl/GetGuaranteedNumberOfVideoEncoderInst
ances"/>
        <wsdl:input>
            <soap:body parts="parameters" use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body parts="parameters" use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetStreamUri">
        <soap:operation
soapAction="http://www.onvif.org/ver10/media/wsdl/GetStreamUri"/>
        <wsdl:input>
            <soap:body parts="parameters" use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body parts="parameters" use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="StartMulticastStreaming">
        <soap:operation
soapAction="http://www.onvif.org/ver10/media/wsdl/StartMulticastStreaming"/>
        <wsdl:input>
            <soap:body parts="parameters" use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body parts="parameters" use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="StopMulticastStreaming">
        <soap:operation
soapAction="http://www.onvif.org/ver10/media/wsdl/StopMulticastStreaming"/>
        <wsdl:input>
            <soap:body parts="parameters" use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body parts="parameters" use="literal"/>
        </wsdl:output>
    </wsdl:operation>

```

```
</wsdl:operation>
<wsdl:operation name="SetSynchronizationPoint">
  <soap:operation
soapAction="http://www.onvif.org/ver10/media/wsdl/SetSynchronizationPoint"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
<wsdl:operation name="GetSnapshotUri">
  <soap:operation
soapAction="http://www.onvif.org/ver10/media/wsdl/GetSnapshotUri"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
<wsdl:operation name="GetVideoSourceModes">
  <soap:operation
soapAction="http://www.onvif.org/ver10/media/wsdl/GetVideoSourceModes"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
<wsdl:operation name="SetVideoSourceMode">
  <soap:operation
soapAction="http://www.onvif.org/ver10/media/wsdl/SetVideoSourceMode"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
<wsdl:operation name="GetOSDs">
  <soap:operation soapAction="http://www.onvif.org/ver10/media/wsdl/GetOSDs"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
<wsdl:operation name="GetOSD">
  <soap:operation soapAction="http://www.onvif.org/ver10/media/wsdl/GetOSD"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
<wsdl:operation name="GetOSDOptions">
  <soap:operation
soapAction="http://www.onvif.org/ver10/media/wsdl/GetOSDOptions"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
<wsdl:operation name="SetOSD">
  <soap:operation soapAction="http://www.onvif.org/ver10/media/wsdl/SetOSD"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
```

```
<wsdl:operation name="CreateOSD">
  <soap:operation soapAction="http://www.onvif.org/ver10/media/wsdl/CreateOSD"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="DeleteOSD">
  <soap:operation soapAction="http://www.onvif.org/ver10/media/wsdl/DeleteOSD"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
</wsdl:binding>
</wsdl:definitions>
```

IECNORM.COM : Click to view the full PDF of IEC 62676-2-31:2019

F.4 Media 2

```
<?xml version="1.0" encoding="utf-8"?><?xml-stylesheet type="text/xsl"
href="../../../ver20/util/onvif-wsdl-viewer.xsl"?><wsdl:definitions
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap12/"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:tr2="http://www.onvif.org/ver20/media/wsdl"
targetNamespace="http://www.onvif.org/ver20/media/wsdl">
  <wsdl:types>
    <xs:schema xmlns:tt="http://www.onvif.org/ver10/schema"
targetNamespace="http://www.onvif.org/ver20/media/wsdl" elementFormDefault="qualified"
version="16.12">
      <xs:import namespace="http://www.onvif.org/ver10/schema"
schemaLocation="../../../ver10/schema/onvif.xsd"/>
      <xs:element name="GetServiceCapabilities">
        <xs:complexType>
          <xs:sequence/>
        </xs:complexType>
      </xs:element>
      <xs:element name="GetServiceCapabilitiesResponse">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Capabilities" type="tr2:Capabilities2"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="Capabilities2">
        <xs:sequence>
          <xs:element name="ProfileCapabilities" type="tr2:ProfileCapabilities"/>
          <xs:element name="StreamingCapabilities"
type="tr2:StreamingCapabilities"/>
          <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="SnapshotUri" type="xs:boolean"/>
        <xs:attribute name="Rotation" type="xs:boolean"/>
        <xs:attribute name="VideoSourceMode" type="xs:boolean"/>
        <xs:attribute name="OSD" type="xs:boolean"/>
        <xs:attribute name="Mask" type="xs:boolean"/>
        <xs:anyAttribute processContents="lax"/>
      </xs:complexType>
      <xs:element name="Capabilities" type="tr2:Capabilities2"/>
      <xs:complexType name="ProfileCapabilities">
        <xs:sequence>
          <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="MaximumNumberOfProfiles" type="xs:int"/>
        <xs:attribute name="ConfigurationsSupported" type="tt:StringAttrList"/>
        <xs:anyAttribute processContents="lax"/>
      </xs:complexType>
      <xs:complexType name="StreamingCapabilities">
        <xs:sequence>
          <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="RTSPStreaming" type="xs:boolean"/>
        <xs:attribute name="RTPMulticast" type="xs:boolean"/>
        <xs:attribute name="RTP_RTSP_TCP" type="xs:boolean"/>
        <xs:attribute name="NonAggregateControl" type="xs:boolean"/>
        <xs:attribute name="RTSPWebsocketUri" type="xs:anyURI"/>
        <xs:anyAttribute processContents="lax"/>
      </xs:complexType>
      <xs:simpleType name="ConfigurationEnumeration">
        <xs:restriction base="xs:string">
          <xs:enumeration value="All"/>
          <xs:enumeration value="VideoSource"/>
          <xs:enumeration value="VideoEncoder"/>
          <xs:enumeration value="AudioSource"/>
          <xs:enumeration value="AudioEncoder"/>
          <xs:enumeration value="AudioOutput"/>
          <xs:enumeration value="AudioDecoder"/>
          <xs:enumeration value="Metadata"/>
          <xs:enumeration value="Analytics"/>
          <xs:enumeration value="PTZ"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:schema>
  </wsdl:types>
</wsdl:definitions>
```

```

        </xs:restriction>
    </xs:simpleType>
    <xs:element name="ConfigurationEnumeration"
type="tr2:ConfigurationEnumeration"/>
    <xs:complexType name="ConfigurationRef">
        <xs:sequence>
            <xs:element name="Type" type="xs:string"/>
            <xs:element name="Token" type="tt:ReferenceToken" minOccurs="0"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="ConfigurationSet">
        <xs:sequence>
            <xs:element name="VideoSource" type="tt:VideoSourceConfiguration"
minOccurs="0"/>
            <xs:element name="AudioSource" type="tt:AudioSourceConfiguration"
minOccurs="0"/>
            <xs:element name="VideoEncoder" type="tt:VideoEncoder2Configuration"
minOccurs="0"/>
            <xs:element name="AudioEncoder" type="tt:AudioEncoder2Configuration"
minOccurs="0"/>
            <xs:element name="Analytics" type="tt:VideoAnalyticsConfiguration"
minOccurs="0"/>
            <xs:element name="PTZ" type="tt:PTZConfiguration" minOccurs="0"/>
            <xs:element name="Metadata" type="tt:MetadataConfiguration"
minOccurs="0"/>
            <xs:element name="AudioOutput" type="tt:AudioOutputConfiguration"
minOccurs="0"/>
            <xs:element name="AudioDecoder" type="tt:AudioDecoderConfiguration"
minOccurs="0"/>
            <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:anyAttribute processContents="lax"/>
    </xs:complexType>
    <xs:complexType name="MediaProfile">
        <xs:sequence>
            <xs:element name="Name" type="tt:Name"/>
            <xs:element name="Configurations" type="tr2:ConfigurationSet"
minOccurs="0"/>
        </xs:sequence>
        <xs:attribute name="token" type="tt:ReferenceToken" use="required"/>
        <xs:attribute name="fixed" type="xs:boolean"/>
        <xs:anyAttribute processContents="lax"/>
    </xs:complexType>
    <xs:element name="CreateProfile">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="Name" type="tt:Name"/>
                <xs:element name="Configuration" type="tr2:ConfigurationRef"
minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="CreateProfileResponse">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="Token" type="tt:ReferenceToken"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="GetProfiles">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="Token" type="tt:ReferenceToken" minOccurs="0"/>
                <xs:element name="Type" type="xs:string" maxOccurs="unbounded"
minOccurs="0"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="GetProfilesResponse">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="Profiles" type="tr2:MediaProfile" minOccurs="0"
maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="AddConfiguration">

```

```

        <xs:complexType>
          <xs:sequence>
            <xs:element name="ProfileToken" type="tt:ReferenceToken"/>
            <xs:element name="Name" type="tt:Name" minOccurs="0"/>
            <xs:element name="Configuration" type="tr2:ConfigurationRef"
minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="AddConfigurationResponse">
        <xs:complexType>
          <xs:sequence/>
        </xs:complexType>
      </xs:element>
      <xs:element name="RemoveConfiguration">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="ProfileToken" type="tt:ReferenceToken"/>
            <xs:element name="Configuration" type="tr2:ConfigurationRef"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="RemoveConfigurationResponse">
        <xs:complexType>
          <xs:sequence/>
        </xs:complexType>
      </xs:element>
      <xs:element name="DeleteProfile">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Token" type="tt:ReferenceToken"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="DeleteProfileResponse">
        <xs:complexType>
          <xs:sequence/>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="GetConfiguration">
        <xs:sequence>
          <xs:element name="ConfigurationToken" type="tt:ReferenceToken"
minOccurs="0"/>
          <xs:element name="ProfileToken" type="tt:ReferenceToken" minOccurs="0"/>
        </xs:sequence>
      </xs:complexType>
      <xs:element name="GetVideoEncoderConfigurations" type="tr2:GetConfiguration"/>
      <xs:element name="GetVideoEncoderConfigurationsResponse">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Configurations"
type="tt:VideoEncoder2Configuration" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="GetVideoSourceConfigurations" type="tr2:GetConfiguration"/>
      <xs:element name="GetVideoSourceConfigurationsResponse">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Configurations" type="tt:VideoSourceConfiguration"
minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="GetAudioEncoderConfigurations" type="tr2:GetConfiguration"/>
      <xs:element name="GetAudioEncoderConfigurationsResponse">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Configurations"
type="tt:AudioEncoder2Configuration" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="GetAudioSourceConfigurations" type="tr2:GetConfiguration"/>
      <xs:element name="GetAudioSourceConfigurationsResponse">
        <xs:complexType>
          <xs:sequence>

```

```

        <xs:element name="Configurations" type="tt:AudioSourceConfiguration"
minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="GetAnalyticsConfigurations" type="tr2:GetConfiguration"/>
<xs:element name="GetAnalyticsConfigurationsResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Configurations"
type="tt:VideoAnalyticsConfiguration" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="GetMetadataConfigurations" type="tr2:GetConfiguration"/>
<xs:element name="GetMetadataConfigurationsResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Configurations" type="tt:MetadataConfiguration"
minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="GetAudioOutputConfigurations" type="tr2:GetConfiguration"/>
<xs:element name="GetAudioOutputConfigurationsResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Configurations" type="tt:AudioOutputConfiguration"
minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="GetAudioDecoderConfigurations" type="tr2:GetConfiguration"/>
<xs:element name="GetAudioDecoderConfigurationsResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Configurations" type="tt:AudioDecoderConfiguration"
minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="SetVideoEncoderConfiguration">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Configuration"
type="tt:VideoEncoder2Configuration"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:complexType name="SetConfigurationResponse">
    <xs:sequence/>
</xs:complexType>
<xs:element name="SetVideoEncoderConfigurationResponse"
type="tr2:SetConfigurationResponse"/>
<xs:element name="SetVideoSourceConfiguration">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Configuration" type="tt:VideoSourceConfiguration"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="SetVideoSourceConfigurationResponse"
type="tr2:SetConfigurationResponse"/>
<xs:element name="SetAudioEncoderConfiguration">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Configuration"
type="tt:AudioEncoder2Configuration"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="SetAudioEncoderConfigurationResponse"
type="tr2:SetConfigurationResponse"/>
<xs:element name="SetAudioSourceConfiguration">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Configuration" type="tt:AudioSourceConfiguration"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>

```

```

        </xs:complexType>
      </xs:element>
      <xs:element name="SetAudioSourceConfigurationResponse"
type="tr2:SetConfigurationResponse"/>
      <xs:element name="SetMetadataConfiguration">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Configuration" type="tt:MetadataConfiguration"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="SetMetadataConfigurationResponse"
type="tr2:SetConfigurationResponse"/>
      <xs:element name="SetAudioOutputConfiguration">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Configuration" type="tt:AudioOutputConfiguration"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="SetAudioOutputConfigurationResponse"
type="tr2:SetConfigurationResponse"/>
      <xs:element name="SetAudioDecoderConfiguration">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Configuration"
type="tt:AudioDecoderConfiguration"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="SetAudioDecoderConfigurationResponse"
type="tr2:SetConfigurationResponse"/>
      <xs:element name="GetVideoSourceConfigurationOptions"
type="tr2:GetConfiguration"/>
      <xs:element name="GetVideoSourceConfigurationOptionsResponse">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Options"
type="tt:VideoSourceConfigurationOptions"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="GetVideoEncoderConfigurationOptions"
type="tr2:GetConfiguration"/>
      <xs:element name="GetVideoEncoderConfigurationOptionsResponse">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Options"
type="tt:VideoEncoder2ConfigurationOptions" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="GetAudioSourceConfigurationOptions"
type="tr2:GetConfiguration"/>
      <xs:element name="GetAudioSourceConfigurationOptionsResponse">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Options"
type="tt:AudioSourceConfigurationOptions"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="GetAudioEncoderConfigurationOptions"
type="tr2:GetConfiguration"/>
      <xs:element name="GetAudioEncoderConfigurationOptionsResponse">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Options"
type="tt:AudioEncoder2ConfigurationOptions" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="GetMetadataConfigurationOptions"
type="tr2:GetConfiguration"/>
      <xs:element name="GetMetadataConfigurationOptionsResponse">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Options" type="tt:MetadataConfigurationOptions"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>

```

```

        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="GetAudioOutputConfigurationOptions"
type="tr2:GetConfiguration"/>
    <xs:element name="GetAudioOutputConfigurationOptionsResponse">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="Options"
type="tt:AudioOutputConfigurationOptions"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="GetAudioDecoderConfigurationOptions"
type="tr2:GetConfiguration"/>
    <xs:element name="GetAudioDecoderConfigurationOptionsResponse">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="Options"
type="tt:AudioEncoder2ConfigurationOptions" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:simpleType name="TransportProtocol">
      <xs:restriction base="xs:string">
        <xs:enumeration value="RtspUnicast"/>
        <xs:enumeration value="RtspMulticast"/>
        <xs:enumeration value="RTSP"/>
        <xs:enumeration value="RtspOverHttp"/>
      </xs:restriction>
    </xs:simpleType>
    <xs:element name="GetVideoEncoderInstances">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="ConfigurationToken" type="tt:ReferenceToken"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:complexType name="EncoderInstance">
      <xs:sequence>
        <xs:element name="Encoding" type="xs:string"/>
        <xs:element name="Number" type="xs:int"/>
      </xs:sequence>
      <xs:anyAttribute processContents="lax"/>
    </xs:complexType>
    <xs:complexType name="EncoderInstanceInfo">
      <xs:sequence>
        <xs:element name="Codec" type="tr2:EncoderInstance" minOccurs="0"
maxOccurs="unbounded"/>
        <xs:element name="Total" type="xs:int"/>
        <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:anyAttribute processContents="lax"/>
    </xs:complexType>
    <xs:element name="GetVideoEncoderInstancesResponse">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="Info" type="tr2:EncoderInstanceInfo"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="GetStreamUri">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="Protocol" type="xs:string"/>
          <xs:element name="ProfileToken" type="tt:ReferenceToken"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="GetStreamUriResponse">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="Uri" type="xs:anyURI"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="SetSynchronizationPoint">

```

```

        <xs:complexType>
          <xs:sequence>
            <xs:element name="ProfileToken" type="tt:ReferenceToken"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="SetSynchronizationPointResponse">
        <xs:complexType>
          <xs:sequence/>
        </xs:complexType>
      </xs:element>
      <xs:element name="GetSnapshotUri">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="ProfileToken" type="tt:ReferenceToken"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="GetSnapshotUriResponse">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Uri" type="xs:anyURI"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="StartStopMulticastStreaming">
        <xs:sequence>
          <xs:element name="ProfileToken" type="tt:ReferenceToken"/>
        </xs:sequence>
      </xs:complexType>
      <xs:element name="StartMulticastStreaming"
type="tr2:StartStopMulticastStreaming"/>
      <xs:element name="StartMulticastStreamingResponse"
type="tr2:SetConfigurationResponse"/>
      <xs:element name="StopMulticastStreaming"
type="tr2:StartStopMulticastStreaming"/>
      <xs:element name="StopMulticastStreamingResponse"
type="tr2:SetConfigurationResponse"/>
      <xs:element name="GetVideoSourceModes">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="VideoSourceToken" type="tt:ReferenceToken"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="GetVideoSourceModesResponse">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="VideoSourceModes" type="tr2:VideoSourceMode"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="SetVideoSourceMode">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="VideoSourceToken" type="tt:ReferenceToken"/>
            <xs:element name="VideoSourceModeToken" type="tt:ReferenceToken"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="SetVideoSourceModeResponse">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Reboot" type="xs:boolean"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:simpleType name="EncodingTypes">
        <xs:list itemType="xs:string"/>
      </xs:simpleType>
      <xs:complexType name="VideoSourceMode">
        <xs:sequence>
          <xs:element name="MaxFramerate" type="xs:float"/>
          <xs:element name="MaxResolution" type="tt:VideoResolution"/>
          <xs:element name="Encodings" type="tr2:EncodingTypes"/>
          <xs:element name="Reboot" type="xs:boolean"/>
          <xs:element name="Description" type="tt:Description" minOccurs="0"/>

```



```

        <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="token" type="tt:ReferenceToken" use="required"/>
    <xs:attribute name="Enabled" type="xs:boolean"/>
    <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:element name="GetOSDs">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="OSDToken" type="tt:ReferenceToken" minOccurs="0"/>
            <xs:element name="ConfigurationToken" type="tt:ReferenceToken"
minOccurs="0"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="GetOSDsResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="OSDs" type="tt:OSDConfiguration" minOccurs="0"
maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="SetOSD">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="OSD" type="tt:OSDConfiguration"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="SetOSDResponse" type="tr2:SetConfigurationResponse"/>
<xs:element name="GetOSDOptions">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="ConfigurationToken" type="tt:ReferenceToken"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="GetOSDOptionsResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="OSDOptions" type="tt:OSDConfigurationOptions"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="CreateOSD">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="OSD" type="tt:OSDConfiguration"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="CreateOSDResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="OSDToken" type="tt:ReferenceToken"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="DeleteOSD">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="OSDToken" type="tt:ReferenceToken"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="DeleteOSDResponse" type="tr2:SetConfigurationResponse"/>
<xs:simpleType name="MaskType">
    <xs:restriction base="xs:string">
        <xs:enumeration value="Color"/>
        <xs:enumeration value="Pixelated"/>
        <xs:enumeration value="Blurred"/>
    </xs:restriction>
</xs:simpleType>
<xs:complexType name="Mask">
    <xs:sequence>
        <xs:element name="ConfigurationToken" type="tt:ReferenceToken"/>
    </xs:sequence>
</xs:complexType>

```



```

        <xs:element name="Polygon" type="tt:Polygon"/>
        <xs:element name="Type" type="xs:string"/>
        <xs:element name="Color" type="tt:Color" minOccurs="0"/>
        <xs:element name="Enabled" type="xs:boolean"/>
        <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="token" type="tt:ReferenceToken"/>
    <xs:attribute name="VideoSourceConfiguration" type="tt:ReferenceToken"/>
    <xs:anyAttribute/>
</xs:complexType>
<xs:element name="GetMasks">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Token" type="tt:ReferenceToken" minOccurs="0"/>
            <xs:element name="ConfigurationToken" type="tt:ReferenceToken"
minOccurs="0"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="GetMasksResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Masks" type="tr2:Mask" minOccurs="0"
maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="SetMask">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Mask" type="tr2:Mask"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="SetMaskResponse" type="tr2:SetConfigurationResponse"/>
<xs:element name="GetMaskOptions">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="ConfigurationToken" type="tt:ReferenceToken"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:complexType name="MaskOptions">
    <xs:sequence>
        <xs:element name="MaxMasks" type="xs:int"/>
        <xs:element name="MaxPoints" type="xs:int"/>
        <xs:element name="Types" type="xs:string" maxOccurs="unbounded"/>
        <xs:element name="Color" type="tt:ColorOptions"/>
        <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="RectangleOnly" type="xs:boolean"/>
    <xs:attribute name="SingleColorOnly" type="xs:boolean"/>
    <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:element name="GetMaskOptionsResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Options" type="tr2:MaskOptions"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="CreateMask">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Mask" type="tr2:Mask"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="CreateMaskResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Token" type="tt:ReferenceToken"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="DeleteMask">

```

```

        <xs:complexType>
            <xs:sequence>
                <xs:element name="Token" type="tt:ReferenceToken"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="DeleteMaskResponse" type="tr2:SetConfigurationResponse"/>
</xs:schema>
</wsdl:types>
<wsdl:message name="GetServiceCapabilitiesRequest">
    <wsdl:part name="parameters" element="tr2:GetServiceCapabilities"/>
</wsdl:message>
<wsdl:message name="GetServiceCapabilitiesResponse">
    <wsdl:part name="parameters" element="tr2:GetServiceCapabilitiesResponse"/>
</wsdl:message>
<wsdl:message name="CreateProfileRequest">
    <wsdl:part name="parameters" element="tr2:CreateProfile"/>
</wsdl:message>
<wsdl:message name="CreateProfileResponse">
    <wsdl:part name="parameters" element="tr2:CreateProfileResponse"/>
</wsdl:message>
<wsdl:message name="GetProfilesRequest">
    <wsdl:part name="parameters" element="tr2:GetProfiles"/>
</wsdl:message>
<wsdl:message name="GetProfilesResponse">
    <wsdl:part name="parameters" element="tr2:GetProfilesResponse"/>
</wsdl:message>
<wsdl:message name="AddConfigurationRequest">
    <wsdl:part name="parameters" element="tr2:AddConfiguration"/>
</wsdl:message>
<wsdl:message name="AddConfigurationResponse">
    <wsdl:part name="parameters" element="tr2:AddConfigurationResponse"/>
</wsdl:message>
<wsdl:message name="RemoveConfigurationRequest">
    <wsdl:part name="parameters" element="tr2:RemoveConfiguration"/>
</wsdl:message>
<wsdl:message name="RemoveConfigurationResponse">
    <wsdl:part name="parameters" element="tr2:RemoveConfigurationResponse"/>
</wsdl:message>
<wsdl:message name="DeleteProfileRequest">
    <wsdl:part name="parameters" element="tr2:DeleteProfile"/>
</wsdl:message>
<wsdl:message name="DeleteProfileResponse">
    <wsdl:part name="parameters" element="tr2:DeleteProfileResponse"/>
</wsdl:message>
<wsdl:message name="GetVideoSourceConfigurationsRequest">
    <wsdl:part name="parameters" element="tr2:GetVideoSourceConfigurations"/>
</wsdl:message>
<wsdl:message name="GetVideoSourceConfigurationsResponse">
    <wsdl:part name="parameters" element="tr2:GetVideoSourceConfigurationsResponse"/>
</wsdl:message>
<wsdl:message name="GetVideoEncoderConfigurationsRequest">
    <wsdl:part name="parameters" element="tr2:GetVideoEncoderConfigurations"/>
</wsdl:message>
<wsdl:message name="GetVideoEncoderConfigurationsResponse">
    <wsdl:part name="parameters"
element="tr2:GetVideoEncoderConfigurationsResponse"/>
</wsdl:message>
<wsdl:message name="GetAudioSourceConfigurationsRequest">
    <wsdl:part name="parameters" element="tr2:GetAudioSourceConfigurations"/>
</wsdl:message>
<wsdl:message name="GetAudioSourceConfigurationsResponse">
    <wsdl:part name="parameters" element="tr2:GetAudioSourceConfigurationsResponse"/>
</wsdl:message>
<wsdl:message name="GetAudioEncoderConfigurationsRequest">
    <wsdl:part name="parameters" element="tr2:GetAudioEncoderConfigurations"/>
</wsdl:message>
<wsdl:message name="GetAudioEncoderConfigurationsResponse">
    <wsdl:part name="parameters"
element="tr2:GetAudioEncoderConfigurationsResponse"/>
</wsdl:message>
<wsdl:message name="GetAnalyticsConfigurationsRequest">
    <wsdl:part name="parameters" element="tr2:GetAnalyticsConfigurations"/>
</wsdl:message>
<wsdl:message name="GetAnalyticsConfigurationsResponse">
    <wsdl:part name="parameters" element="tr2:GetAnalyticsConfigurationsResponse"/>
</wsdl:message>
<wsdl:message name="GetMetadataConfigurationsRequest">

```

```
<wsdl:part name="parameters" element="tr2:GetMetadataConfigurations"/>
</wsdl:message>
<wsdl:message name="GetMetadataConfigurationsResponse">
  <wsdl:part name="parameters" element="tr2:GetMetadataConfigurationsResponse"/>
</wsdl:message>
<wsdl:message name="GetAudioOutputConfigurationsRequest">
  <wsdl:part name="parameters" element="tr2:GetAudioOutputConfigurations"/>
</wsdl:message>
<wsdl:message name="GetAudioOutputConfigurationsResponse">
  <wsdl:part name="parameters" element="tr2:GetAudioOutputConfigurationsResponse"/>
</wsdl:message>
<wsdl:message name="GetAudioDecoderConfigurationsRequest">
  <wsdl:part name="parameters" element="tr2:GetAudioDecoderConfigurations"/>
</wsdl:message>
<wsdl:message name="GetAudioDecoderConfigurationsResponse">
  <wsdl:part name="parameters"
element="tr2:GetAudioDecoderConfigurationsResponse"/>
</wsdl:message>
<wsdl:message name="SetVideoSourceConfigurationRequest">
  <wsdl:part name="parameters" element="tr2:SetVideoSourceConfiguration"/>
</wsdl:message>
<wsdl:message name="SetVideoSourceConfigurationResponse">
  <wsdl:part name="parameters" element="tr2:SetVideoSourceConfigurationResponse"/>
</wsdl:message>
<wsdl:message name="SetVideoEncoderConfigurationRequest">
  <wsdl:part name="parameters" element="tr2:SetVideoEncoderConfiguration"/>
</wsdl:message>
<wsdl:message name="SetVideoEncoderConfigurationResponse">
  <wsdl:part name="parameters" element="tr2:SetVideoEncoderConfigurationResponse"/>
</wsdl:message>
<wsdl:message name="SetAudioSourceConfigurationRequest">
  <wsdl:part name="parameters" element="tr2:SetAudioSourceConfiguration"/>
</wsdl:message>
<wsdl:message name="SetAudioSourceConfigurationResponse">
  <wsdl:part name="parameters" element="tr2:SetAudioSourceConfigurationResponse"/>
</wsdl:message>
<wsdl:message name="SetAudioEncoderConfigurationRequest">
  <wsdl:part name="parameters" element="tr2:SetAudioEncoderConfiguration"/>
</wsdl:message>
<wsdl:message name="SetAudioEncoderConfigurationResponse">
  <wsdl:part name="parameters" element="tr2:SetAudioEncoderConfigurationResponse"/>
</wsdl:message>
<wsdl:message name="SetMetadataConfigurationRequest">
  <wsdl:part name="parameters" element="tr2:SetMetadataConfiguration"/>
</wsdl:message>
<wsdl:message name="SetMetadataConfigurationResponse">
  <wsdl:part name="parameters" element="tr2:SetMetadataConfigurationResponse"/>
</wsdl:message>
<wsdl:message name="SetAudioOutputConfigurationRequest">
  <wsdl:part name="parameters" element="tr2:SetAudioOutputConfiguration"/>
</wsdl:message>
<wsdl:message name="SetAudioOutputConfigurationResponse">
  <wsdl:part name="parameters" element="tr2:SetAudioOutputConfigurationResponse"/>
</wsdl:message>
<wsdl:message name="SetAudioDecoderConfigurationRequest">
  <wsdl:part name="parameters" element="tr2:SetAudioDecoderConfiguration"/>
</wsdl:message>
<wsdl:message name="SetAudioDecoderConfigurationResponse">
  <wsdl:part name="parameters" element="tr2:SetAudioDecoderConfigurationResponse"/>
</wsdl:message>
<wsdl:message name="GetVideoSourceConfigurationOptionsRequest">
  <wsdl:part name="parameters" element="tr2:GetVideoSourceConfigurationOptions"/>
</wsdl:message>
<wsdl:message name="GetVideoSourceConfigurationOptionsResponse">
  <wsdl:part name="parameters"
element="tr2:GetVideoSourceConfigurationOptionsResponse"/>
</wsdl:message>
<wsdl:message name="GetVideoEncoderConfigurationOptionsRequest">
  <wsdl:part name="parameters" element="tr2:GetVideoEncoderConfigurationOptions"/>
</wsdl:message>
<wsdl:message name="GetVideoEncoderConfigurationOptionsResponse">
  <wsdl:part name="parameters"
element="tr2:GetVideoEncoderConfigurationOptionsResponse"/>
</wsdl:message>
<wsdl:message name="GetAudioSourceConfigurationOptionsRequest">
  <wsdl:part name="parameters" element="tr2:GetAudioSourceConfigurationOptions"/>
</wsdl:message>
<wsdl:message name="GetAudioSourceConfigurationOptionsResponse">
```

```

        <wsdl:part name="parameters"
element="tr2:GetAudioSourceConfigurationOptionsResponse"/>
    </wsdl:message>
    <wsdl:message name="GetAudioEncoderConfigurationOptionsRequest">
        <wsdl:part name="parameters" element="tr2:GetAudioEncoderConfigurationOptions"/>
    </wsdl:message>
    <wsdl:message name="GetAudioEncoderConfigurationOptionsResponse">
        <wsdl:part name="parameters"
element="tr2:GetAudioEncoderConfigurationOptionsResponse"/>
    </wsdl:message>
    <wsdl:message name="GetMetadataConfigurationOptionsRequest">
        <wsdl:part name="parameters" element="tr2:GetMetadataConfigurationOptions"/>
    </wsdl:message>
    <wsdl:message name="GetMetadataConfigurationOptionsResponse">
        <wsdl:part name="parameters"
element="tr2:GetMetadataConfigurationOptionsResponse"/>
    </wsdl:message>
    <wsdl:message name="GetAudioOutputConfigurationOptionsRequest">
        <wsdl:part name="parameters" element="tr2:GetAudioOutputConfigurationOptions"/>
    </wsdl:message>
    <wsdl:message name="GetAudioOutputConfigurationOptionsResponse">
        <wsdl:part name="parameters"
element="tr2:GetAudioOutputConfigurationOptionsResponse"/>
    </wsdl:message>
    <wsdl:message name="GetAudioDecoderConfigurationOptionsRequest">
        <wsdl:part name="parameters" element="tr2:GetAudioDecoderConfigurationOptions"/>
    </wsdl:message>
    <wsdl:message name="GetAudioDecoderConfigurationOptionsResponse">
        <wsdl:part name="parameters"
element="tr2:GetAudioDecoderConfigurationOptionsResponse"/>
    </wsdl:message>
    <wsdl:message name="GetVideoEncoderInstancesRequest">
        <wsdl:part name="parameters" element="tr2:GetVideoEncoderInstances"/>
    </wsdl:message>
    <wsdl:message name="GetVideoEncoderInstancesResponse">
        <wsdl:part name="parameters" element="tr2:GetVideoEncoderInstancesResponse"/>
    </wsdl:message>
    <wsdl:message name="GetStreamUriRequest">
        <wsdl:part name="parameters" element="tr2:GetStreamUri"/>
    </wsdl:message>
    <wsdl:message name="GetStreamUriResponse">
        <wsdl:part name="parameters" element="tr2:GetStreamUriResponse"/>
    </wsdl:message>
    <wsdl:message name="SetSynchronizationPointRequest">
        <wsdl:part name="parameters" element="tr2:SetSynchronizationPoint"/>
    </wsdl:message>
    <wsdl:message name="SetSynchronizationPointResponse">
        <wsdl:part name="parameters" element="tr2:SetSynchronizationPointResponse"/>
    </wsdl:message>
    <wsdl:message name="GetSnapshotUriRequest">
        <wsdl:part name="parameters" element="tr2:GetSnapshotUri"/>
    </wsdl:message>
    <wsdl:message name="GetSnapshotUriResponse">
        <wsdl:part name="parameters" element="tr2:GetSnapshotUriResponse"/>
    </wsdl:message>
    <wsdl:message name="StartMulticastStreamingRequest">
        <wsdl:part name="parameters" element="tr2:StartMulticastStreaming"/>
    </wsdl:message>
    <wsdl:message name="StopMulticastStreamingRequest">
        <wsdl:part name="parameters" element="tr2:StopMulticastStreaming"/>
    </wsdl:message>
    <wsdl:message name="StopMulticastStreamingResponse">
        <wsdl:part name="parameters" element="tr2:StopMulticastStreamingResponse"/>
    </wsdl:message>
    <wsdl:message name="StartMulticastStreamingResponse">
        <wsdl:part name="parameters" element="tr2:StartMulticastStreamingResponse"/>
    </wsdl:message>
    <wsdl:message name="GetVideoSourceModesRequest">
        <wsdl:part name="parameters" element="tr2:GetVideoSourceModes"/>
    </wsdl:message>
    <wsdl:message name="GetVideoSourceModesResponse">
        <wsdl:part name="parameters" element="tr2:GetVideoSourceModesResponse"/>
    </wsdl:message>
    <wsdl:message name="SetVideoSourceModeRequest">
        <wsdl:part name="parameters" element="tr2:SetVideoSourceMode"/>
    </wsdl:message>
    <wsdl:message name="SetVideoSourceModeResponse">
        <wsdl:part name="parameters" element="tr2:SetVideoSourceModeResponse"/>
    </wsdl:message>

```

```
</wsdl:message>
<wsdl:message name="GetOSDsRequest">
  <wsdl:part name="parameters" element="tr2:GetOSDs"/>
</wsdl:message>
<wsdl:message name="GetOSDsResponse">
  <wsdl:part name="parameters" element="tr2:GetOSDsResponse"/>
</wsdl:message>
<wsdl:message name="GetOSDOptionsRequest">
  <wsdl:part name="parameters" element="tr2:GetOSDOptions"/>
</wsdl:message>
<wsdl:message name="GetOSDOptionsResponse">
  <wsdl:part name="parameters" element="tr2:GetOSDOptionsResponse"/>
</wsdl:message>
<wsdl:message name="SetOSDRequest">
  <wsdl:part name="parameters" element="tr2:SetOSD"/>
</wsdl:message>
<wsdl:message name="SetOSDResponse">
  <wsdl:part name="parameters" element="tr2:SetOSDResponse"/>
</wsdl:message>
<wsdl:message name="CreateOSDRequest">
  <wsdl:part name="parameters" element="tr2:CreateOSD"/>
</wsdl:message>
<wsdl:message name="CreateOSDResponse">
  <wsdl:part name="parameters" element="tr2:CreateOSDResponse"/>
</wsdl:message>
<wsdl:message name="DeleteOSDRequest">
  <wsdl:part name="parameters" element="tr2:DeleteOSD"/>
</wsdl:message>
<wsdl:message name="DeleteOSDResponse">
  <wsdl:part name="parameters" element="tr2:DeleteOSDResponse"/>
</wsdl:message>
<wsdl:message name="GetMasksRequest">
  <wsdl:part name="parameters" element="tr2:GetMasks"/>
</wsdl:message>
<wsdl:message name="GetMasksResponse">
  <wsdl:part name="parameters" element="tr2:GetMasksResponse"/>
</wsdl:message>
<wsdl:message name="GetMaskOptionsRequest">
  <wsdl:part name="parameters" element="tr2:GetMaskOptions"/>
</wsdl:message>
<wsdl:message name="GetMaskOptionsResponse">
  <wsdl:part name="parameters" element="tr2:GetMaskOptionsResponse"/>
</wsdl:message>
<wsdl:message name="SetMaskRequest">
  <wsdl:part name="parameters" element="tr2:SetMask"/>
</wsdl:message>
<wsdl:message name="SetMaskResponse">
  <wsdl:part name="parameters" element="tr2:SetMaskResponse"/>
</wsdl:message>
<wsdl:message name="CreateMaskRequest">
  <wsdl:part name="parameters" element="tr2:CreateMask"/>
</wsdl:message>
<wsdl:message name="CreateMaskResponse">
  <wsdl:part name="parameters" element="tr2:CreateMaskResponse"/>
</wsdl:message>
<wsdl:message name="DeleteMaskRequest">
  <wsdl:part name="parameters" element="tr2:DeleteMask"/>
</wsdl:message>
<wsdl:message name="DeleteMaskResponse">
  <wsdl:part name="parameters" element="tr2:DeleteMaskResponse"/>
</wsdl:message>
<wsdl:portType name="Media2">
  <wsdl:operation name="GetServiceCapabilities">
    <wsdl:input message="tr2:GetServiceCapabilitiesRequest"/>
    <wsdl:output message="tr2:GetServiceCapabilitiesResponse"/>
  </wsdl:operation>
  <wsdl:operation name="CreateProfile">
    <wsdl:input message="tr2:CreateProfileRequest"/>
    <wsdl:output message="tr2:CreateProfileResponse"/>
  </wsdl:operation>
  <wsdl:operation name="GetProfiles">
    <wsdl:input message="tr2:GetProfilesRequest"/>
    <wsdl:output message="tr2:GetProfilesResponse"/>
  </wsdl:operation>
  <wsdl:operation name="AddConfiguration">
    <wsdl:input message="tr2:AddConfigurationRequest"/>
    <wsdl:output message="tr2:AddConfigurationResponse"/>
  </wsdl:operation>
</wsdl:portType>
```



```

<wsdl:operation name="RemoveConfiguration">
  <wsdl:input message="tr2:RemoveConfigurationRequest"/>
  <wsdl:output message="tr2:RemoveConfigurationResponse"/>
</wsdl:operation>
<wsdl:operation name="DeleteProfile">
  <wsdl:input message="tr2:DeleteProfileRequest"/>
  <wsdl:output message="tr2:DeleteProfileResponse"/>
</wsdl:operation>
<wsdl:operation name="GetVideoSourceConfigurations">
  <wsdl:input message="tr2:GetVideoSourceConfigurationsRequest"/>
  <wsdl:output message="tr2:GetVideoSourceConfigurationsResponse"/>
</wsdl:operation>
<wsdl:operation name="GetVideoEncoderConfigurations">
  <wsdl:input message="tr2:GetVideoEncoderConfigurationsRequest"/>
  <wsdl:output message="tr2:GetVideoEncoderConfigurationsResponse"/>
</wsdl:operation>
<wsdl:operation name="GetAudioSourceConfigurations">
  <wsdl:input message="tr2:GetAudioSourceConfigurationsRequest"/>
  <wsdl:output message="tr2:GetAudioSourceConfigurationsResponse"/>
</wsdl:operation>
<wsdl:operation name="GetAudioEncoderConfigurations">
  <wsdl:input message="tr2:GetAudioEncoderConfigurationsRequest"/>
  <wsdl:output message="tr2:GetAudioEncoderConfigurationsResponse"/>
</wsdl:operation>
<wsdl:operation name="GetAnalyticsConfigurations">
  <wsdl:input message="tr2:GetAnalyticsConfigurationsRequest"/>
  <wsdl:output message="tr2:GetAnalyticsConfigurationsResponse"/>
</wsdl:operation>
<wsdl:operation name="GetMetadataConfigurations">
  <wsdl:input message="tr2:GetMetadataConfigurationsRequest"/>
  <wsdl:output message="tr2:GetMetadataConfigurationsResponse"/>
</wsdl:operation>
<wsdl:operation name="GetAudioOutputConfigurations">
  <wsdl:input message="tr2:GetAudioOutputConfigurationsRequest"/>
  <wsdl:output message="tr2:GetAudioOutputConfigurationsResponse"/>
</wsdl:operation>
<wsdl:operation name="GetAudioDecoderConfigurations">
  <wsdl:input message="tr2:GetAudioDecoderConfigurationsRequest"/>
  <wsdl:output message="tr2:GetAudioDecoderConfigurationsResponse"/>
</wsdl:operation>
<wsdl:operation name="SetVideoSourceConfiguration">
  <wsdl:input message="tr2:SetVideoSourceConfigurationRequest"/>
  <wsdl:output message="tr2:SetVideoSourceConfigurationResponse"/>
</wsdl:operation>
<wsdl:operation name="SetVideoEncoderConfiguration">
  <wsdl:input message="tr2:SetVideoEncoderConfigurationRequest"/>
  <wsdl:output message="tr2:SetVideoEncoderConfigurationResponse"/>
</wsdl:operation>
<wsdl:operation name="SetAudioSourceConfiguration">
  <wsdl:input message="tr2:SetAudioSourceConfigurationRequest"/>
  <wsdl:output message="tr2:SetAudioSourceConfigurationResponse"/>
</wsdl:operation>
<wsdl:operation name="SetAudioEncoderConfiguration">
  <wsdl:input message="tr2:SetAudioEncoderConfigurationRequest"/>
  <wsdl:output message="tr2:SetAudioEncoderConfigurationResponse"/>
</wsdl:operation>
<wsdl:operation name="SetMetadataConfiguration">
  <wsdl:input message="tr2:SetMetadataConfigurationRequest"/>
  <wsdl:output message="tr2:SetMetadataConfigurationResponse"/>
</wsdl:operation>
<wsdl:operation name="SetAudioOutputConfiguration">
  <wsdl:input message="tr2:SetAudioOutputConfigurationRequest"/>
  <wsdl:output message="tr2:SetAudioOutputConfigurationResponse"/>
</wsdl:operation>
<wsdl:operation name="SetAudioDecoderConfiguration">
  <wsdl:input message="tr2:SetAudioDecoderConfigurationRequest"/>
  <wsdl:output message="tr2:SetAudioDecoderConfigurationResponse"/>
</wsdl:operation>
<wsdl:operation name="GetVideoSourceConfigurationOptions">
  <wsdl:input message="tr2:GetVideoSourceConfigurationOptionsRequest"/>
  <wsdl:output message="tr2:GetVideoSourceConfigurationOptionsResponse"/>
</wsdl:operation>
<wsdl:operation name="GetVideoEncoderConfigurationOptions">
  <wsdl:input message="tr2:GetVideoEncoderConfigurationOptionsRequest"/>
  <wsdl:output message="tr2:GetVideoEncoderConfigurationOptionsResponse"/>
</wsdl:operation>
<wsdl:operation name="GetAudioSourceConfigurationOptions">
  <wsdl:input message="tr2:GetAudioSourceConfigurationOptionsRequest"/>

```

```

    <wsdl:output message="tr2:GetAudioSourceConfigurationOptionsResponse"/>
</wsdl:operation>
<wsdl:operation name="GetAudioEncoderConfigurationOptions">
    <wsdl:input message="tr2:GetAudioEncoderConfigurationOptionsRequest"/>
    <wsdl:output message="tr2:GetAudioEncoderConfigurationOptionsResponse"/>
</wsdl:operation>
<wsdl:operation name="GetMetadataConfigurationOptions">
    <wsdl:input message="tr2:GetMetadataConfigurationOptionsRequest"/>
    <wsdl:output message="tr2:GetMetadataConfigurationOptionsResponse"/>
</wsdl:operation>
<wsdl:operation name="GetAudioOutputConfigurationOptions">
    <wsdl:input message="tr2:GetAudioOutputConfigurationOptionsRequest"/>
    <wsdl:output message="tr2:GetAudioOutputConfigurationOptionsResponse"/>
</wsdl:operation>
<wsdl:operation name="GetAudioDecoderConfigurationOptions">
    <wsdl:input message="tr2:GetAudioDecoderConfigurationOptionsRequest"/>
    <wsdl:output message="tr2:GetAudioDecoderConfigurationOptionsResponse"/>
</wsdl:operation>
<wsdl:operation name="GetVideoEncoderInstances">
    <wsdl:input message="tr2:GetVideoEncoderInstancesRequest"/>
    <wsdl:output message="tr2:GetVideoEncoderInstancesResponse"/>
</wsdl:operation>
<wsdl:operation name="GetStreamUri">
    <wsdl:input message="tr2:GetStreamUriRequest"/>
    <wsdl:output message="tr2:GetStreamUriResponse"/>
</wsdl:operation>
<wsdl:operation name="StartMulticastStreaming">
    <wsdl:input message="tr2:StartMulticastStreamingRequest"/>
    <wsdl:output message="tr2:StartMulticastStreamingResponse"/>
</wsdl:operation>
<wsdl:operation name="StopMulticastStreaming">
    <wsdl:input message="tr2:StopMulticastStreamingRequest"/>
    <wsdl:output message="tr2:StopMulticastStreamingResponse"/>
</wsdl:operation>
<wsdl:operation name="SetSynchronizationPoint">
    <wsdl:input message="tr2:SetSynchronizationPointRequest"/>
    <wsdl:output message="tr2:SetSynchronizationPointResponse"/>
</wsdl:operation>
<wsdl:operation name="GetSnapshotUri">
    <wsdl:input message="tr2:GetSnapshotUriRequest"/>
    <wsdl:output message="tr2:GetSnapshotUriResponse"/>
</wsdl:operation>
<wsdl:operation name="GetVideoSourceModes">
    <wsdl:input message="tr2:GetVideoSourceModesRequest"/>
    <wsdl:output message="tr2:GetVideoSourceModesResponse"/>
</wsdl:operation>
<wsdl:operation name="SetVideoSourceMode">
    <wsdl:input message="tr2:SetVideoSourceModeRequest"/>
    <wsdl:output message="tr2:SetVideoSourceModeResponse"/>
</wsdl:operation>
<wsdl:operation name="GetOSDs">
    <wsdl:input message="tr2:GetOSDsRequest"/>
    <wsdl:output message="tr2:GetOSDsResponse"/>
</wsdl:operation>
<wsdl:operation name="GetOSDOptions">
    <wsdl:input message="tr2:GetOSDOptionsRequest"/>
    <wsdl:output message="tr2:GetOSDOptionsResponse"/>
</wsdl:operation>
<wsdl:operation name="SetOSD">
    <wsdl:input message="tr2:SetOSDRequest"/>
    <wsdl:output message="tr2:SetOSDResponse"/>
</wsdl:operation>
<wsdl:operation name="CreateOSD">
    <wsdl:input message="tr2:CreateOSDRequest"/>
    <wsdl:output message="tr2:CreateOSDResponse"/>
</wsdl:operation>
<wsdl:operation name="DeleteOSD">
    <wsdl:input message="tr2>DeleteOSDRequest"/>
    <wsdl:output message="tr2>DeleteOSDResponse"/>
</wsdl:operation>
<wsdl:operation name="GetMasks">
    <wsdl:input message="tr2:GetMasksRequest"/>
    <wsdl:output message="tr2:GetMasksResponse"/>
</wsdl:operation>
<wsdl:operation name="GetMaskOptions">
    <wsdl:input message="tr2:GetMaskOptionsRequest"/>
    <wsdl:output message="tr2:GetMaskOptionsResponse"/>
</wsdl:operation>

```

```

<wsdl:operation name="SetMask">
  <wsdl:input message="tr2:SetMaskRequest"/>
  <wsdl:output message="tr2:SetMaskResponse"/>
</wsdl:operation>
<wsdl:operation name="CreateMask">
  <wsdl:input message="tr2:CreateMaskRequest"/>
  <wsdl:output message="tr2:CreateMaskResponse"/>
</wsdl:operation>
<wsdl:operation name="DeleteMask">
  <wsdl:input message="tr2:DeleteMaskRequest"/>
  <wsdl:output message="tr2:DeleteMaskResponse"/>
</wsdl:operation>
</wsdl:portType>
<wsdl:binding name="Media2Binding" type="tr2:Media2">
  <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
  <wsdl:operation name="GetServiceCapabilities">
    <soap:operation
soapAction="http://www.onvif.org/ver20/media/wsdl/GetServiceCapabilities"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="CreateProfile">
    <soap:operation
soapAction="http://www.onvif.org/ver20/media/wsdl/CreateProfile"/>
    <wsdl:input>
      <soap:body parts="parameters" use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body parts="parameters" use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="GetProfiles">
    <soap:operation
soapAction="http://www.onvif.org/ver20/media/wsdl/GetProfiles"/>
    <wsdl:input>
      <soap:body parts="parameters" use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body parts="parameters" use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="AddConfiguration">
    <soap:operation
soapAction="http://www.onvif.org/ver20/media/wsdl/AddConfiguration"/>
    <wsdl:input>
      <soap:body parts="parameters" use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body parts="parameters" use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="RemoveConfiguration">
    <soap:operation
soapAction="http://www.onvif.org/ver20/media/wsdl/RemoveConfiguration"/>
    <wsdl:input>
      <soap:body parts="parameters" use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body parts="parameters" use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="DeleteProfile">
    <soap:operation
soapAction="http://www.onvif.org/ver20/media/wsdl/DeleteProfile"/>
    <wsdl:input>
      <soap:body parts="parameters" use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body parts="parameters" use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="GetVideoSourceConfigurations">
    <soap:operation
soapAction="http://www.onvif.org/ver20/media/wsdl/GetVideoSourceConfigurations"/>

```



```
<wsdl:input>
  <soap:body parts="parameters" use="literal"/>
</wsdl:input>
<wsdl:output>
  <soap:body parts="parameters" use="literal"/>
</wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetVideoEncoderConfigurations">
  <soap:operation
soapAction="http://www.onvif.org/ver20/media/wsdl/GetVideoEncoderConfigurations"/>
  <wsdl:input>
    <soap:body parts="parameters" use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body parts="parameters" use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetAudioSourceConfigurations">
  <soap:operation
soapAction="http://www.onvif.org/ver20/media/wsdl/GetAudioSourceConfigurations"/>
  <wsdl:input>
    <soap:body parts="parameters" use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body parts="parameters" use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetAudioEncoderConfigurations">
  <soap:operation
soapAction="http://www.onvif.org/ver20/media/wsdl/GetAudioEncoderConfigurations"/>
  <wsdl:input>
    <soap:body parts="parameters" use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body parts="parameters" use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetAnalyticsConfigurations">
  <soap:operation
soapAction="http://www.onvif.org/ver20/media/wsdl/GetAnalyticsConfigurations"/>
  <wsdl:input>
    <soap:body parts="parameters" use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body parts="parameters" use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetMetadataConfigurations">
  <soap:operation
soapAction="http://www.onvif.org/ver20/media/wsdl/GetMetadataConfigurations"/>
  <wsdl:input>
    <soap:body parts="parameters" use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body parts="parameters" use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetAudioOutputConfigurations">
  <soap:operation
soapAction="http://www.onvif.org/ver20/media/wsdl/GetAudioOutputConfigurations"/>
  <wsdl:input>
    <soap:body parts="parameters" use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body parts="parameters" use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetAudioDecoderConfigurations">
  <soap:operation
soapAction="http://www.onvif.org/ver20/media/wsdl/GetAudioDecoderConfigurations"/>
  <wsdl:input>
    <soap:body parts="parameters" use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body parts="parameters" use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="SetVideoSourceConfiguration">
```

```

        <soap:operation
soapAction="http://www.onvif.org/ver20/media/wsd/VideoSourceConfiguration"/>
        <wsdl:input>
            <soap:body parts="parameters" use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body parts="parameters" use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="SetVideoEncoderConfiguration">
        <soap:operation
soapAction="http://www.onvif.org/ver20/media/wsd/VideoEncoderConfiguration"/>
        <wsdl:input>
            <soap:body parts="parameters" use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body parts="parameters" use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="SetAudioSourceConfiguration">
        <soap:operation
soapAction="http://www.onvif.org/ver20/media/wsd/AudioSourceConfiguration"/>
        <wsdl:input>
            <soap:body parts="parameters" use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body parts="parameters" use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="SetAudioEncoderConfiguration">
        <soap:operation
soapAction="http://www.onvif.org/ver20/media/wsd/AudioEncoderConfiguration"/>
        <wsdl:input>
            <soap:body parts="parameters" use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body parts="parameters" use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="SetMetadataConfiguration">
        <soap:operation
soapAction="http://www.onvif.org/ver20/media/wsd/MetadataConfiguration"/>
        <wsdl:input>
            <soap:body parts="parameters" use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body parts="parameters" use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="SetAudioOutputConfiguration">
        <soap:operation
soapAction="http://www.onvif.org/ver20/media/wsd/AudioOutputConfiguration"/>
        <wsdl:input>
            <soap:body parts="parameters" use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body parts="parameters" use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="SetAudioDecoderConfiguration">
        <soap:operation
soapAction="http://www.onvif.org/ver20/media/wsd/AudioDecoderConfiguration"/>
        <wsdl:input>
            <soap:body parts="parameters" use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body parts="parameters" use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetVideoSourceConfigurationOptions">
        <soap:operation
soapAction="http://www.onvif.org/ver20/media/wsd/GetVideoSourceConfigurationOptions/"
>
        <wsdl:input>
            <soap:body parts="parameters" use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body parts="parameters" use="literal"/>
        </wsdl:output>
    </wsdl:operation>

```

```

        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetVideoEncoderConfigurationOptions">
        <soap:operation
soapAction="http://www.onvif.org/ver20/media/wsdl/GetVideoEncoderConfigurationOptions"/
>
        <wsdl:input>
            <soap:body parts="parameters" use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body parts="parameters" use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetAudioSourceConfigurationOptions">
        <soap:operation
soapAction="http://www.onvif.org/ver20/media/wsdl/GetAudioSourceConfigurationOptions"/>
        <wsdl:input>
            <soap:body parts="parameters" use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body parts="parameters" use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetAudioEncoderConfigurationOptions">
        <soap:operation
soapAction="http://www.onvif.org/ver20/media/wsdl/GetAudioEncoderConfigurationOptions"/
>
        <wsdl:input>
            <soap:body parts="parameters" use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body parts="parameters" use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetMetadataConfigurationOptions">
        <soap:operation
soapAction="http://www.onvif.org/ver20/media/wsdl/GetMetadataConfigurationOptions"/>
        <wsdl:input>
            <soap:body parts="parameters" use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body parts="parameters" use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetAudioOutputConfigurationOptions">
        <soap:operation
soapAction="http://www.onvif.org/ver20/media/wsdl/GetAudioOutputConfigurationOptions"/>
        <wsdl:input>
            <soap:body parts="parameters" use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body parts="parameters" use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetAudioDecoderConfigurationOptions">
        <soap:operation
soapAction="http://www.onvif.org/ver20/media/wsdl/GetAudioDecoderConfigurationOptions"/
>
        <wsdl:input>
            <soap:body parts="parameters" use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body parts="parameters" use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetVideoEncoderInstances">
        <soap:operation
soapAction="http://www.onvif.org/ver20/media/wsdl/GetVideoEncoderInstances"/>
        <wsdl:input>
            <soap:body parts="parameters" use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body parts="parameters" use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetStreamUri">
        <soap:operation
soapAction="http://www.onvif.org/ver20/media/wsdl/GetStreamUri"/>

```

```

        <wsdl:input>
            <soap:body parts="parameters" use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body parts="parameters" use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="StartMulticastStreaming">
        <soap:operation
soapAction="http://www.onvif.org/ver20/media/wsdl/StartMulticastStreaming"/>
        <wsdl:input>
            <soap:body parts="parameters" use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body parts="parameters" use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="StopMulticastStreaming">
        <soap:operation
soapAction="http://www.onvif.org/ver20/media/wsdl/StopMulticastStreaming"/>
        <wsdl:input>
            <soap:body parts="parameters" use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body parts="parameters" use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="SetSynchronizationPoint">
        <soap:operation
soapAction="http://www.onvif.org/ver20/media/wsdl/SetSynchronizationPoint"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetSnapshotUri">
        <soap:operation
soapAction="http://www.onvif.org/ver20/media/wsdl/GetSnapshotUri"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetVideoSourceModes">
        <soap:operation
soapAction="http://www.onvif.org/ver20/media/wsdl/GetVideoSourceModes"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="SetVideoSourceMode">
        <soap:operation
soapAction="http://www.onvif.org/ver20/media/wsdl/SetVideoSourceMode"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetOSDs">
        <soap:operation soapAction="http://www.onvif.org/ver20/media/wsdl/GetOSDs"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetOSDOptions">
        <soap:operation

```

```
soapAction="http://www.onvif.org/ver20/media/wsdl/GetOSDOptions"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="SetOSD">
  <soap:operation soapAction="http://www.onvif.org/ver20/media/wsdl/SetOSD"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="CreateOSD">
  <soap:operation soapAction="http://www.onvif.org/ver20/media/wsdl/CreateOSD"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="DeleteOSD">
  <soap:operation soapAction="http://www.onvif.org/ver20/media/wsdl/DeleteOSD"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetMasks">
  <soap:operation soapAction="http://www.onvif.org/ver20/media/wsdl/GetMasks"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetMaskOptions">
  <soap:operation
soapAction="http://www.onvif.org/ver20/media/wsdl/GetMaskOptions"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="SetMask">
  <soap:operation soapAction="http://www.onvif.org/ver20/media/wsdl/SetMask"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="CreateMask">
  <soap:operation
soapAction="http://www.onvif.org/ver20/media/wsdl/CreateMask"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="DeleteMask">
  <soap:operation
soapAction="http://www.onvif.org/ver20/media/wsdl/DeleteMask"/>
  <wsdl:input>
    <soap:body use="literal"/>
```

```
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
</wsdl:binding>
</wsdl:definitions>
```

IECNORM.COM : Click to view the full PDF of IEC 62676-2-31:2019

F.5 PTZ

```
<?xml version="1.0" encoding="utf-8"?><?xml-stylesheet type="text/xsl"
href="../../../ver20/util/onvif-wsdl-viewer.xsl"?><wsdl:definitions
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:tptz="http://www.onvif.org/ver20/ptz/wsdl"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap12/" name="PTZService"
targetNamespace="http://www.onvif.org/ver20/ptz/wsdl">
  <wsdl:types>
    <xs:schema xmlns:tt="http://www.onvif.org/ver10/schema"
targetNamespace="http://www.onvif.org/ver20/ptz/wsdl" elementFormDefault="qualified"
version="17.06">
      <xs:import namespace="http://www.onvif.org/ver10/schema"
schemaLocation="../../../ver10/schema/onvif.xsd"/>
      <xs:element name="GetServiceCapabilities">
        <xs:complexType>
          <xs:sequence/>
        </xs:complexType>
      </xs:element>
      <xs:element name="GetServiceCapabilitiesResponse">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Capabilities" type="tptz:Capabilities"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="Capabilities">
        <xs:sequence>
          <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="EFlip" type="xs:boolean"/>
        <xs:attribute name="Reverse" type="xs:boolean"/>
        <xs:attribute name="GetCompatibleConfigurations" type="xs:boolean"/>
        <xs:attribute name="MoveStatus" type="xs:boolean"/>
        <xs:attribute name="StatusPosition" type="xs:boolean"/>
        <xs:anyAttribute processContents="lax"/>
      </xs:complexType>
      <xs:element name="Capabilities" type="tptz:Capabilities"/>
      <xs:element name="GetNodes">
        <xs:complexType/>
      </xs:element>
      <xs:element name="GetNodesResponse">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="PTZNode" type="tt:PTZNode" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="GetNode">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="NodeToken" type="tt:ReferenceToken"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="GetNodeResponse">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="PTZNode" type="tt:PTZNode"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="GetConfigurations">
        <xs:complexType/>
      </xs:element>
      <xs:element name="GetConfigurationsResponse">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="PTZConfiguration" type="tt:PTZConfiguration"
minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="GetConfiguration">
```

```

<xs:complexType>
  <xs:sequence>
    <xs:element name="PTZConfigurationToken" type="tt:ReferenceToken"/>
  </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="GetConfigurationResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="PTZConfiguration" type="tt:PTZConfiguration"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="SetConfiguration">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="PTZConfiguration" type="tt:PTZConfiguration"/>
      <xs:element name="ForcePersistence" type="xs:boolean"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="SetConfigurationResponse">
  <xs:complexType>
    <xs:sequence minOccurs="0"/>
  </xs:complexType>
</xs:element>
<xs:element name="GetConfigurationOptions">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="ConfigurationToken" type="tt:ReferenceToken"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="GetConfigurationOptionsResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="PTZConfigurationOptions"
type="tt:PTZConfigurationOptions"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="SendAuxiliaryCommand">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="ProfileToken" type="tt:ReferenceToken"/>
      <xs:element name="AuxiliaryData" type="tt:AuxiliaryData"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="SendAuxiliaryCommandResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="AuxiliaryResponse" type="tt:AuxiliaryData"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="GetPresets">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="ProfileToken" type="tt:ReferenceToken"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="GetPresetsResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Preset" type="tt:PTZPreset" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="SetPreset">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="ProfileToken" type="tt:ReferenceToken"/>
      <xs:element name="PresetName" type="xs:string" minOccurs="0"/>
      <xs:element name="PresetToken" type="tt:ReferenceToken"
minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

```



```

        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="SetPresetResponse">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="PresetToken" type="tt:ReferenceToken"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="RemovePreset">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="ProfileToken" type="tt:ReferenceToken"/>
          <xs:element name="PresetToken" type="tt:ReferenceToken"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="RemovePresetResponse">
      <xs:complexType/>
    </xs:element>
    <xs:element name="GotoPreset">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="ProfileToken" type="tt:ReferenceToken"/>
          <xs:element name="PresetToken" type="tt:ReferenceToken"/>
          <xs:element name="Speed" type="tt:PTZSpeed" minOccurs="0"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="GotoPresetResponse">
      <xs:complexType/>
    </xs:element>
    <xs:element name="GetStatus">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="ProfileToken" type="tt:ReferenceToken"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="GetStatusResponse">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="PTZStatus" type="tt:PTZStatus"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="GotoHomePosition">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="ProfileToken" type="tt:ReferenceToken"/>
          <xs:element name="Speed" type="tt:PTZSpeed" minOccurs="0"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="GotoHomePositionResponse">
      <xs:complexType>
        <xs:sequence/>
      </xs:complexType>
    </xs:element>
    <xs:element name="SetHomePosition">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="ProfileToken" type="tt:ReferenceToken"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="SetHomePositionResponse">
      <xs:complexType>
        <xs:sequence/>
      </xs:complexType>
    </xs:element>
    <xs:element name="ContinuousMove">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="ProfileToken" type="tt:ReferenceToken"/>
          <xs:element name="Velocity" type="tt:PTZSpeed"/>
          <xs:element name="Timeout" type="xs:duration" minOccurs="0"/>

```

```

        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="ContinuousMoveResponse">
      <xs:complexType>
        <xs:sequence/>
      </xs:complexType>
    </xs:element>
    <xs:element name="RelativeMove">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="ProfileToken" type="tt:ReferenceToken"/>
          <xs:element name="Translation" type="tt:PTZVector"/>
          <xs:element name="Speed" type="tt:PTZSpeed" minOccurs="0"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="RelativeMoveResponse">
      <xs:complexType>
        <xs:sequence/>
      </xs:complexType>
    </xs:element>
    <xs:element name="AbsoluteMove">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="ProfileToken" type="tt:ReferenceToken"/>
          <xs:element name="Position" type="tt:PTZVector"/>
          <xs:element name="Speed" type="tt:PTZSpeed" minOccurs="0"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="AbsoluteMoveResponse">
      <xs:complexType>
        <xs:sequence/>
      </xs:complexType>
    </xs:element>
    <xs:element name="Stop">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="ProfileToken" type="tt:ReferenceToken"/>
          <xs:element name="PanTilt" type="xs:boolean" minOccurs="0"/>
          <xs:element name="Zoom" type="xs:boolean" minOccurs="0"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="StopResponse">
      <xs:complexType>
        <xs:sequence/>
      </xs:complexType>
    </xs:element>
    <xs:element name="GetPresetTours">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="ProfileToken" type="tt:ReferenceToken"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="GetPresetToursResponse">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="PresetTour" type="tt:PresetTour" minOccurs="0"
maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="GetPresetTour">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="ProfileToken" type="tt:ReferenceToken"/>
          <xs:element name="PresetTourToken" type="tt:ReferenceToken"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="GetPresetTourResponse">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="PresetTour" type="tt:PresetTour"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>

```

```

        </xs:complexType>
      </xs:element>
      <xs:element name="GetPresetTourOptions">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="ProfileToken" type="tt:ReferenceToken"/>
            <xs:element name="PresetTourToken" type="tt:ReferenceToken"
minOccurs="0"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="GetPresetTourOptionsResponse">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Options" type="tt:PTZPresetTourOptions"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="CreatePresetTour">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="ProfileToken" type="tt:ReferenceToken"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="CreatePresetTourResponse">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="PresetTourToken" type="tt:ReferenceToken"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="ModifyPresetTour">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="ProfileToken" type="tt:ReferenceToken"/>
            <xs:element name="PresetTour" type="tt:PresetTour"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="ModifyPresetTourResponse">
        <xs:complexType>
          <xs:sequence/>
        </xs:complexType>
      </xs:element>
      <xs:element name="OperatePresetTour">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="ProfileToken" type="tt:ReferenceToken"/>
            <xs:element name="PresetTourToken" type="tt:ReferenceToken"/>
            <xs:element name="Operation" type="tt:PTZPresetTourOperation"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="OperatePresetTourResponse">
        <xs:complexType>
          <xs:sequence/>
        </xs:complexType>
      </xs:element>
      <xs:element name="RemovePresetTour">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="ProfileToken" type="tt:ReferenceToken"/>
            <xs:element name="PresetTourToken" type="tt:ReferenceToken"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="RemovePresetTourResponse">
        <xs:complexType>
          <xs:sequence/>
        </xs:complexType>
      </xs:element>
      <xs:element name="GetCompatibleConfigurations">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="ProfileToken" type="tt:ReferenceToken"/>
          </xs:sequence>
        </xs:complexType>

```

```

        </xs:element>
        <xs:element name="GetCompatibleConfigurationsResponse">
            <xs:complexType>
                <xs:sequence>
                    <xs:element name="PTZConfiguration" type="tt:PTZConfiguration"
minOccurs="0" maxOccurs="unbounded"/>
                </xs:sequence>
            </xs:complexType>
        </xs:element>
    </xs:schema>
</wsdl:types>
<wsdl:message name="GetServiceCapabilitiesRequest">
    <wsdl:part name="parameters" element="tptz:GetServiceCapabilities"/>
</wsdl:message>
<wsdl:message name="GetServiceCapabilitiesResponse">
    <wsdl:part name="parameters" element="tptz:GetServiceCapabilitiesResponse"/>
</wsdl:message>
<wsdl:message name="GetNodesRequest">
    <wsdl:part name="parameters" element="tptz:GetNodes"/>
</wsdl:message>
<wsdl:message name="GetNodesResponse">
    <wsdl:part name="parameters" element="tptz:GetNodesResponse"/>
</wsdl:message>
<wsdl:message name="GetNodeRequest">
    <wsdl:part name="parameters" element="tptz:GetNode"/>
</wsdl:message>
<wsdl:message name="GetNodeResponse">
    <wsdl:part name="parameters" element="tptz:GetNodeResponse"/>
</wsdl:message>
<wsdl:message name="GetConfigurationsRequest">
    <wsdl:part name="parameters" element="tptz:GetConfigurations"/>
</wsdl:message>
<wsdl:message name="GetConfigurationsResponse">
    <wsdl:part name="parameters" element="tptz:GetConfigurationsResponse"/>
</wsdl:message>
<wsdl:message name="GetConfigurationRequest">
    <wsdl:part name="parameters" element="tptz:GetConfiguration"/>
</wsdl:message>
<wsdl:message name="GetConfigurationResponse">
    <wsdl:part name="parameters" element="tptz:GetConfigurationResponse"/>
</wsdl:message>
<wsdl:message name="SetConfigurationRequest">
    <wsdl:part name="parameters" element="tptz:SetConfiguration"/>
</wsdl:message>
<wsdl:message name="SetConfigurationResponse">
    <wsdl:part name="parameters" element="tptz:SetConfigurationResponse"/>
</wsdl:message>
<wsdl:message name="GetConfigurationOptionsRequest">
    <wsdl:part name="parameters" element="tptz:GetConfigurationOptions"/>
</wsdl:message>
<wsdl:message name="GetConfigurationOptionsResponse">
    <wsdl:part name="parameters" element="tptz:GetConfigurationOptionsResponse"/>
</wsdl:message>
<wsdl:message name="GetPresetsRequest">
    <wsdl:part name="parameters" element="tptz:GetPresets"/>
</wsdl:message>
<wsdl:message name="GetPresetsResponse">
    <wsdl:part name="parameters" element="tptz:GetPresetsResponse"/>
</wsdl:message>
<wsdl:message name="SetPresetRequest">
    <wsdl:part name="parameters" element="tptz:SetPreset"/>
</wsdl:message>
<wsdl:message name="SetPresetResponse">
    <wsdl:part name="parameters" element="tptz:SetPresetResponse"/>
</wsdl:message>
<wsdl:message name="RemovePresetRequest">
    <wsdl:part name="parameters" element="tptz:RemovePreset"/>
</wsdl:message>
<wsdl:message name="RemovePresetResponse">
    <wsdl:part name="parameters" element="tptz:RemovePresetResponse"/>
</wsdl:message>
<wsdl:message name="GotoPresetRequest">
    <wsdl:part name="parameters" element="tptz:GotoPreset"/>
</wsdl:message>
<wsdl:message name="GotoPresetResponse">
    <wsdl:part name="parameters" element="tptz:GotoPresetResponse"/>
</wsdl:message>
<wsdl:message name="GetStatusRequest">

```

```
<wsdl:part name="parameters" element="tptz:GetStatus"/>
</wsdl:message>
<wsdl:message name="GetStatusResponse">
  <wsdl:part name="parameters" element="tptz:GetStatusResponse"/>
</wsdl:message>
<wsdl:message name="SendAuxiliaryCommandRequest">
  <wsdl:part name="parameters" element="tptz:SendAuxiliaryCommand"/>
</wsdl:message>
<wsdl:message name="SendAuxiliaryCommandResponse">
  <wsdl:part name="parameters" element="tptz:SendAuxiliaryCommandResponse"/>
</wsdl:message>
<wsdl:message name="GotoHomePositionRequest">
  <wsdl:part name="parameters" element="tptz:GotoHomePosition"/>
</wsdl:message>
<wsdl:message name="GotoHomePositionResponse">
  <wsdl:part name="parameters" element="tptz:GotoHomePositionResponse"/>
</wsdl:message>
<wsdl:message name="SetHomePositionRequest">
  <wsdl:part name="parameters" element="tptz:SetHomePosition"/>
</wsdl:message>
<wsdl:message name="SetHomePositionResponse">
  <wsdl:part name="parameters" element="tptz:SetHomePositionResponse"/>
</wsdl:message>
<wsdl:message name="ContinuousMoveRequest">
  <wsdl:part name="parameters" element="tptz:ContinuousMove"/>
</wsdl:message>
<wsdl:message name="ContinuousMoveResponse">
  <wsdl:part name="parameters" element="tptz:ContinuousMoveResponse"/>
</wsdl:message>
<wsdl:message name="RelativeMoveRequest">
  <wsdl:part name="parameters" element="tptz:RelativeMove"/>
</wsdl:message>
<wsdl:message name="RelativeMoveResponse">
  <wsdl:part name="parameters" element="tptz:RelativeMoveResponse"/>
</wsdl:message>
<wsdl:message name="AbsoluteMoveRequest">
  <wsdl:part name="parameters" element="tptz:AbsoluteMove"/>
</wsdl:message>
<wsdl:message name="AbsoluteMoveResponse">
  <wsdl:part name="parameters" element="tptz:AbsoluteMoveResponse"/>
</wsdl:message>
<wsdl:message name="StopRequest">
  <wsdl:part name="parameters" element="tptz:Stop"/>
</wsdl:message>
<wsdl:message name="StopResponse">
  <wsdl:part name="parameters" element="tptz:StopResponse"/>
</wsdl:message>
<wsdl:message name="GetPresetToursRequest">
  <wsdl:part name="parameters" element="tptz:GetPresetTours"/>
</wsdl:message>
<wsdl:message name="GetPresetToursResponse">
  <wsdl:part name="parameters" element="tptz:GetPresetToursResponse"/>
</wsdl:message>
<wsdl:message name="GetPresetTourRequest">
  <wsdl:part name="parameters" element="tptz:GetPresetTour"/>
</wsdl:message>
<wsdl:message name="GetPresetTourResponse">
  <wsdl:part name="parameters" element="tptz:GetPresetTourResponse"/>
</wsdl:message>
<wsdl:message name="GetPresetTourOptionsRequest">
  <wsdl:part name="parameters" element="tptz:GetPresetTourOptions"/>
</wsdl:message>
<wsdl:message name="GetPresetTourOptionsResponse">
  <wsdl:part name="parameters" element="tptz:GetPresetTourOptionsResponse"/>
</wsdl:message>
<wsdl:message name="CreatePresetTourRequest">
  <wsdl:part name="parameters" element="tptz:CreatePresetTour"/>
</wsdl:message>
<wsdl:message name="CreatePresetTourResponse">
  <wsdl:part name="parameters" element="tptz:CreatePresetTourResponse"/>
</wsdl:message>
<wsdl:message name="ModifyPresetTourRequest">
  <wsdl:part name="parameters" element="tptz:ModifyPresetTour"/>
</wsdl:message>
<wsdl:message name="ModifyPresetTourResponse">
  <wsdl:part name="parameters" element="tptz:ModifyPresetTourResponse"/>
</wsdl:message>
<wsdl:message name="OperatePresetTourRequest">
```

```

    <wsdl:part name="parameters" element="tptz:OperatePresetTour"/>
</wsdl:message>
<wsdl:message name="OperatePresetTourResponse">
    <wsdl:part name="parameters" element="tptz:OperatePresetTourResponse"/>
</wsdl:message>
<wsdl:message name="RemovePresetTourRequest">
    <wsdl:part name="parameters" element="tptz:RemovePresetTour"/>
</wsdl:message>
<wsdl:message name="RemovePresetTourResponse">
    <wsdl:part name="parameters" element="tptz:RemovePresetTourResponse"/>
</wsdl:message>
<wsdl:message name="GetCompatibleConfigurationsRequest">
    <wsdl:part name="parameters" element="tptz:GetCompatibleConfigurations"/>
</wsdl:message>
<wsdl:message name="GetCompatibleConfigurationsResponse">
    <wsdl:part name="parameters" element="tptz:GetCompatibleConfigurationsResponse"/>
</wsdl:message>
<wsdl:portType name="PTZ">
    <wsdl:operation name="GetServiceCapabilities">
        <wsdl:input message="tptz:GetServiceCapabilitiesRequest"/>
        <wsdl:output message="tptz:GetServiceCapabilitiesResponse"/>
    </wsdl:operation>
    <wsdl:operation name="GetNodes">
        <wsdl:input message="tptz:GetNodesRequest"/>
        <wsdl:output message="tptz:GetNodesResponse"/>
    </wsdl:operation>
    <wsdl:operation name="GetNode">
        <wsdl:input message="tptz:GetNodeRequest"/>
        <wsdl:output message="tptz:GetNodeResponse"/>
    </wsdl:operation>
    <wsdl:operation name="GetConfiguration">
        <wsdl:input message="tptz:GetConfigurationRequest"/>
        <wsdl:output message="tptz:GetConfigurationResponse"/>
    </wsdl:operation>
    <wsdl:operation name="GetConfigurations">
        <wsdl:input message="tptz:GetConfigurationsRequest"/>
        <wsdl:output message="tptz:GetConfigurationsResponse"/>
    </wsdl:operation>
    <wsdl:operation name="SetConfiguration">
        <wsdl:input message="tptz:SetConfigurationRequest"/>
        <wsdl:output message="tptz:SetConfigurationResponse"/>
    </wsdl:operation>
    <wsdl:operation name="GetConfigurationOptions">
        <wsdl:input message="tptz:GetConfigurationOptionsRequest"/>
        <wsdl:output message="tptz:GetConfigurationOptionsResponse"/>
    </wsdl:operation>
    <wsdl:operation name="SendAuxiliaryCommand">
        <wsdl:input message="tptz:SendAuxiliaryCommandRequest"/>
        <wsdl:output message="tptz:SendAuxiliaryCommandResponse"/>
    </wsdl:operation>
    <wsdl:operation name="GetPresets">
        <wsdl:input message="tptz:GetPresetsRequest"/>
        <wsdl:output message="tptz:GetPresetsResponse"/>
    </wsdl:operation>
    <wsdl:operation name="SetPreset">
        <wsdl:input message="tptz:SetPresetRequest"/>
        <wsdl:output message="tptz:SetPresetResponse"/>
    </wsdl:operation>
    <wsdl:operation name="RemovePreset">
        <wsdl:input message="tptz:RemovePresetRequest"/>
        <wsdl:output message="tptz:RemovePresetResponse"/>
    </wsdl:operation>
    <wsdl:operation name="GotoPreset">
        <wsdl:input message="tptz:GotoPresetRequest"/>
        <wsdl:output message="tptz:GotoPresetResponse"/>
    </wsdl:operation>
    <wsdl:operation name="GotoHomePosition">
        <wsdl:input message="tptz:GotoHomePositionRequest"/>
        <wsdl:output message="tptz:GotoHomePositionResponse"/>
    </wsdl:operation>
    <wsdl:operation name="SetHomePosition">
        <wsdl:input message="tptz:SetHomePositionRequest"/>
        <wsdl:output message="tptz:SetHomePositionResponse"/>
    </wsdl:operation>
    <wsdl:operation name="ContinuousMove">
        <wsdl:input message="tptz:ContinuousMoveRequest"/>
        <wsdl:output message="tptz:ContinuousMoveResponse"/>
    </wsdl:operation>

```

```

<wsdl:operation name="RelativeMove">
  <wsdl:input message="tptz:RelativeMoveRequest"/>
  <wsdl:output message="tptz:RelativeMoveResponse"/>
</wsdl:operation>
<wsdl:operation name="GetStatus">
  <wsdl:input message="tptz:GetStatusRequest"/>
  <wsdl:output message="tptz:GetStatusResponse"/>
</wsdl:operation>
<wsdl:operation name="AbsoluteMove">
  <wsdl:input message="tptz:AbsoluteMoveRequest"/>
  <wsdl:output message="tptz:AbsoluteMoveResponse"/>
</wsdl:operation>
<wsdl:operation name="Stop">
  <wsdl:input message="tptz:StopRequest"/>
  <wsdl:output message="tptz:StopResponse"/>
</wsdl:operation>
<wsdl:operation name="GetPresetTours">
  <wsdl:input message="tptz:GetPresetToursRequest"/>
  <wsdl:output message="tptz:GetPresetToursResponse"/>
</wsdl:operation>
<wsdl:operation name="GetPresetTour">
  <wsdl:input message="tptz:GetPresetTourRequest"/>
  <wsdl:output message="tptz:GetPresetTourResponse"/>
</wsdl:operation>
<wsdl:operation name="GetPresetTourOptions">
  <wsdl:input message="tptz:GetPresetTourOptionsRequest"/>
  <wsdl:output message="tptz:GetPresetTourOptionsResponse"/>
</wsdl:operation>
<wsdl:operation name="CreatePresetTour">
  <wsdl:input message="tptz:CreatePresetTourRequest"/>
  <wsdl:output message="tptz:CreatePresetTourResponse"/>
</wsdl:operation>
<wsdl:operation name="ModifyPresetTour">
  <wsdl:input message="tptz:ModifyPresetTourRequest"/>
  <wsdl:output message="tptz:ModifyPresetTourResponse"/>
</wsdl:operation>
<wsdl:operation name="OperatePresetTour">
  <wsdl:input message="tptz:OperatePresetTourRequest"/>
  <wsdl:output message="tptz:OperatePresetTourResponse"/>
</wsdl:operation>
<wsdl:operation name="RemovePresetTour">
  <wsdl:input message="tptz:RemovePresetTourRequest"/>
  <wsdl:output message="tptz:RemovePresetTourResponse"/>
</wsdl:operation>
<wsdl:operation name="GetCompatibleConfigurations">
  <wsdl:input message="tptz:GetCompatibleConfigurationsRequest"/>
  <wsdl:output message="tptz:GetCompatibleConfigurationsResponse"/>
</wsdl:operation>
</wsdl:portType>
<wsdl:binding name="PTZBinding" type="tptz:PTZ">
  <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
  <wsdl:operation name="GetServiceCapabilities">
    <soap:operation
      soapAction="http://www.onvif.org/ver20/ptz/wsdl/GetServiceCapabilities"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="GetConfigurations">
    <soap:operation
      soapAction="http://www.onvif.org/ver20/ptz/wsdl/GetConfigurations"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="GetPresets">
    <soap:operation soapAction="http://www.onvif.org/ver20/ptz/wsdl/GetPresets"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>

```



```

        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="SetPreset">
        <soap:operation soapAction="http://www.onvif.org/ver20/ptz/wsdl/SetPreset"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="RemovePreset">
        <soap:operation
soapAction="http://www.onvif.org/ver20/ptz/wsdl/RemovePreset"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GotoPreset">
        <soap:operation soapAction="http://www.onvif.org/ver20/ptz/wsdl/GotoPreset"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetStatus">
        <soap:operation soapAction="http://www.onvif.org/ver20/ptz/wsdl/GetStatus"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetConfiguration">
        <soap:operation
soapAction="http://www.onvif.org/ver20/ptz/wsdl/GetConfiguration"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetNodes">
        <soap:operation soapAction="http://www.onvif.org/ver20/ptz/wsdl/GetNodes"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetNode">
        <soap:operation soapAction="http://www.onvif.org/ver20/ptz/wsdl/GetNode"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="SetConfiguration">
        <soap:operation
soapAction="http://www.onvif.org/ver20/ptz/wsdl/SetConfiguration"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetConfigurationOptions">

```



```
<soap:operation
soapAction="http://www.onvif.org/ver20/ptz/wsdl/GetConfigurationOptions"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GotoHomePosition">
  <soap:operation
soapAction="http://www.onvif.org/ver20/ptz/wsdl/GotoHomePosition"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
<wsdl:operation name="SetHomePosition">
  <soap:operation
soapAction="http://www.onvif.org/ver20/ptz/wsdl/SetHomePosition"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
<wsdl:operation name="ContinuousMove">
  <soap:operation
soapAction="http://www.onvif.org/ver20/ptz/wsdl/ContinuousMove"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
<wsdl:operation name="RelativeMove">
  <soap:operation
soapAction="http://www.onvif.org/ver20/ptz/wsdl/RelativeMove"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
<wsdl:operation name="SendAuxiliaryCommand">
  <soap:operation
soapAction="http://www.onvif.org/ver20/ptz/wsdl/SendAuxiliaryCommand"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
<wsdl:operation name="AbsoluteMove">
  <soap:operation
soapAction="http://www.onvif.org/ver20/ptz/wsdl/AbsoluteMove"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
<wsdl:operation name="Stop">
  <soap:operation soapAction="http://www.onvif.org/ver20/ptz/wsdl/Stop"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
```

```

<wsdl:operation name="GetPresetTours">
  <soap:operation
soapAction="http://www.onvif.org/ver20/ptz/wsdl/GetPresetTours"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetPresetTour">
  <soap:operation
soapAction="http://www.onvif.org/ver20/ptz/wsdl/GetPresetTour"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetPresetTourOptions">
  <soap:operation
soapAction="http://www.onvif.org/ver20/ptz/wsdl/GetPresetTourOptions"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="CreatePresetTour">
  <soap:operation
soapAction="http://www.onvif.org/ver20/ptz/wsdl/CreatePresetTour"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="ModifyPresetTour">
  <soap:operation
soapAction="http://www.onvif.org/ver20/ptz/wsdl/ModifyPresetTour"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="OperatePresetTour">
  <soap:operation
soapAction="http://www.onvif.org/ver20/ptz/wsdl/OperatePresetTour"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="RemovePresetTour">
  <soap:operation
soapAction="http://www.onvif.org/ver20/ptz/wsdl/RemovePresetTour"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetCompatibleConfigurations">
  <soap:operation
soapAction="http://www.onvif.org/ver20/ptz/wsdl/GetCompatibleConfigurations"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>

```

```
        </wsdl:output>  
    </wsdl:operation>  
    </wsdl:binding>  
</wsdl:definitions>
```

IECNORM.COM : Click to view the full PDF of IEC 62676-2-31:2019

F.6 Analytics

```
<?xml version="1.0" encoding="utf-8"?><?xml-stylesheet type="text/xsl"
href="../../../ver20/util/onvif-wsdl-viewer.xsl"?><wsdl:definitions
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap12/"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:tan="http://www.onvif.org/ver20/analytics/wsdl"
targetNamespace="http://www.onvif.org/ver20/analytics/wsdl">
  <wsdl:types>
    <xs:schema xmlns:tt="http://www.onvif.org/ver10/schema"
targetNamespace="http://www.onvif.org/ver20/analytics/wsdl"
elementFormDefault="qualified" version="16.12">
      <xs:import namespace="http://www.onvif.org/ver10/schema"
schemaLocation="../../../ver10/schema/onvif.xsd"/>
      <xs:element name="GetServiceCapabilities">
        <xs:complexType>
          <xs:sequence/>
        </xs:complexType>
      </xs:element>
      <xs:element name="GetServiceCapabilitiesResponse">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Capabilities" type="tan:Capabilities"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="Capabilities">
        <xs:sequence>
          <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="RuleSupport" type="xs:boolean"/>
        <xs:attribute name="AnalyticsModuleSupport" type="xs:boolean"/>
        <xs:attribute name="CellBasedSceneDescriptionSupported" type="xs:boolean"/>
        <xs:attribute name="RuleOptionsSupported" type="xs:boolean"/>
        <xs:anyAttribute processContents="lax"/>
      </xs:complexType>
      <xs:element name="Capabilities" type="tan:Capabilities"/>
      <xs:element name="GetSupportedRules">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="ConfigurationToken" type="tt:ReferenceToken"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="GetSupportedRulesResponse">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="SupportedRules" type="tt:SupportedRules"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="CreateRules">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="ConfigurationToken" type="tt:ReferenceToken"/>
            <xs:element name="Rule" type="tt:Config" minOccurs="1"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="CreateRulesResponse">
        <xs:complexType/>
      </xs:element>
      <xs:element name="DeleteRules">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="ConfigurationToken" type="tt:ReferenceToken"/>
            <xs:element name="RuleName" type="xs:string" minOccurs="1"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="DeleteRulesResponse">
        <xs:complexType/>
      </xs:element>
    </xs:schema>
  </wsdl:types>

```

```

    <xs:element name="ModifyRules">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="ConfigurationToken" type="tt:ReferenceToken"/>
          <xs:element name="Rule" type="tt:Config" minOccurs="1"
maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="ModifyRulesResponse">
      <xs:complexType/>
    </xs:element>
    <xs:element name="GetRules">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="ConfigurationToken" type="tt:ReferenceToken"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="GetRulesResponse">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="Rule" type="tt:Config" minOccurs="0"
maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="GetRuleOptions">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="RuleType" type="xs:QName" minOccurs="0"/>
          <xs:element name="ConfigurationToken" type="tt:ReferenceToken"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="GetRuleOptionsResponse">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="RuleOptions" type="tan:ConfigOptions" minOccurs="0"
maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:complexType name="ConfigOptions">
      <xs:sequence>
        <xs:any namespace="##any"/>
      </xs:sequence>
      <xs:attribute name="Name" type="xs:string" use="required"/>
      <xs:attribute name="Type" type="xs:QName" use="required"/>
      <xs:anyAttribute processContents="lax"/>
    </xs:complexType>
    <xs:element name="GetSupportedAnalyticsModules">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="ConfigurationToken" type="tt:ReferenceToken"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="GetSupportedAnalyticsModulesResponse">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="SupportedAnalyticsModules"
type="tt:SupportedAnalyticsModules"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="CreateAnalyticsModules">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="ConfigurationToken" type="tt:ReferenceToken"/>
          <xs:element name="AnalyticsModule" type="tt:Config"
maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="CreateAnalyticsModulesResponse">
      <xs:complexType/>
    </xs:element>

```

```

    <xs:element name="DeleteAnalyticsModules">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="ConfigurationToken" type="tt:ReferenceToken"/>
          <xs:element name="AnalyticsModuleName" type="xs:string"
maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="DeleteAnalyticsModulesResponse">
      <xs:complexType/>
    </xs:element>
    <xs:element name="ModifyAnalyticsModules">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="ConfigurationToken" type="tt:ReferenceToken"/>
          <xs:element name="AnalyticsModule" type="tt:Config"
maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="ModifyAnalyticsModulesResponse">
      <xs:complexType/>
    </xs:element>
    <xs:element name="GetAnalyticsModules">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="ConfigurationToken" type="tt:ReferenceToken"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="GetAnalyticsModulesResponse">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="AnalyticsModule" type="tt:Config" minOccurs="0"
maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:schema>
</wsdl:types>
<wsdl:message name="GetServiceCapabilitiesRequest">
  <wsdl:part name="parameters" element="tan:GetServiceCapabilities"/>
</wsdl:message>
<wsdl:message name="GetServiceCapabilitiesResponse">
  <wsdl:part name="parameters" element="tan:GetServiceCapabilitiesResponse"/>
</wsdl:message>
<wsdl:message name="GetSupportedRulesRequest">
  <wsdl:part name="parameters" element="tan:GetSupportedRules"/>
</wsdl:message>
<wsdl:message name="GetSupportedRulesResponse">
  <wsdl:part name="parameters" element="tan:GetSupportedRulesResponse"/>
</wsdl:message>
<wsdl:message name="CreateRulesRequest">
  <wsdl:part name="parameters" element="tan:CreateRules"/>
</wsdl:message>
<wsdl:message name="CreateRulesResponse">
  <wsdl:part name="parameters" element="tan:CreateRulesResponse"/>
</wsdl:message>
<wsdl:message name="DeleteRulesRequest">
  <wsdl:part name="parameters" element="tan:DeleteRules"/>
</wsdl:message>
<wsdl:message name="DeleteRulesResponse">
  <wsdl:part name="parameters" element="tan:DeleteRulesResponse"/>
</wsdl:message>
<wsdl:message name="GetRulesRequest">
  <wsdl:part name="parameters" element="tan:GetRules"/>
</wsdl:message>
<wsdl:message name="GetRulesResponse">
  <wsdl:part name="parameters" element="tan:GetRulesResponse"/>
</wsdl:message>
<wsdl:message name="GetRuleOptionsRequest">
  <wsdl:part name="parameters" element="tan:GetRuleOptions"/>
</wsdl:message>
<wsdl:message name="GetRuleOptionsResponse">
  <wsdl:part name="parameters" element="tan:GetRuleOptionsResponse"/>
</wsdl:message>
<wsdl:message name="GetSupportedAnalyticsModulesResponse">

```

```

    <wsdl:part name="parameters" element="tan:GetSupportedAnalyticsModulesResponse"/>
  </wsdl:message>
  <wsdl:message name="GetSupportedAnalyticsModulesRequest">
    <wsdl:part name="parameters" element="tan:GetSupportedAnalyticsModules"/>
  </wsdl:message>
  <wsdl:message name="CreateAnalyticsModulesRequest">
    <wsdl:part name="parameters" element="tan:CreateAnalyticsModules"/>
  </wsdl:message>
  <wsdl:message name="CreateAnalyticsModulesResponse">
    <wsdl:part name="parameters" element="tan:CreateAnalyticsModulesResponse"/>
  </wsdl:message>
  <wsdl:message name="DeleteAnalyticsModulesRequest">
    <wsdl:part name="parameters" element="tan:DeleteAnalyticsModules"/>
  </wsdl:message>
  <wsdl:message name="DeleteAnalyticsModulesResponse">
    <wsdl:part name="parameters" element="tan:DeleteAnalyticsModulesResponse"/>
  </wsdl:message>
  <wsdl:message name="GetAnalyticsModulesRequest">
    <wsdl:part name="parameters" element="tan:GetAnalyticsModules"/>
  </wsdl:message>
  <wsdl:message name="GetAnalyticsModulesResponse">
    <wsdl:part name="parameters" element="tan:GetAnalyticsModulesResponse"/>
  </wsdl:message>
  <wsdl:message name="ModifyRulesRequest">
    <wsdl:part name="parameters" element="tan:ModifyRules"/>
  </wsdl:message>
  <wsdl:message name="ModifyRulesResponse">
    <wsdl:part name="parameters" element="tan:ModifyRulesResponse"/>
  </wsdl:message>
  <wsdl:message name="ModifyAnalyticsModulesRequest">
    <wsdl:part name="parameters" element="tan:ModifyAnalyticsModules"/>
  </wsdl:message>
  <wsdl:message name="ModifyAnalyticsModulesResponse">
    <wsdl:part name="parameters" element="tan:ModifyAnalyticsModulesResponse"/>
  </wsdl:message>
  <wsdl:portType name="RuleEnginePort">
    <wsdl:operation name="GetSupportedRules">
      <wsdl:input message="tan:GetSupportedRulesRequest"/>
      <wsdl:output message="tan:GetSupportedRulesResponse"/>
    </wsdl:operation>
    <wsdl:operation name="CreateRules">
      <wsdl:input message="tan:CreateRulesRequest"/>
      <wsdl:output message="tan:CreateRulesResponse"/>
    </wsdl:operation>
    <wsdl:operation name="DeleteRules">
      <wsdl:input message="tan:DeleteRulesRequest"/>
      <wsdl:output message="tan:DeleteRulesResponse"/>
    </wsdl:operation>
    <wsdl:operation name="GetRules">
      <wsdl:input message="tan:GetRulesRequest"/>
      <wsdl:output message="tan:GetRulesResponse"/>
    </wsdl:operation>
    <wsdl:operation name="GetRuleOptions">
      <wsdl:input message="tan:GetRuleOptionsRequest"/>
      <wsdl:output message="tan:GetRuleOptionsResponse"/>
    </wsdl:operation>
    <wsdl:operation name="ModifyRules">
      <wsdl:input message="tan:ModifyRulesRequest"/>
      <wsdl:output message="tan:ModifyRulesResponse"/>
    </wsdl:operation>
  </wsdl:portType>
  <wsdl:portType name="AnalyticsEnginePort">
    <wsdl:operation name="GetServiceCapabilities">
      <wsdl:input message="tan:GetServiceCapabilitiesRequest"/>
      <wsdl:output message="tan:GetServiceCapabilitiesResponse"/>
    </wsdl:operation>
    <wsdl:operation name="GetSupportedAnalyticsModules">
      <wsdl:input message="tan:GetSupportedAnalyticsModulesRequest"/>
      <wsdl:output message="tan:GetSupportedAnalyticsModulesResponse"/>
    </wsdl:operation>
    <wsdl:operation name="CreateAnalyticsModules">
      <wsdl:input message="tan:CreateAnalyticsModulesRequest"/>
      <wsdl:output message="tan:CreateAnalyticsModulesResponse"/>
    </wsdl:operation>
    <wsdl:operation name="DeleteAnalyticsModules">
      <wsdl:input message="tan:DeleteAnalyticsModulesRequest"/>
      <wsdl:output message="tan:DeleteAnalyticsModulesResponse"/>
    </wsdl:operation>
  </wsdl:portType>

```

```

<wsdl:operation name="GetAnalyticsModules">
  <wsdl:input message="tan:GetAnalyticsModulesRequest"/>
  <wsdl:output message="tan:GetAnalyticsModulesResponse"/>
</wsdl:operation>
<wsdl:operation name="ModifyAnalyticsModules">
  <wsdl:input message="tan:ModifyAnalyticsModulesRequest"/>
  <wsdl:output message="tan:ModifyAnalyticsModulesResponse"/>
</wsdl:operation>
</wsdl:portType>
<wsdl:binding name="RuleEngineBinding" type="tan:RuleEnginePort">
  <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
  <wsdl:operation name="GetSupportedRules">
    <soap:operation
      soapAction="http://www.onvif.org/ver20/analytics/wsdl/GetSupportedRules"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="CreateRules">
    <soap:operation
      soapAction="http://www.onvif.org/ver20/analytics/wsdl/CreateRules"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="DeleteRules">
    <soap:operation
      soapAction="http://www.onvif.org/ver20/analytics/wsdl/DeleteRules"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="GetRules">
    <soap:operation
      soapAction="http://www.onvif.org/ver20/analytics/wsdl/GetRules"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="GetRuleOptions">
    <soap:operation
      soapAction="http://www.onvif.org/ver20/analytics/wsdl/GetRuleOptions"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="ModifyRules">
    <soap:operation
      soapAction="http://www.onvif.org/ver20/analytics/wsdl/ModifyRules"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
</wsdl:binding>
<wsdl:binding name="AnalyticsEngineBinding" type="tan:AnalyticsEnginePort">
  <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
  <wsdl:operation name="GetServiceCapabilities">
    <soap:operation
      soapAction="http://www.onvif.org/ver20/analytics/wsdl/GetServiceCapabilities"/>
    <wsdl:input>

```



```
<soap:body use="literal"/>
</wsdl:input>
<wsdl:output>
  <soap:body use="literal"/>
</wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetSupportedAnalyticsModules">
  <soap:operation
soapAction="http://www.onvif.org/ver20/analytics/wsdl/GetSupportedAnalyticsModules"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="CreateAnalyticsModules">
  <soap:operation
soapAction="http://www.onvif.org/ver20/analytics/wsdl/CreateAnalyticsModules"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="DeleteAnalyticsModules">
  <soap:operation
soapAction="http://www.onvif.org/ver20/analytics/wsdl/DeleteAnalyticsModules"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetAnalyticsModules">
  <soap:operation
soapAction="http://www.onvif.org/ver20/analytics/wsdl/GetAnalyticsModules"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="ModifyAnalyticsModules">
  <soap:operation
soapAction="http://www.onvif.org/ver20/analytics/wsdl/ModifyAnalyticsModules"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
</wsdl:binding>
</wsdl:definitions>
```

F.7 Common schema

This section extends the generic data types as defined by Clause B.4 of IEC 60839-11-31.

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:tt="http://www.onvif.org/ver10/schema" xmlns:wsnt="http://docs.oasis-
open.org/wsn/b-2" targetNamespace="http://www.onvif.org/ver10/schema"
elementFormDefault="qualified" version="18.05">
  <xs:simpleType name="ReferenceToken">
    <xs:restriction base="xs:string">
      <xs:maxLength value="64"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:complexType name="Vector2D">
    <xs:attribute name="x" type="xs:float" use="required"/>
    <xs:attribute name="y" type="xs:float" use="required"/>
    <xs:attribute name="space" type="xs:anyURI" use="optional"/>
  </xs:complexType>
  <xs:complexType name="Vector1D">
    <xs:attribute name="x" type="xs:float" use="required"/>
    <xs:attribute name="space" type="xs:anyURI" use="optional"/>
  </xs:complexType>
  <xs:complexType name="PTZVector">
    <xs:sequence>
      <xs:element name="PanTilt" type="tt:Vector2D" minOccurs="0"/>
      <xs:element name="Zoom" type="tt:Vector1D" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="PTZStatus">
    <xs:sequence>
      <xs:element name="Position" type="tt:PTZVector" minOccurs="0"/>
      <xs:element name="MoveStatus" type="tt:PTZMoveStatus" minOccurs="0"/>
      <xs:element name="Error" type="xs:string" minOccurs="0"/>
      <xs:element name="UtcTime" type="xs:dateTime"/>
      <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:anyAttribute processContents="lax"/>
  </xs:complexType>
  <xs:complexType name="PTZMoveStatus">
    <xs:sequence>
      <xs:element name="PanTilt" type="tt:MoveStatus" minOccurs="0"/>
      <xs:element name="Zoom" type="tt:MoveStatus" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
  <xs:simpleType name="MoveStatus">
    <xs:restriction base="xs:string">
      <xs:enumeration value="IDLE"/>
      <xs:enumeration value="MOVING"/>
      <xs:enumeration value="UNKNOWN"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:complexType name="Vector">
    <xs:attribute name="x" type="xs:float"/>
    <xs:attribute name="y" type="xs:float"/>
  </xs:complexType>
  <xs:complexType name="Rectangle">
    <xs:attribute name="bottom" type="xs:float"/>
    <xs:attribute name="top" type="xs:float"/>
    <xs:attribute name="right" type="xs:float"/>
    <xs:attribute name="left" type="xs:float"/>
  </xs:complexType>
  <xs:complexType name="Polygon">
    <xs:sequence>
      <xs:element name="Point" type="tt:Vector" minOccurs="3" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Polygon" type="tt:Polygon"/>
  <xs:complexType name="Color">
    <xs:attribute name="X" type="xs:float" use="required"/>
    <xs:attribute name="Y" type="xs:float" use="required"/>
    <xs:attribute name="Z" type="xs:float" use="required"/>
    <xs:attribute name="Colorspace" type="xs:anyURI"/>
  </xs:complexType>
  <xs:complexType name="ColorCovariance">
```

```

<xs:attribute name="XX" type="xs:float" use="required"/>
<xs:attribute name="YY" type="xs:float" use="required"/>
<xs:attribute name="ZZ" type="xs:float" use="required"/>
<xs:attribute name="XY" type="xs:float"/>
<xs:attribute name="XZ" type="xs:float"/>
<xs:attribute name="YZ" type="xs:float"/>
<xs:attribute name="Colorspace" type="xs:anyURI"/>
</xs:complexType>
<xs:complexType name="Transformation">
  <xs:sequence>
    <xs:element name="Translate" type="tt:Vector" minOccurs="0"/>
    <xs:element name="Scale" type="tt:Vector" minOccurs="0"/>
    <xs:element name="Extension" type="tt:TransformationExtension" minOccurs="0"/>
  </xs:sequence>
  <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="TransformationExtension">
  <xs:sequence>
    <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="GeoLocation">
  <xs:sequence>
    <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="lon" type="xs:double"/>
  <xs:attribute name="lat" type="xs:double"/>
  <xs:attribute name="elevation" type="xs:float"/>
  <xs:anyAttribute/>
</xs:complexType>
<xs:complexType name="GeoOrientation">
  <xs:sequence>
    <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="roll" type="xs:float"/>
  <xs:attribute name="pitch" type="xs:float"/>
  <xs:attribute name="yaw" type="xs:float"/>
  <xs:anyAttribute/>
</xs:complexType>
<xs:complexType name="LocalLocation">
  <xs:sequence>
    <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="x" type="xs:float"/>
  <xs:attribute name="y" type="xs:float"/>
  <xs:attribute name="z" type="xs:float"/>
  <xs:anyAttribute/>
</xs:complexType>
<xs:complexType name="LocalOrientation">
  <xs:sequence>
    <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="pan" type="xs:float"/>
  <xs:attribute name="tilt" type="xs:float"/>
  <xs:attribute name="roll" type="xs:float"/>
  <xs:anyAttribute/>
</xs:complexType>
<xs:simpleType name="Entity">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Device"/>
    <xs:enumeration value="VideoSource"/>
    <xs:enumeration value="AudioSource"/>
  </xs:restriction>
</xs:simpleType>
<xs:complexType name="LocationEntity">
  <xs:sequence>
    <xs:element name="GeoLocation" type="tt:GeoLocation" minOccurs="0"/>
    <xs:element name="GeoOrientation" type="tt:GeoOrientation" minOccurs="0"/>
    <xs:element name="LocalLocation" type="tt:LocalLocation" minOccurs="0"/>
    <xs:element name="LocalOrientation" type="tt:LocalOrientation" minOccurs="0"/>
  </xs:sequence>
  <xs:attribute name="Entity" type="xs:string"/>
  <xs:attribute name="Token" type="tt:ReferenceToken"/>

```

```

    <xs:attribute name="Fixed" type="xs:boolean"/>
    <xs:attribute name="GeoSource" type="xs:anyURI"/>
    <xs:attribute name="AutoGeo" type="xs:boolean"/>
  </xs:complexType>
</xs:schema>

<xs:schema xmlns:tt="http://www.onvif.org/ver10/schema"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:xmime="http://www.w3.org/2005/05/xmlmime" xmlns:wsnt="http://docs.oasis-
  open.org/wsn/b-2" xmlns:xop="http://www.w3.org/2004/08/xop/include"
  xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope"
  targetNamespace="http://www.onvif.org/ver10/schema">
  <xs:include schemaLocation="common.xsd"/>
  <xs:complexType name="IntRectangle">
    <xs:attribute name="x" type="xs:int" use="required"/>
    <xs:attribute name="y" type="xs:int" use="required"/>
    <xs:attribute name="width" type="xs:int" use="required"/>
    <xs:attribute name="height" type="xs:int" use="required"/>
  </xs:complexType>
  <xs:complexType name="IntRectangleRange">
    <xs:sequence>
      <xs:element name="XRange" type="tt:IntRange"/>
      <xs:element name="YRange" type="tt:IntRange"/>
      <xs:element name="WidthRange" type="tt:IntRange"/>
      <xs:element name="HeightRange" type="tt:IntRange"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="IntRange">
    <xs:sequence>
      <xs:element name="Min" type="xs:int"/>
      <xs:element name="Max" type="xs:int"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="FloatRange">
    <xs:sequence>
      <xs:element name="Min" type="xs:float"/>
      <xs:element name="Max" type="xs:float"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="DurationRange">
    <xs:sequence>
      <xs:element name="Min" type="xs:duration"/>
      <xs:element name="Max" type="xs:duration"/>
    </xs:sequence>
  </xs:complexType>
  <xs:simpleType name="IntAttrList">
    <xs:list itemType="xs:int"/>
  </xs:simpleType>
  <xs:simpleType name="FloatAttrList">
    <xs:list itemType="xs:float"/>
  </xs:simpleType>
  <xs:simpleType name="ReferenceTokenList">
    <xs:list itemType="tt:ReferenceToken"/>
  </xs:simpleType>
  <xs:complexType name="AnyHolder">
    <xs:sequence>
      <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="VideoSource">
    <xs:complexContent>
      <xs:extension base="tt:DeviceEntity">
        <xs:sequence>
          <xs:element name="Framerate" type="xs:float"/>
          <xs:element name="Resolution" type="tt:VideoResolution"/>
          <xs:element name="Imaging" type="tt:ImagingSettings" minOccurs="0"/>
          <xs:element name="Extension" type="tt:VideoSourceExtension" minOccurs="0"/>
        </xs:sequence>
        <xs:anyAttribute processContents="lax"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="VideoSourceExtension">
    <xs:sequence>
      <xs:any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

```

```

        <xs:element name="Imaging" type="tt:ImagingSettings20" minOccurs="0"/>
        <xs:element name="Extension" type="tt:VideoSourceExtension2" minOccurs="0"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="VideoSourceExtension2">
    <xs:sequence>
        <xs:any namespace="##targetNamespace" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="AudioSource">
    <xs:complexContent>
        <xs:extension base="tt:DeviceEntity">
            <xs:sequence>
                <xs:element name="Channels" type="xs:int"/>
                <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
            </xs:sequence>
            <xs:anyAttribute processContents="lax"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="Profile">
    <xs:sequence>
        <xs:element name="Name" type="tt:Name"/>
        <xs:element name="VideoSourceConfiguration" type="tt:VideoSourceConfiguration"
minOccurs="0"/>
        <xs:element name="AudioSourceConfiguration" type="tt:AudioSourceConfiguration"
minOccurs="0"/>
        <xs:element name="VideoEncoderConfiguration" type="tt:VideoEncoderConfiguration"
minOccurs="0"/>
        <xs:element name="AudioEncoderConfiguration" type="tt:AudioEncoderConfiguration"
minOccurs="0"/>
        <xs:element name="VideoAnalyticsConfiguration"
type="tt:VideoAnalyticsConfiguration" minOccurs="0"/>
        <xs:element name="PTZConfiguration" type="tt:PTZConfiguration" minOccurs="0"/>
        <xs:element name="MetadataConfiguration" type="tt:MetadataConfiguration"
minOccurs="0"/>
        <xs:element name="Extension" type="tt:ProfileExtension" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="token" type="tt:ReferenceToken" use="required"/>
    <xs:attribute name="fixed" type="xs:boolean"/>
    <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="ProfileExtension">
    <xs:sequence>
        <xs:any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
        <xs:element name="AudioOutputConfiguration" type="tt:AudioOutputConfiguration"
minOccurs="0"/>
        <xs:element name="AudioDecoderConfiguration" type="tt:AudioDecoderConfiguration"
minOccurs="0"/>
        <xs:element name="Extension" type="tt:ProfileExtension2" minOccurs="0"/>
    </xs:sequence>
    <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="ProfileExtension2">
    <xs:sequence>
        <xs:any namespace="##targetNamespace" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
    <xs:element name="VideoSourceConfiguration" type="tt:VideoSourceConfiguration"/>
    <xs:element name="AudioSourceConfiguration" type="tt:AudioSourceConfiguration"/>
    <xs:element name="VideoEncoderConfiguration" type="tt:VideoEncoderConfiguration"/>
    <xs:element name="AudioEncoderConfiguration" type="tt:AudioEncoderConfiguration"/>
    <xs:element name="VideoAnalyticsConfiguration"
type="tt:VideoAnalyticsConfiguration"/>
    <xs:element name="PTZConfiguration" type="tt:PTZConfiguration"/>
    <xs:element name="MetadataConfiguration" type="tt:MetadataConfiguration"/>
    <xs:element name="AudioOutputConfiguration" type="tt:AudioOutputConfiguration"/>
    <xs:element name="AudioDecoderConfiguration" type="tt:AudioDecoderConfiguration"/>
    <xs:complexType name="ConfigurationEntity">
        <xs:sequence>
            <xs:element name="Name" type="tt:Name"/>
            <xs:element name="UseCount" type="xs:int"/>
        </xs:sequence>
        <xs:attribute name="token" type="tt:ReferenceToken" use="required"/>

```

```

</xs:complexType>
<xs:complexType name="VideoSourceConfiguration">
  <xs:complexContent>
    <xs:extension base="tt:ConfigurationEntity">
      <xs:sequence>
        <xs:element name="SourceToken" type="tt:ReferenceToken"/>
        <xs:element name="Bounds" type="tt:IntRectangle"/>
        <xs:any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
        <xs:element name="Extension" type="tt:VideoSourceConfigurationExtension"
minOccurs="0"/>
      </xs:sequence>
      <xs:attribute name="ViewMode" type="xs:string"/>
      <xs:anyAttribute processContents="lax"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="VideoSourceConfigurationExtension">
  <xs:sequence>
    <xs:element name="Rotate" type="tt:Rotate" minOccurs="0"/>
    <xs:element name="Extension" type="tt:VideoSourceConfigurationExtension2"
minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="VideoSourceConfigurationExtension2">
  <xs:sequence>
    <xs:element name="LensDescription" type="tt:LensDescription" minOccurs="0"
maxOccurs="unbounded"/>
    <xs:element name="SceneOrientation" type="tt:SceneOrientation" minOccurs="0"
maxOccurs="1"/>
    <xs:any namespace="##targetNamespace" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="Rotate">
  <xs:sequence>
    <xs:element name="Mode" type="tt:RotateMode"/>
    <xs:element name="Degree" type="xs:int" minOccurs="0"/>
    <xs:element name="Extension" type="tt:RotateExtension" minOccurs="0"/>
  </xs:sequence>
  <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="RotateExtension">
  <xs:sequence>
    <xs:any namespace="##targetNamespace" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<xs:simpleType name="RotateMode">
  <xs:restriction base="xs:string">
    <xs:enumeration value="OFF"/>
    <xs:enumeration value="ON"/>
    <xs:enumeration value="AUTO"/>
  </xs:restriction>
</xs:simpleType>
<xs:complexType name="LensProjection">
  <xs:sequence>
    <xs:element name="Angle" type="xs:float"/>
    <xs:element name="Radius" type="xs:float"/>
    <xs:element name="Transmittance" type="xs:float" minOccurs="0"/>
    <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="LensOffset">
  <xs:attribute name="x" type="xs:float"/>
  <xs:attribute name="y" type="xs:float"/>
  <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="LensDescription">
  <xs:sequence>
    <xs:element name="Offset" type="tt:LensOffset"/>
    <xs:element name="Projection" type="tt:LensProjection" maxOccurs="unbounded"/>
    <xs:element name="XFactor" type="xs:float"/>
    <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>

```

```

    <xs:attribute name="FocalLength" type="xs:float"/>
    <xs:anyAttribute processContents="lax"/>
  </xs:complexType>
  <xs:complexType name="VideoSourceConfigurationOptions">
    <xs:sequence>
      <xs:element name="BoundsRange" type="tt:IntRectangleRange"/>
      <xs:element name="VideoSourceTokensAvailable" type="tt:ReferenceToken"
maxOccurs="unbounded"/>
      <xs:element name="Extension" type="tt:VideoSourceConfigurationOptionsExtension"
minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="MaximumNumberOfProfiles" type="xs:int"/>
    <xs:anyAttribute processContents="lax"/>
  </xs:complexType>
  <xs:complexType name="VideoSourceConfigurationOptionsExtension">
    <xs:sequence>
      <xs:any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
      <xs:element name="Rotate" type="tt:RotateOptions" minOccurs="0"/>
      <xs:element name="Extension" type="tt:VideoSourceConfigurationOptionsExtension2"
minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="VideoSourceConfigurationOptionsExtension2">
    <xs:sequence>
      <xs:element name="SceneOrientationMode" type="tt:SceneOrientationMode"
minOccurs="0" maxOccurs="unbounded"/>
      <xs:any namespace="##targetNamespace" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="RotateOptions">
    <xs:sequence>
      <xs:element name="Mode" type="tt:RotateMode" maxOccurs="unbounded"/>
      <xs:element name="DegreeList" type="tt:IntList" minOccurs="0"/>
      <xs:element name="Extension" type="tt:RotateOptionsExtension" minOccurs="0"/>
    </xs:sequence>
    <xs:anyAttribute processContents="lax"/>
  </xs:complexType>
  <xs:complexType name="RotateOptionsExtension">
    <xs:sequence>
      <xs:any namespace="##targetNamespace" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:simpleType name="SceneOrientationMode">
    <xs:restriction base="xs:string">
      <xs:enumeration value="MANUAL"/>
      <xs:enumeration value="AUTO"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="SceneOrientationOption">
    <xs:restriction base="xs:string">
      <xs:enumeration value="Below"/>
      <xs:enumeration value="Horizon"/>
      <xs:enumeration value="Above"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:complexType name="SceneOrientation">
    <xs:sequence>
      <xs:element name="Mode" type="tt:SceneOrientationMode"/>
      <xs:element name="Orientation" type="xs:string" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
  <xs:simpleType name="ViewModes">
    <xs:restriction base="xs:string">
      <xs:enumeration value="tt:Fisheye"/>
      <xs:enumeration value="tt:360Panorama"/>
      <xs:enumeration value="tt:180Panorama"/>
      <xs:enumeration value="tt:Quad"/>
      <xs:enumeration value="tt:Original"/>
      <xs:enumeration value="tt:LeftHalf"/>
      <xs:enumeration value="tt:RightHalf"/>
      <xs:enumeration value="tt:Dewarp"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:complexType name="VideoEncoderConfiguration">
    <xs:complexContent>

```



```

<xs:extension base="tt:ConfigurationEntity">
  <xs:sequence>
    <xs:element name="Encoding" type="tt:VideoEncoding"/>
    <xs:element name="Resolution" type="tt:VideoResolution"/>
    <xs:element name="Quality" type="xs:float"/>
    <xs:element name="RateControl" type="tt:VideoRateControl" minOccurs="0"/>
    <xs:element name="MPEG4" type="tt:Mpeg4Configuration" minOccurs="0"/>
    <xs:element name="H264" type="tt:H264Configuration" minOccurs="0"/>
    <xs:element name="Multicast" type="tt:MulticastConfiguration"/>
    <xs:element name="SessionTimeout" type="xs:duration"/>
    <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:anyAttribute processContents="lax"/>
</xs:extension>
</xs:complexType>
</xs:complexType>
<xs:simpleType name="VideoEncoding">
  <xs:restriction base="xs:string">
    <xs:enumeration value="JPEG"/>
    <xs:enumeration value="MPEG4"/>
    <xs:enumeration value="H264"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="Mpeg4Profile">
  <xs:restriction base="xs:string">
    <xs:enumeration value="SP"/>
    <xs:enumeration value="ASP"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="H264Profile">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Baseline"/>
    <xs:enumeration value="Main"/>
    <xs:enumeration value="Extended"/>
    <xs:enumeration value="High"/>
  </xs:restriction>
</xs:simpleType>
<xs:complexType name="VideoResolution">
  <xs:sequence>
    <xs:element name="Width" type="xs:int"/>
    <xs:element name="Height" type="xs:int"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="VideoRateControl">
  <xs:sequence>
    <xs:element name="FrameRateLimit" type="xs:int"/>
    <xs:element name="EncodingInterval" type="xs:int"/>
    <xs:element name="BitrateLimit" type="xs:int"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="Mpeg4Configuration">
  <xs:sequence>
    <xs:element name="GovLength" type="xs:int"/>
    <xs:element name="Mpeg4Profile" type="tt:Mpeg4Profile"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="H264Configuration">
  <xs:sequence>
    <xs:element name="GovLength" type="xs:int"/>
    <xs:element name="H264Profile" type="tt:H264Profile"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="VideoEncoderConfigurationOptions">
  <xs:sequence>
    <xs:element name="QualityRange" type="tt:IntRange"/>
    <xs:element name="JPEG" type="tt:JpegOptions" minOccurs="0"/>
    <xs:element name="MPEG4" type="tt:Mpeg4Options" minOccurs="0"/>
    <xs:element name="H264" type="tt:H264Options" minOccurs="0"/>
    <xs:element name="Extension" type="tt:VideoEncoderOptionsExtension"
minOccurs="0"/>
  </xs:sequence>
  <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="VideoEncoderOptionsExtension">
  <xs:sequence>
    <xs:any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

```



```

        <xs:element name="JPEG" type="tt:JpegOptions2" minOccurs="0"/>
        <xs:element name="MPEG4" type="tt:Mpeg4Options2" minOccurs="0"/>
        <xs:element name="H264" type="tt:H264Options2" minOccurs="0"/>
        <xs:element name="Extension" type="tt:VideoEncoderOptionsExtension2"
minOccurs="0"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="VideoEncoderOptionsExtension2">
    <xs:sequence>
        <xs:any namespace="##targetNamespace" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="JpegOptions">
    <xs:sequence>
        <xs:element name="ResolutionsAvailable" type="tt:VideoResolution"
maxOccurs="unbounded"/>
        <xs:element name="FrameRateRange" type="tt:IntRange"/>
        <xs:element name="EncodingIntervalRange" type="tt:IntRange"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="JpegOptions2">
    <xs:complexContent>
        <xs:extension base="tt:JpegOptions">
            <xs:sequence>
                <xs:element name="BitrateRange" type="tt:IntRange"/>
                <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
            </xs:sequence>
            <xs:anyAttribute processContents="lax"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="Mpeg4Options">
    <xs:sequence>
        <xs:element name="ResolutionsAvailable" type="tt:VideoResolution"
maxOccurs="unbounded"/>
        <xs:element name="GovLengthRange" type="tt:IntRange"/>
        <xs:element name="FrameRateRange" type="tt:IntRange"/>
        <xs:element name="EncodingIntervalRange" type="tt:IntRange"/>
        <xs:element name="Mpeg4ProfilesSupported" type="tt:Mpeg4Profile"
maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="Mpeg4Options2">
    <xs:complexContent>
        <xs:extension base="tt:Mpeg4Options">
            <xs:sequence>
                <xs:element name="BitrateRange" type="tt:IntRange"/>
                <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
            </xs:sequence>
            <xs:anyAttribute processContents="lax"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="H264Options">
    <xs:sequence>
        <xs:element name="ResolutionsAvailable" type="tt:VideoResolution"
maxOccurs="unbounded"/>
        <xs:element name="GovLengthRange" type="tt:IntRange"/>
        <xs:element name="FrameRateRange" type="tt:IntRange"/>
        <xs:element name="EncodingIntervalRange" type="tt:IntRange"/>
        <xs:element name="H264ProfilesSupported" type="tt:H264Profile"
maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="H264Options2">
    <xs:complexContent>
        <xs:extension base="tt:H264Options">
            <xs:sequence>
                <xs:element name="BitrateRange" type="tt:IntRange"/>
                <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
            </xs:sequence>
            <xs:anyAttribute processContents="lax"/>
        </xs:extension>
    </xs:complexContent>

```

```

</xs:complexType>
<xs:simpleType name="VideoEncodingMimeNames">
  <xs:restriction base="xs:string">
    <xs:enumeration value="JPEG"/>
    <xs:enumeration value="MPV4-ES"/>
    <xs:enumeration value="H264"/>
    <xs:enumeration value="H265"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="VideoEncodingProfiles">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Simple"/>
    <xs:enumeration value="AdvancedSimple"/>
    <xs:enumeration value="Baseline"/>
    <xs:enumeration value="Main"/>
    <xs:enumeration value="Main10"/>
    <xs:enumeration value="Extended"/>
    <xs:enumeration value="High"/>
  </xs:restriction>
</xs:simpleType>
<xs:complexType name="VideoEncoder2Configuration">
  <xs:complexContent>
    <xs:extension base="tt:ConfigurationEntity">
      <xs:sequence>
        <xs:element name="Encoding" type="xs:string"/>
        <xs:element name="Resolution" type="tt:VideoResolution2"/>
        <xs:element name="RateControl" type="tt:VideoRateControl2" minOccurs="0"/>
        <xs:element name="Multicast" type="tt:MulticastConfiguration" minOccurs="0"/>
        <xs:element name="Quality" type="xs:float"/>
        <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attribute name="GovLength" type="xs:int"/>
      <xs:attribute name="Profile" type="xs:string"/>
      <xs:anyAttribute processContents="lax"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="VideoResolution2">
  <xs:sequence>
    <xs:element name="Width" type="xs:int"/>
    <xs:element name="Height" type="xs:int"/>
    <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="VideoRateControl2">
  <xs:sequence>
    <xs:element name="FrameRateLimit" type="xs:float"/>
    <xs:element name="BitrateLimit" type="xs:int"/>
    <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="ConstantBitRate" type="xs:boolean"/>
  <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="VideoEncoder2ConfigurationOptions">
  <xs:sequence>
    <xs:element name="Encoding" type="xs:string"/>
    <xs:element name="QualityRange" type="tt:FloatRange"/>
    <xs:element name="ResolutionsAvailable" type="tt:VideoResolution2"
maxOccurs="unbounded"/>
    <xs:element name="BitrateRange" type="tt:IntRange"/>
    <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="GovLengthRange" type="tt:IntAttrList"/>
  <xs:attribute name="FrameRatesSupported" type="tt:FloatAttrList"/>
  <xs:attribute name="ProfilesSupported" type="tt:StringAttrList"/>
  <xs:attribute name="ConstantBitRateSupported" type="xs:boolean"/>
  <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="AudioSourceConfiguration">
  <xs:complexContent>
    <xs:extension base="tt:ConfigurationEntity">
      <xs:sequence>
        <xs:element name="SourceToken" type="tt:ReferenceToken"/>

```

```

        <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:anyAttribute processContents="lax"/>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="AudioSourceConfigurationOptions">
    <xs:sequence>
        <xs:element name="InputTokensAvailable" type="tt:ReferenceToken"
maxOccurs="unbounded"/>
        <xs:element name="Extension" type="tt:AudioSourceOptionsExtension"
minOccurs="0"/>
    </xs:sequence>
    <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="AudioSourceOptionsExtension">
    <xs:sequence>
        <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="AudioEncoderConfiguration">
    <xs:complexContent>
        <xs:extension base="tt:ConfigurationEntity">
            <xs:sequence>
                <xs:element name="Encoding" type="tt:AudioEncoding"/>
                <xs:element name="Bitrate" type="xs:int"/>
                <xs:element name="SampleRate" type="xs:int"/>
                <xs:element name="Multicast" type="tt:MulticastConfiguration"/>
                <xs:element name="SessionTimeout" type="xs:duration"/>
                <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
            </xs:sequence>
            <xs:anyAttribute processContents="lax"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:simpleType name="AudioEncoding">
    <xs:restriction base="xs:string">
        <xs:enumeration value="G711"/>
        <xs:enumeration value="G726"/>
        <xs:enumeration value="AAC"/>
    </xs:restriction>
</xs:simpleType>
<xs:complexType name="AudioEncoderConfigurationOptions">
    <xs:sequence>
        <xs:element name="Options" type="tt:AudioEncoderConfigurationOption"
minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="AudioEncoderConfigurationOption">
    <xs:sequence>
        <xs:element name="Encoding" type="tt:AudioEncoding"/>
        <xs:element name="BitrateList" type="tt:IntList"/>
        <xs:element name="SampleRateList" type="tt:IntList"/>
        <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:simpleType name="AudioEncodingMimeNames">
    <xs:restriction base="xs:string">
        <xs:enumeration value="PCMU"/>
        <xs:enumeration value="G726"/>
        <xs:enumeration value="MP4A-LATM"/>
    </xs:restriction>
</xs:simpleType>
<xs:complexType name="AudioEncoder2Configuration">
    <xs:complexContent>
        <xs:extension base="tt:ConfigurationEntity">
            <xs:sequence>
                <xs:element name="Encoding" type="xs:string"/>
                <xs:element name="Multicast" type="tt:MulticastConfiguration" minOccurs="0"/>
                <xs:element name="Bitrate" type="xs:int"/>
                <xs:element name="SampleRate" type="xs:int"/>
                <xs:any namespace="##any" processContents="lax" minOccurs="0"

```

```

maxOccurs="unbounded"/>
</xs:sequence>
<xs:anyAttribute processContents="lax"/>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="AudioEncoder2ConfigurationOptions">
  <xs:sequence>
    <xs:element name="Encoding" type="xs:string"/>
    <xs:element name="BitrateList" type="tt:IntList"/>
    <xs:element name="SampleRateList" type="tt:IntList"/>
    <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="VideoAnalyticsConfiguration">
  <xs:complexContent>
    <xs:extension base="tt:ConfigurationEntity">
      <xs:sequence>
        <xs:element name="AnalyticsEngineConfiguration"
type="tt:AnalyticsEngineConfiguration"/>
        <xs:element name="RuleEngineConfiguration"
type="tt:RuleEngineConfiguration"/>
        <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:anyAttribute processContents="lax"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="MetadataConfiguration">
  <xs:complexContent>
    <xs:extension base="tt:ConfigurationEntity">
      <xs:sequence>
        <xs:element name="PTZStatus" type="tt:PTZFilter" minOccurs="0"/>
        <xs:element name="Events" type="tt:EventSubscription" minOccurs="0"/>
        <xs:element name="Analytics" type="xs:boolean" minOccurs="0"/>
        <xs:element name="Multicast" type="tt:MulticastConfiguration"/>
        <xs:element name="SessionTimeout" type="xs:duration"/>
        <xs:any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:element name="AnalyticsEngineConfiguration"
type="tt:AnalyticsEngineConfiguration" minOccurs="0"/>
      <xs:element name="Extension" type="tt:MetadataConfigurationExtension"
minOccurs="0"/>
    </xs:extension>
  </xs:complexContent>
  <xs:attribute name="CompressionType" type="xs:string"/>
  <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="MetadataConfigurationExtension">
  <xs:sequence>
    <xs:any namespace="##targetNamespace" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="PTZFilter">
  <xs:sequence>
    <xs:element name="Status" type="xs:boolean"/>
    <xs:element name="Position" type="xs:boolean"/>
  </xs:sequence>
  <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="EventSubscription">
  <xs:sequence>
    <xs:element name="Filter" type="wsnt:FilterType" minOccurs="0"/>
    <xs:element name="SubscriptionPolicy" minOccurs="0">
      <xs:complexType>
        <xs:sequence>
          <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

```

```

    </xs:sequence>
    <xs:anyAttribute processContents="lax"/>
  </xs:complexType>
  <xs:complexType name="MetadataConfigurationOptions">
    <xs:sequence>
      <xs:element name="PTZStatusFilterOptions" type="tt:PTZStatusFilterOptions"/>
      <xs:any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
      <xs:element name="Extension" type="tt:MetadataConfigurationOptionsExtension"
minOccurs="0"/>
    </xs:sequence>
    <xs:anyAttribute processContents="lax"/>
  </xs:complexType>
  <xs:complexType name="MetadataConfigurationOptionsExtension">
    <xs:sequence>
      <xs:element name="CompressionType" type="xs:string" minOccurs="0"
maxOccurs="unbounded"/>
      <xs:element name="Extension" type="tt:MetadataConfigurationOptionsExtension2"
minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="MetadataConfigurationOptionsExtension2">
    <xs:sequence>
      <xs:any namespace="##targetNamespace" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:simpleType name="MetadataCompressionType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="None"/>
      <xs:enumeration value="GZIP"/>
      <xs:enumeration value="EXI"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:complexType name="PTZStatusFilterOptions">
    <xs:sequence>
      <xs:element name="PanTiltStatusSupported" type="xs:boolean"/>
      <xs:element name="ZoomStatusSupported" type="xs:boolean"/>
      <xs:any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
      <xs:element name="PanTiltPositionSupported" type="xs:boolean" minOccurs="0"/>
      <xs:element name="ZoomPositionSupported" type="xs:boolean" minOccurs="0"/>
      <xs:element name="Extension" type="tt:PTZStatusFilterOptionsExtension"
minOccurs="0"/>
    </xs:sequence>
    <xs:anyAttribute processContents="lax"/>
  </xs:complexType>
  <xs:complexType name="PTZStatusFilterOptionsExtension">
    <xs:sequence>
      <xs:any namespace="##targetNamespace" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="VideoOutput">
    <xs:complexContent>
      <xs:extension base="tt:DeviceEntity">
        <xs:sequence>
          <xs:element name="Layout" type="tt:Layout"/>
          <xs:element name="Resolution" type="tt:VideoResolution" minOccurs="0"/>
          <xs:element name="RefreshRate" type="xs:float" minOccurs="0"/>
          <xs:element name="AspectRatio" type="xs:float" minOccurs="0"/>
          <xs:element name="Extension" type="tt:VideoOutputExtension" minOccurs="0"/>
        </xs:sequence>
        <xs:anyAttribute processContents="lax"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="VideoOutputExtension">
    <xs:sequence>
      <xs:any namespace="##targetNamespace" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="VideoOutputConfiguration">
    <xs:complexContent>
      <xs:extension base="tt:ConfigurationEntity">
        <xs:sequence>
          <xs:element name="OutputToken" type="tt:ReferenceToken"/>

```

```

        <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:anyAttribute processContents="lax"/>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="VideoOutputConfigurationOptions">
    <xs:sequence>
        <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="VideoDecoderConfigurationOptions">
    <xs:sequence>
        <xs:element name="JpegDecOptions" type="tt:JpegDecOptions" minOccurs="0"/>
        <xs:element name="H264DecOptions" type="tt:H264DecOptions" minOccurs="0"/>
        <xs:element name="Mpeg4DecOptions" type="tt:Mpeg4DecOptions" minOccurs="0"/>
        <xs:element name="Extension" type="tt:VideoDecoderConfigurationOptionsExtension"
minOccurs="0"/>
    </xs:sequence>
    <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="H264DecOptions">
    <xs:sequence>
        <xs:element name="ResolutionsAvailable" type="tt:VideoResolution"
maxOccurs="unbounded"/>
        <xs:element name="SupportedH264Profiles" type="tt:H264Profile"
maxOccurs="unbounded"/>
        <xs:element name="SupportedInputBitrate" type="tt:IntRange"/>
        <xs:element name="SupportedFrameRate" type="tt:IntRange"/>
        <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="JpegDecOptions">
    <xs:sequence>
        <xs:element name="ResolutionsAvailable" type="tt:VideoResolution"
maxOccurs="unbounded"/>
        <xs:element name="SupportedInputBitrate" type="tt:IntRange"/>
        <xs:element name="SupportedFrameRate" type="tt:IntRange"/>
        <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="Mpeg4DecOptions">
    <xs:sequence>
        <xs:element name="ResolutionsAvailable" type="tt:VideoResolution"
maxOccurs="unbounded"/>
        <xs:element name="SupportedMpeg4Profiles" type="tt:Mpeg4Profile"
maxOccurs="unbounded"/>
        <xs:element name="SupportedInputBitrate" type="tt:IntRange"/>
        <xs:element name="SupportedFrameRate" type="tt:IntRange"/>
        <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="VideoDecoderConfigurationOptionsExtension">
    <xs:sequence>
        <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="AudioOutput">
    <xs:complexContent>
        <xs:extension base="tt:DeviceEntity">
            <xs:sequence>
                <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
            </xs:sequence>
            <xs:anyAttribute processContents="lax"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>

```

```

<xs:complexType name="AudioOutputConfiguration">
  <xs:complexContent>
    <xs:extension base="tt:ConfigurationEntity">
      <xs:sequence>
        <xs:element name="OutputToken" type="tt:ReferenceToken"/>
        <xs:element name="SendPrimacy" type="xs:anyURI" minOccurs="0"/>
        <xs:element name="OutputLevel" type="xs:int"/>
        <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="AudioOutputConfigurationOptions">
  <xs:sequence>
    <xs:element name="OutputTokensAvailable" type="tt:ReferenceToken"
maxOccurs="unbounded"/>
    <xs:element name="SendPrimacyOptions" type="xs:anyURI" minOccurs="0"
maxOccurs="unbounded"/>
    <xs:element name="OutputLevelRange" type="tt:IntRange"/>
    <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="AudioDecoderConfiguration">
  <xs:complexContent>
    <xs:extension base="tt:ConfigurationEntity">
      <xs:sequence>
        <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="AudioDecoderConfigurationOptions">
  <xs:sequence>
    <xs:element name="AACDecOptions" type="tt:AACDecOptions" minOccurs="0"/>
    <xs:element name="G711DecOptions" type="tt:G711DecOptions" minOccurs="0"/>
    <xs:element name="G726DecOptions" type="tt:G726DecOptions" minOccurs="0"/>
    <xs:element name="Extension" type="tt:AudioDecoderConfigurationOptionsExtension"
minOccurs="0"/>
  </xs:sequence>
  <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="G711DecOptions">
  <xs:sequence>
    <xs:element name="Bitrate" type="tt:IntList"/>
    <xs:element name="SampleRateRange" type="tt:IntList"/>
    <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="AACDecOptions">
  <xs:sequence>
    <xs:element name="Bitrate" type="tt:IntList"/>
    <xs:element name="SampleRateRange" type="tt:IntList"/>
    <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="G726DecOptions">
  <xs:sequence>
    <xs:element name="Bitrate" type="tt:IntList"/>
    <xs:element name="SampleRateRange" type="tt:IntList"/>
    <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="AudioDecoderConfigurationOptionsExtension">
  <xs:sequence>
    <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>

```



```

    </xs:sequence>
</xs:complexType>
<xs:complexType name="MulticastConfiguration">
    <xs:sequence>
        <xs:element name="Address" type="tt:IPAddress"/>
        <xs:element name="Port" type="xs:int"/>
        <xs:element name="TTL" type="xs:int"/>
        <xs:element name="AutoStart" type="xs:boolean"/>
        <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="StreamSetup">
    <xs:sequence>
        <xs:element name="Stream" type="tt:StreamType"/>
        <xs:element name="Transport" type="tt:Transport"/>
        <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:simpleType name="StreamType">
    <xs:restriction base="xs:string">
        <xs:enumeration value="RTP-Unicast"/>
        <xs:enumeration value="RTP-Multicast"/>
    </xs:restriction>
</xs:simpleType>
<xs:complexType name="Transport">
    <xs:sequence>
        <xs:element name="Protocol" type="tt:TransportProtocol"/>
        <xs:element name="Tunnel" type="tt:Transport" minOccurs="0"/>
    </xs:sequence>
</xs:complexType>
<xs:simpleType name="TransportProtocol">
    <xs:restriction base="xs:string">
        <xs:enumeration value="UDP"/>
        <xs:enumeration value="TCP"/>
        <xs:enumeration value="RTSP"/>
        <xs:enumeration value="HTTP"/>
    </xs:restriction>
</xs:simpleType>
<xs:complexType name="MediaUri">
    <xs:sequence>
        <xs:element name="Uri" type="xs:anyURI"/>
        <xs:element name="InvalidAfterConnect" type="xs:boolean"/>
        <xs:element name="InvalidAfterReboot" type="xs:boolean"/>
        <xs:element name="Timeout" type="xs:duration"/>
        <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="GeoLocation">
    <xs:sequence>
        <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="lon" type="xs:double"/>
    <xs:attribute name="lat" type="xs:double"/>
    <xs:attribute name="elevation" type="xs:float"/>
    <xs:anyAttribute/>
</xs:complexType>
<xs:complexType name="GeoOrientation">
    <xs:sequence>
        <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="roll" type="xs:float"/>
    <xs:attribute name="pitch" type="xs:float"/>
    <xs:attribute name="yaw" type="xs:float"/>
    <xs:anyAttribute/>
</xs:complexType>
<xs:complexType name="LocalLocation">
    <xs:sequence>
        <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>

```



```

    <xs:attribute name="x" type="xs:float"/>
    <xs:attribute name="y" type="xs:float"/>
    <xs:attribute name="z" type="xs:float"/>
    <xs:anyAttribute/>
  </xs:complexType>
  <xs:complexType name="LocalOrientation">
    <xs:sequence>
      <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="pan" type="xs:float"/>
    <xs:attribute name="tilt" type="xs:float"/>
    <xs:attribute name="roll" type="xs:float"/>
    <xs:anyAttribute/>
  </xs:complexType>
  <xs:simpleType name="Entity">
    <xs:restriction base="xs:string">
      <xs:enumeration value="Device"/>
      <xs:enumeration value="VideoSource"/>
      <xs:enumeration value="AudioSource"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:complexType name="LocationEntity">
    <xs:sequence>
      <xs:element name="GeoLocation" type="tt:GeoLocation" minOccurs="0"/>
      <xs:element name="GeoOrientation" type="tt:GeoOrientation" minOccurs="0"/>
      <xs:element name="LocalLocation" type="tt:LocalLocation" minOccurs="0"/>
      <xs:element name="LocalOrientation" type="tt:LocalOrientation" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="Entity" type="xs:string"/>
    <xs:attribute name="Token" type="tt:ReferenceToken"/>
    <xs:attribute name="Fixed" type="xs:boolean"/>
    <xs:attribute name="GeoSource" type="xs:anyURI"/>
    <xs:attribute name="AutoGeo" type="xs:boolean"/>
  </xs:complexType>
  <xs:complexType name="PTZNode">
    <xs:complexContent>
      <xs:extension base="tt:DeviceEntity">
        <xs:sequence>
          <xs:element name="Name" type="tt:Name" minOccurs="0"/>
          <xs:element name="SupportedPTZSpaces" type="tt:PTZSpaces"/>
          <xs:element name="MaximumNumberOfPresets" type="xs:int"/>
          <xs:element name="HomeSupported" type="xs:boolean"/>
          <xs:element name="AuxiliaryCommands" type="tt:AuxiliaryData" minOccurs="0"
maxOccurs="unbounded"/>
          <xs:element name="Extension" type="tt:PTZNodeExtension" minOccurs="0"/>
        </xs:sequence>
        <xs:attribute name="FixedHomePosition" type="xs:boolean"/>
        <xs:attribute name="GeoMove" type="xs:boolean"/>
        <xs:anyAttribute processContents="lax"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="PTZNodeExtension">
    <xs:sequence>
      <xs:any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
      <xs:element name="SupportedPresetTour" type="tt:PTZPresetTourSupported"
minOccurs="0"/>
      <xs:element name="Extension" type="tt:PTZNodeExtension2" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="PTZNodeExtension2">
    <xs:sequence>
      <xs:any namespace="##targetNamespace" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="PTZPresetTourSupported">
    <xs:sequence>
      <xs:element name="MaximumNumberOfPresetTours" type="xs:int"/>
      <xs:element name="PTZPresetTourOperation" type="tt:PTZPresetTourOperation"
minOccurs="0" maxOccurs="unbounded"/>
      <xs:element name="Extension" type="tt:PTZPresetTourSupportedExtension"
minOccurs="0"/>
    </xs:sequence>
    <xs:anyAttribute processContents="lax"/>
  </xs:complexType>

```

```

<xs:complexType name="PTZPresetTourSupportedExtension">
  <xs:sequence>
    <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="PTZConfiguration">
  <xs:complexContent>
    <xs:extension base="tt:ConfigurationEntity">
      <xs:sequence>
        <xs:element name="NodeToken" type="tt:ReferenceToken"/>
        <xs:element name="DefaultAbsolutePanTiltPositionSpace" type="xs:anyURI"
minOccurs="0"/>
        <xs:element name="DefaultAbsoluteZoomPositionSpace" type="xs:anyURI"
minOccurs="0"/>
        <xs:element name="DefaultRelativePanTiltTranslationSpace" type="xs:anyURI"
minOccurs="0"/>
        <xs:element name="DefaultRelativeZoomTranslationSpace" type="xs:anyURI"
minOccurs="0"/>
        <xs:element name="DefaultContinuousPanTiltVelocitySpace" type="xs:anyURI"
minOccurs="0"/>
        <xs:element name="DefaultContinuousZoomVelocitySpace" type="xs:anyURI"
minOccurs="0"/>
        <xs:element name="DefaultPTZSpeed" type="tt:PTZSpeed" minOccurs="0"/>
        <xs:element name="DefaultPTZTimeout" type="xs:duration" minOccurs="0"/>
        <xs:element name="PanTiltLimits" type="tt:PanTiltLimits" minOccurs="0"/>
        <xs:element name="ZoomLimits" type="tt:ZoomLimits" minOccurs="0"/>
        <xs:element name="Extension" type="tt:PTZConfigurationExtension"
minOccurs="0"/>
      </xs:sequence>
      <xs:attribute name="MoveRamp" type="xs:int"/>
      <xs:attribute name="PresetRamp" type="xs:int"/>
      <xs:attribute name="PresetTourRamp" type="xs:int"/>
      <xs:anyAttribute processContents="lax"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="PTZConfigurationExtension">
  <xs:sequence>
    <xs:any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    <xs:element name="PTControlDirection" type="tt:PTControlDirection"
minOccurs="0"/>
    <xs:element name="Extension" type="tt:PTZConfigurationExtension2" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="PTZConfigurationExtension2">
  <xs:sequence>
    <xs:any namespace="##targetNamespace" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="PTControlDirection">
  <xs:sequence>
    <xs:element name="EFlip" type="tt:EFlip" minOccurs="0"/>
    <xs:element name="Reverse" type="tt:Reverse" minOccurs="0"/>
    <xs:element name="Extension" type="tt:PTControlDirectionExtension"
minOccurs="0"/>
  </xs:sequence>
  <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="PTControlDirectionExtension">
  <xs:sequence>
    <xs:any namespace="##targetNamespace" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="EFlip">
  <xs:sequence>
    <xs:element name="Mode" type="tt:EFlipMode"/>
    <xs:any namespace="##targetNamespace" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="Reverse">
  <xs:sequence>

```

```

        <xs:element name="Mode" type="tt:ReverseMode"/>
        <xs:any namespace="##targetNamespace" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:simpleType name="EFlipMode">
    <xs:restriction base="xs:string">
        <xs:enumeration value="OFF"/>
        <xs:enumeration value="ON"/>
        <xs:enumeration value="Extended"/>
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="ReverseMode">
    <xs:restriction base="xs:string">
        <xs:enumeration value="OFF"/>
        <xs:enumeration value="ON"/>
        <xs:enumeration value="AUTO"/>
        <xs:enumeration value="Extended"/>
    </xs:restriction>
</xs:simpleType>
<xs:complexType name="PTZConfigurationOptions">
    <xs:sequence>
        <xs:element name="Spaces" type="tt:PTZSpaces"/>
        <xs:element name="PTZTimeout" type="tt:DurationRange"/>
        <xs:any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
        <xs:element name="PTControlDirection" type="tt:PTControlDirectionOptions"
minOccurs="0"/>
        <xs:element name="Extension" type="tt:PTZConfigurationOptions2" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="PTZRamps" type="tt:IntAttrList"/>
    <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="PTZConfigurationOptions2">
    <xs:sequence>
        <xs:any namespace="##targetNamespace" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="PTControlDirectionOptions">
    <xs:sequence>
        <xs:element name="EFlip" type="tt:EFlipOptions" minOccurs="0"/>
        <xs:element name="Reverse" type="tt:ReverseOptions" minOccurs="0"/>
        <xs:element name="Extension" type="tt:PTControlDirectionOptionsExtension"
minOccurs="0"/>
    </xs:sequence>
    <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="PTControlDirectionOptionsExtension">
    <xs:sequence>
        <xs:any namespace="##targetNamespace" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="EFlipOptions">
    <xs:sequence>
        <xs:element name="Mode" type="tt:EFlipMode" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="Extension" type="tt:EFlipOptionsExtension" minOccurs="0"/>
    </xs:sequence>
    <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="EFlipOptionsExtension">
    <xs:sequence>
        <xs:any namespace="##targetNamespace" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="ReverseOptions">
    <xs:sequence>
        <xs:element name="Mode" type="tt:ReverseMode" minOccurs="0"
maxOccurs="unbounded"/>
        <xs:element name="Extension" type="tt:ReverseOptionsExtension" minOccurs="0"/>
    </xs:sequence>
    <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="ReverseOptionsExtension">
    <xs:sequence>

```

```

        <xs:any namespace="##targetNamespace" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="PanTiltLimits">
    <xs:sequence>
        <xs:element name="Range" type="tt:Space2DDescription"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="ZoomLimits">
    <xs:sequence>
        <xs:element name="Range" type="tt:Space1DDescription"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="PTZSpaces">
    <xs:sequence>
        <xs:element name="AbsolutePanTiltPositionSpace" type="tt:Space2DDescription"
minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="AbsoluteZoomPositionSpace" type="tt:Space1DDescription"
minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="RelativePanTiltTranslationSpace" type="tt:Space2DDescription"
minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="RelativeZoomTranslationSpace" type="tt:Space1DDescription"
minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="ContinuousPanTiltVelocitySpace" type="tt:Space2DDescription"
minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="ContinuousZoomVelocitySpace" type="tt:Space1DDescription"
minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="PanTiltSpeedSpace" type="tt:Space1DDescription" minOccurs="0"
maxOccurs="unbounded"/>
        <xs:element name="ZoomSpeedSpace" type="tt:Space1DDescription" minOccurs="0"
maxOccurs="unbounded"/>
        <xs:element name="Extension" type="tt:PTZSpacesExtension" minOccurs="0"/>
    </xs:sequence>
    <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="PTZSpacesExtension">
    <xs:sequence>
        <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="Space2DDescription">
    <xs:sequence>
        <xs:element name="URI" type="xs:anyURI"/>
        <xs:element name="XRange" type="tt:FloatRange"/>
        <xs:element name="YRange" type="tt:FloatRange"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="Space1DDescription">
    <xs:sequence>
        <xs:element name="URI" type="xs:anyURI"/>
        <xs:element name="XRange" type="tt:FloatRange"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="PTZSpeed">
    <xs:sequence>
        <xs:element name="PanTilt" type="tt:Vector2D" minOccurs="0"/>
        <xs:element name="Zoom" type="tt:Vector1D" minOccurs="0"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="PTZPreset">
    <xs:sequence>
        <xs:element name="Name" type="tt:Name" minOccurs="0"/>
        <xs:element name="PTZPosition" type="tt:PTZVector" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="token" type="tt:ReferenceToken"/>
    <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:simpleType name="PTZPresetTourState">
    <xs:restriction base="xs:string">
        <xs:enumeration value="Idle"/>
        <xs:enumeration value="Paused"/>
        <xs:enumeration value="Extended"/>
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="PTZPresetTourDirection">
    <xs:restriction base="xs:string">

```

```

        <xs:enumeration value="Forward"/>
        <xs:enumeration value="Backward"/>
        <xs:enumeration value="Extended"/>
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="PTZPresetTourOperation">
    <xs:restriction base="xs:string">
        <xs:enumeration value="Start"/>
        <xs:enumeration value="Stop"/>
        <xs:enumeration value="Pause"/>
        <xs:enumeration value="Extended"/>
    </xs:restriction>
</xs:simpleType>
<xs:complexType name="PresetTour">
    <xs:sequence>
        <xs:element name="Name" type="tt:Name" minOccurs="0"/>
        <xs:element name="Status" type="tt:PTZPresetTourStatus"/>
        <xs:element name="AutoStart" type="xs:boolean"/>
        <xs:element name="StartingCondition" type="tt:PTZPresetTourStartingCondition"/>
        <xs:element name="TourSpot" type="tt:PTZPresetTourSpot" minOccurs="0"
maxOccurs="unbounded"/>
        <xs:element name="Extension" type="tt:PTZPresetTourExtension" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="token" type="tt:ReferenceToken"/>
    <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="PTZPresetTourExtension">
    <xs:sequence>
        <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="PTZPresetTourSpot">
    <xs:sequence>
        <xs:element name="PresetDetail" type="tt:PTZPresetTourPresetDetail"/>
        <xs:element name="Speed" type="tt:PTZSpeed" minOccurs="0"/>
        <xs:element name="StayTime" type="xs:duration" minOccurs="0"/>
        <xs:element name="Extension" type="tt:PTZPresetTourSpotExtension" minOccurs="0"/>
    </xs:sequence>
    <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="PTZPresetTourSpotExtension">
    <xs:sequence>
        <xs:any namespace="##targetNamespace" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="PTZPresetTourPresetDetail">
    <xs:sequence>
        <xs:choice>
            <xs:element name="PresetToken" type="tt:ReferenceToken"/>
            <xs:element name="Home" type="xs:boolean"/>
            <xs:element name="PTZPosition" type="tt:PTZVector"/>
            <xs:element name="TypeExtension" type="tt:PTZPresetTourTypeExtension"/>
        </xs:choice>
        <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="PTZPresetTourTypeExtension">
    <xs:sequence>
        <xs:any namespace="##targetNamespace" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="PTZPresetTourStatus">
    <xs:sequence>
        <xs:element name="State" type="tt:PTZPresetTourState"/>
        <xs:element name="CurrentTourSpot" type="tt:PTZPresetTourSpot" minOccurs="0"/>
        <xs:element name="Extension" type="tt:PTZPresetTourStatusExtension"
minOccurs="0"/>
    </xs:sequence>
    <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="PTZPresetTourStatusExtension">
    <xs:sequence>
        <xs:any namespace="##any" processContents="lax" minOccurs="0"

```

```

maxOccurs="unbounded"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="PTZPresetTourStartingCondition">
  <xs:sequence>
    <xs:element name="RecurringTime" type="xs:int" minOccurs="0"/>
    <xs:element name="RecurringDuration" type="xs:duration" minOccurs="0"/>
    <xs:element name="Direction" type="tt:PTZPresetTourDirection" minOccurs="0"/>
    <xs:element name="Extension" type="tt:PTZPresetTourStartingConditionExtension"
minOccurs="0"/>
  </xs:sequence>
  <xs:attribute name="RandomPresetOrder" type="xs:boolean"/>
  <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="PTZPresetTourStartingConditionExtension">
  <xs:sequence>
    <xs:any namespace="##targetNamespace" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="PTZPresetTourOptions">
  <xs:sequence>
    <xs:element name="AutoStart" type="xs:boolean"/>
    <xs:element name="StartingCondition"
type="tt:PTZPresetTourStartingConditionOptions"/>
    <xs:element name="TourSpot" type="tt:PTZPresetTourSpotOptions"/>
    <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="PTZPresetTourSpotOptions">
  <xs:sequence>
    <xs:element name="PresetDetail" type="tt:PTZPresetTourPresetDetailOptions"/>
    <xs:element name="StayTime" type="tt:DurationRange"/>
    <xs:any namespace="##targetNamespace" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="PTZPresetTourPresetDetailOptions">
  <xs:sequence>
    <xs:element name="PresetToken" type="tt:ReferenceToken" minOccurs="0"
maxOccurs="unbounded"/>
    <xs:element name="Home" type="xs:boolean" minOccurs="0"/>
    <xs:element name="PanTiltPositionSpace" type="tt:Space2DDescription"
minOccurs="0"/>
    <xs:element name="ZoomPositionSpace" type="tt:Space1DDescription" minOccurs="0"/>
    <xs:element name="Extension" type="tt:PTZPresetTourPresetDetailOptionsExtension"
minOccurs="0"/>
  </xs:sequence>
  <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="PTZPresetTourPresetDetailOptionsExtension">
  <xs:sequence>
    <xs:any namespace="##targetNamespace" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="PTZPresetTourStartingConditionOptions">
  <xs:sequence>
    <xs:element name="RecurringTime" type="tt:IntRange" minOccurs="0"/>
    <xs:element name="RecurringDuration" type="tt:DurationRange" minOccurs="0"/>
    <xs:element name="Direction" type="tt:PTZPresetTourDirection" minOccurs="0"
maxOccurs="unbounded"/>
    <xs:element name="Extension"
type="tt:PTZPresetTourStartingConditionOptionsExtension" minOccurs="0"/>
  </xs:sequence>
  <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="PTZPresetTourStartingConditionOptionsExtension">
  <xs:sequence>
    <xs:any namespace="##targetNamespace" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="ImagingStatus">
  <xs:sequence>

```

```
<xs:element name="FocusStatus" type="tt:FocusStatus"/>
<xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
</xs:sequence>
<xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="FocusStatus">
<xs:sequence>
<xs:element name="Position" type="xs:float"/>
<xs:element name="MoveStatus" type="tt:MoveStatus"/>
<xs:element name="Error" type="xs:string"/>
<xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
</xs:sequence>
<xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="FocusConfiguration">
<xs:sequence>
<xs:element name="AutoFocusMode" type="tt:AutoFocusMode"/>
<xs:element name="DefaultSpeed" type="xs:float"/>
<xs:element name="NearLimit" type="xs:float"/>
<xs:element name="FarLimit" type="xs:float"/>
<xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
</xs:sequence>
<xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:simpleType name="AutoFocusMode">
<xs:restriction base="xs:string">
<xs:enumeration value="AUTO"/>
<xs:enumeration value="MANUAL"/>
</xs:restriction>
</xs:simpleType>
<xs:complexType name="ImagingSettings">
<xs:sequence>
<xs:element name="BacklightCompensation" type="tt:BacklightCompensation"
minOccurs="0"/>
<xs:element name="Brightness" type="xs:float" minOccurs="0"/>
<xs:element name="ColorSaturation" type="xs:float" minOccurs="0"/>
<xs:element name="Contrast" type="xs:float" minOccurs="0"/>
<xs:element name="Exposure" type="tt:Exposure" minOccurs="0"/>
<xs:element name="Focus" type="tt:FocusConfiguration" minOccurs="0"/>
<xs:element name="IrCutFilter" type="tt:IrCutFilterMode" minOccurs="0"/>
<xs:element name="Sharpness" type="xs:float" minOccurs="0"/>
<xs:element name="WideDynamicRange" type="tt:WideDynamicRange" minOccurs="0"/>
<xs:element name="WhiteBalance" type="tt:WhiteBalance" minOccurs="0"/>
<xs:element name="Extension" type="tt:ImagingSettingsExtension" minOccurs="0"/>
</xs:sequence>
<xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="ImagingSettingsExtension">
<xs:sequence>
<xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="Exposure">
<xs:sequence>
<xs:element name="Mode" type="tt:ExposureMode"/>
<xs:element name="Priority" type="tt:ExposurePriority"/>
<xs:element name="Window" type="tt:Rectangle"/>
<xs:element name="MinExposureTime" type="xs:float"/>
<xs:element name="MaxExposureTime" type="xs:float"/>
<xs:element name="MinGain" type="xs:float"/>
<xs:element name="MaxGain" type="xs:float"/>
<xs:element name="MinIris" type="xs:float"/>
<xs:element name="MaxIris" type="xs:float"/>
<xs:element name="ExposureTime" type="xs:float"/>
<xs:element name="Gain" type="xs:float"/>
<xs:element name="Iris" type="xs:float"/>
</xs:sequence>
</xs:complexType>
<xs:simpleType name="WideDynamicMode">
<xs:restriction base="xs:string">
<xs:enumeration value="OFF"/>
<xs:enumeration value="ON"/>
</xs:restriction>
</xs:simpleType>
```



```

<xs:complexType name="WideDynamicRange">
  <xs:sequence>
    <xs:element name="Mode" type="tt:WideDynamicMode"/>
    <xs:element name="Level" type="xs:float"/>
  </xs:sequence>
</xs:complexType>
<xs:simpleType name="BacklightCompensationMode">
  <xs:restriction base="xs:string">
    <xs:enumeration value="OFF"/>
    <xs:enumeration value="ON"/>
  </xs:restriction>
</xs:simpleType>
<xs:complexType name="BacklightCompensation">
  <xs:sequence>
    <xs:element name="Mode" type="tt:BacklightCompensationMode"/>
    <xs:element name="Level" type="xs:float"/>
  </xs:sequence>
</xs:complexType>
<xs:simpleType name="ExposurePriority">
  <xs:restriction base="xs:string">
    <xs:enumeration value="LowNoise"/>
    <xs:enumeration value="FrameRate"/>
  </xs:restriction>
</xs:simpleType>
<xs:complexType name="ImagingOptions">
  <xs:sequence>
    <xs:element name="BacklightCompensation" type="tt:BacklightCompensationOptions"/>
    <xs:element name="Brightness" type="tt:FloatRange"/>
    <xs:element name="ColorSaturation" type="tt:FloatRange"/>
    <xs:element name="Contrast" type="tt:FloatRange"/>
    <xs:element name="Exposure" type="tt:ExposureOptions"/>
    <xs:element name="Focus" type="tt:FocusOptions"/>
    <xs:element name="IrCutFilterModes" type="tt:IrCutFilterMode"
maxOccurs="unbounded"/>
    <xs:element name="Sharpness" type="tt:FloatRange"/>
    <xs:element name="WideDynamicRange" type="tt:WideDynamicRangeOptions"/>
    <xs:element name="WhiteBalance" type="tt:WhiteBalanceOptions"/>
    <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="WideDynamicRangeOptions">
  <xs:sequence>
    <xs:element name="Mode" type="tt:WideDynamicMode" maxOccurs="unbounded"/>
    <xs:element name="Level" type="tt:FloatRange"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="BacklightCompensationOptions">
  <xs:sequence>
    <xs:element name="Mode" type="tt:WideDynamicMode" maxOccurs="unbounded"/>
    <xs:element name="Level" type="tt:FloatRange"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="FocusOptions">
  <xs:sequence>
    <xs:element name="AutoFocusModes" type="tt:AutoFocusMode" minOccurs="0"
maxOccurs="unbounded"/>
    <xs:element name="DefaultSpeed" type="tt:FloatRange"/>
    <xs:element name="NearLimit" type="tt:FloatRange"/>
    <xs:element name="FarLimit" type="tt:FloatRange"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="ExposureOptions">
  <xs:sequence>
    <xs:element name="Mode" type="tt:ExposureMode" maxOccurs="unbounded"/>
    <xs:element name="Priority" type="tt:ExposurePriority" maxOccurs="unbounded"/>
    <xs:element name="MinExposureTime" type="tt:FloatRange"/>
    <xs:element name="MaxExposureTime" type="tt:FloatRange"/>
    <xs:element name="MinGain" type="tt:FloatRange"/>
    <xs:element name="MaxGain" type="tt:FloatRange"/>
    <xs:element name="MinIris" type="tt:FloatRange"/>
    <xs:element name="MaxIris" type="tt:FloatRange"/>
    <xs:element name="ExposureTime" type="tt:FloatRange"/>
    <xs:element name="Gain" type="tt:FloatRange"/>
    <xs:element name="Iris" type="tt:FloatRange"/>
  </xs:sequence>
</xs:complexType>

```



```
<xs:complexType name="WhiteBalanceOptions">
  <xs:sequence>
    <xs:element name="Mode" type="tt:WhiteBalanceMode" maxOccurs="unbounded"/>
    <xs:element name="YrGain" type="tt:FloatRange"/>
    <xs:element name="YbGain" type="tt:FloatRange"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="FocusMove">
  <xs:sequence>
    <xs:element name="Absolute" type="tt:AbsoluteFocus" minOccurs="0"/>
    <xs:element name="Relative" type="tt:RelativeFocus" minOccurs="0"/>
    <xs:element name="Continuous" type="tt:ContinuousFocus" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="AbsoluteFocus">
  <xs:sequence>
    <xs:element name="Position" type="xs:float"/>
    <xs:element name="Speed" type="xs:float" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="RelativeFocus">
  <xs:sequence>
    <xs:element name="Distance" type="xs:float"/>
    <xs:element name="Speed" type="xs:float" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="ContinuousFocus">
  <xs:sequence>
    <xs:element name="Speed" type="xs:float"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="MoveOptions">
  <xs:sequence>
    <xs:element name="Absolute" type="tt:AbsoluteFocusOptions" minOccurs="0"/>
    <xs:element name="Relative" type="tt:RelativeFocusOptions" minOccurs="0"/>
    <xs:element name="Continuous" type="tt:ContinuousFocusOptions" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="AbsoluteFocusOptions">
  <xs:sequence>
    <xs:element name="Position" type="tt:FloatRange"/>
    <xs:element name="Speed" type="tt:FloatRange" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="RelativeFocusOptions">
  <xs:sequence>
    <xs:element name="Distance" type="tt:FloatRange"/>
    <xs:element name="Speed" type="tt:FloatRange"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="ContinuousFocusOptions">
  <xs:sequence>
    <xs:element name="Speed" type="tt:FloatRange"/>
  </xs:sequence>
</xs:complexType>
<xs:simpleType name="ExposureMode">
  <xs:restriction base="xs:string">
    <xs:enumeration value="AUTO"/>
    <xs:enumeration value="MANUAL"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="Enabled">
  <xs:restriction base="xs:string">
    <xs:enumeration value="ENABLED"/>
    <xs:enumeration value="DISABLED"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="WhiteBalanceMode">
  <xs:restriction base="xs:string">
    <xs:enumeration value="AUTO"/>
    <xs:enumeration value="MANUAL"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="IrCutFilterMode">
  <xs:restriction base="xs:string">
    <xs:enumeration value="ON"/>
    <xs:enumeration value="OFF"/>
    <xs:enumeration value="AUTO"/>
  </xs:restriction>
</xs:simpleType>
```

```

    </xs:restriction>
  </xs:simpleType>
  <xs:complexType name="WhiteBalance">
    <xs:sequence>
      <xs:element name="Mode" type="tt:WhiteBalanceMode"/>
      <xs:element name="CrGain" type="xs:float"/>
      <xs:element name="CbGain" type="xs:float"/>
      <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:anyAttribute processContents="lax"/>
  </xs:complexType>
  <xs:complexType name="ImagingStatus20">
    <xs:sequence>
      <xs:element name="FocusStatus20" type="tt:FocusStatus20" minOccurs="0"/>
      <xs:element name="Extension" type="tt:ImagingStatus20Extension" minOccurs="0"/>
    </xs:sequence>
    <xs:anyAttribute processContents="lax"/>
  </xs:complexType>
  <xs:complexType name="ImagingStatus20Extension">
    <xs:sequence>
      <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="FocusStatus20">
    <xs:sequence>
      <xs:element name="Position" type="xs:float"/>
      <xs:element name="MoveStatus" type="tt:MoveStatus"/>
      <xs:element name="Error" type="xs:string" minOccurs="0"/>
      <xs:element name="Extension" type="tt:FocusStatus20Extension" minOccurs="0"/>
    </xs:sequence>
    <xs:anyAttribute processContents="lax"/>
  </xs:complexType>
  <xs:complexType name="FocusStatus20Extension">
    <xs:sequence>
      <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="ImagingSettings20">
    <xs:sequence>
      <xs:element name="BacklightCompensation" type="tt:BacklightCompensation20"
minOccurs="0"/>
      <xs:element name="Brightness" type="xs:float" minOccurs="0"/>
      <xs:element name="ColorSaturation" type="xs:float" minOccurs="0"/>
      <xs:element name="Contrast" type="xs:float" minOccurs="0"/>
      <xs:element name="Exposure" type="tt:Exposure20" minOccurs="0"/>
      <xs:element name="Focus" type="tt:FocusConfiguration20" minOccurs="0"/>
      <xs:element name="IrCutFilter" type="tt:IrCutFilterMode" minOccurs="0"/>
      <xs:element name="Sharpness" type="xs:float" minOccurs="0"/>
      <xs:element name="WideDynamicRange" type="tt:WideDynamicRange20" minOccurs="0"/>
      <xs:element name="WhiteBalance" type="tt:WhiteBalance20" minOccurs="0"/>
      <xs:element name="Extension" type="tt:ImagingSettingsExtension20" minOccurs="0"/>
    </xs:sequence>
    <xs:anyAttribute processContents="lax"/>
  </xs:complexType>
  <xs:complexType name="ImagingSettingsExtension20">
    <xs:sequence>
      <xs:any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
      <xs:element name="ImageStabilization" type="tt:ImageStabilization"
minOccurs="0"/>
      <xs:element name="Extension" type="tt:ImagingSettingsExtension202"
minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="ImagingSettingsExtension202">
    <xs:sequence>
      <xs:element name="IrCutFilterAutoAdjustment" type="tt:IrCutFilterAutoAdjustment"
minOccurs="0" maxOccurs="unbounded"/>
      <xs:element name="Extension" type="tt:ImagingSettingsExtension203"
minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="ImagingSettingsExtension203">
    <xs:sequence>
      <xs:element name="ToneCompensation" type="tt:ToneCompensation" minOccurs="0"/>

```

```

        <xs:element name="Defogging" type="tt:Defogging" minOccurs="0"/>
        <xs:element name="NoiseReduction" type="tt:NoiseReduction" minOccurs="0"/>
        <xs:element name="Extension" type="tt:ImagingSettingsExtension204"
minOccurs="0"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="ImagingSettingsExtension204">
    <xs:sequence>
        <xs:any namespace="##targetNamespace" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="ImageStabilization">
    <xs:sequence>
        <xs:element name="Mode" type="tt:ImageStabilizationMode"/>
        <xs:element name="Level" type="xs:float" minOccurs="0"/>
        <xs:element name="Extension" type="tt:ImageStabilizationExtension"
minOccurs="0"/>
    </xs:sequence>
    <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="ImageStabilizationExtension">
    <xs:sequence>
        <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
<xs:simpleType name="ImageStabilizationMode">
    <xs:restriction base="xs:string">
        <xs:enumeration value="OFF"/>
        <xs:enumeration value="ON"/>
        <xs:enumeration value="AUTO"/>
        <xs:enumeration value="Extended"/>
    </xs:restriction>
</xs:simpleType>
<xs:complexType name="IrCutFilterAutoAdjustment">
    <xs:sequence>
        <xs:element name="BoundaryType" type="xs:string"/>
        <xs:element name="BoundaryOffset" type="xs:float" minOccurs="0"/>
        <xs:element name="ResponseTime" type="xs:duration" minOccurs="0"/>
        <xs:element name="Extension" type="tt:IrCutFilterAutoAdjustmentExtension"
minOccurs="0"/>
    </xs:sequence>
    <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="IrCutFilterAutoAdjustmentExtension">
    <xs:sequence>
        <xs:any namespace="##targetNamespace" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
<xs:simpleType name="IrCutFilterAutoBoundaryType">
    <xs:restriction base="xs:string">
        <xs:enumeration value="Common"/>
        <xs:enumeration value="ToOn"/>
        <xs:enumeration value="ToOff"/>
        <xs:enumeration value="Extended"/>
    </xs:restriction>
</xs:simpleType>
<xs:complexType name="WideDynamicRange20">
    <xs:sequence>
        <xs:element name="Mode" type="tt:WideDynamicMode"/>
        <xs:element name="Level" type="xs:float" minOccurs="0"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="BacklightCompensation20">
    <xs:sequence>
        <xs:element name="Mode" type="tt:BacklightCompensationMode"/>
        <xs:element name="Level" type="xs:float" minOccurs="0"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="Exposure20">
    <xs:sequence>
        <xs:element name="Mode" type="tt:ExposureMode"/>
        <xs:element name="Priority" type="tt:ExposurePriority" minOccurs="0"/>
        <xs:element name="Window" type="tt:Rectangle" minOccurs="0"/>
        <xs:element name="MinExposureTime" type="xs:float" minOccurs="0"/>
        <xs:element name="MaxExposureTime" type="xs:float" minOccurs="0"/>
    </xs:sequence>

```

```

<xs:element name="MinGain" type="xs:float" minOccurs="0"/>
<xs:element name="MaxGain" type="xs:float" minOccurs="0"/>
<xs:element name="MinIris" type="xs:float" minOccurs="0"/>
<xs:element name="MaxIris" type="xs:float" minOccurs="0"/>
<xs:element name="ExposureTime" type="xs:float" minOccurs="0"/>
<xs:element name="Gain" type="xs:float" minOccurs="0"/>
<xs:element name="Iris" type="xs:float" minOccurs="0"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="ToneCompensation">
  <xs:sequence>
    <xs:element name="Mode" type="xs:string"/>
    <xs:element name="Level" type="xs:float" minOccurs="0"/>
    <xs:element name="Extension" type="tt:ToneCompensationExtension" minOccurs="0"/>
  </xs:sequence>
  <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="ToneCompensationExtension">
  <xs:sequence>
    <xs:any namespace="##targetNamespace" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<xs:simpleType name="ToneCompensationMode">
  <xs:restriction base="xs:string">
    <xs:enumeration value="OFF"/>
    <xs:enumeration value="ON"/>
    <xs:enumeration value="AUTO"/>
  </xs:restriction>
</xs:simpleType>
<xs:complexType name="Defogging">
  <xs:sequence>
    <xs:element name="Mode" type="xs:string"/>
    <xs:element name="Level" type="xs:float" minOccurs="0"/>
    <xs:element name="Extension" type="tt:DefoggingExtension" minOccurs="0"/>
  </xs:sequence>
  <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="DefoggingExtension">
  <xs:sequence>
    <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<xs:simpleType name="DefoggingMode">
  <xs:restriction base="xs:string">
    <xs:enumeration value="OFF"/>
    <xs:enumeration value="ON"/>
    <xs:enumeration value="AUTO"/>
  </xs:restriction>
</xs:simpleType>
<xs:complexType name="NoiseReduction">
  <xs:sequence>
    <xs:element name="Level" type="xs:float"/>
    <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="ImagingOptions20">
  <xs:sequence>
    <xs:element name="BacklightCompensation" type="tt:BacklightCompensationOptions20"
minOccurs="0"/>
    <xs:element name="Brightness" type="tt:FloatRange" minOccurs="0"/>
    <xs:element name="ColorSaturation" type="tt:FloatRange" minOccurs="0"/>
    <xs:element name="Contrast" type="tt:FloatRange" minOccurs="0"/>
    <xs:element name="Exposure" type="tt:ExposureOptions20" minOccurs="0"/>
    <xs:element name="Focus" type="tt:FocusOptions20" minOccurs="0"/>
    <xs:element name="IrCutFilterModes" type="tt:IrCutFilterMode" minOccurs="0"
maxOccurs="unbounded"/>
    <xs:element name="Sharpness" type="tt:FloatRange" minOccurs="0"/>
    <xs:element name="WideDynamicRange" type="tt:WideDynamicRangeOptions20"
minOccurs="0"/>
    <xs:element name="WhiteBalance" type="tt:WhiteBalanceOptions20" minOccurs="0"/>
    <xs:element name="Extension" type="tt:ImagingOptions20Extension" minOccurs="0"/>
  </xs:sequence>
  <xs:anyAttribute processContents="lax"/>
</xs:complexType>

```

```

<xs:complexType name="ImagingOptions20Extension">
  <xs:sequence>
    <xs:any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    <xs:element name="ImageStabilization" type="tt:ImageStabilizationOptions"
minOccurs="0"/>
    <xs:element name="Extension" type="tt:ImagingOptions20Extension2" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="ImagingOptions20Extension2">
  <xs:sequence>
    <xs:element name="IrCutFilterAutoAdjustment"
type="tt:IrCutFilterAutoAdjustmentOptions" minOccurs="0"/>
    <xs:element name="Extension" type="tt:ImagingOptions20Extension3" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="ImagingOptions20Extension3">
  <xs:sequence>
    <xs:element name="ToneCompensationOptions" type="tt:ToneCompensationOptions"
minOccurs="0"/>
    <xs:element name="DefoggingOptions" type="tt:DefoggingOptions" minOccurs="0"/>
    <xs:element name="NoiseReductionOptions" type="tt:NoiseReductionOptions"
minOccurs="0"/>
    <xs:element name="Extension" type="tt:ImagingOptions20Extension4" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="ImagingOptions20Extension4">
  <xs:sequence>
    <xs:any namespace="##targetNamespace" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="ImageStabilizationOptions">
  <xs:sequence>
    <xs:element name="Mode" type="tt:ImageStabilizationMode" maxOccurs="unbounded"/>
    <xs:element name="Level" type="tt:FloatRange" minOccurs="0"/>
    <xs:element name="Extension" type="tt:ImageStabilizationOptionsExtension"
minOccurs="0"/>
  </xs:sequence>
  <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="ImageStabilizationOptionsExtension">
  <xs:sequence>
    <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="IrCutFilterAutoAdjustmentOptions">
  <xs:sequence>
    <xs:element name="BoundaryType" type="xs:string" maxOccurs="unbounded"/>
    <xs:element name="BoundaryOffset" type="xs:boolean" minOccurs="0"/>
    <xs:element name="ResponseTimeRange" type="tt:DurationRange" minOccurs="0"/>
    <xs:element name="Extension" type="tt:IrCutFilterAutoAdjustmentOptionsExtension"
minOccurs="0"/>
  </xs:sequence>
  <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="IrCutFilterAutoAdjustmentOptionsExtension">
  <xs:sequence>
    <xs:any namespace="##targetNamespace" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="WideDynamicRangeOptions20">
  <xs:sequence>
    <xs:element name="Mode" type="tt:WideDynamicMode" maxOccurs="unbounded"/>
    <xs:element name="Level" type="tt:FloatRange" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="BacklightCompensationOptions20">
  <xs:sequence>
    <xs:element name="Mode" type="tt:BacklightCompensationMode"
maxOccurs="unbounded"/>
    <xs:element name="Level" type="tt:FloatRange" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="ExposureOptions20">
  <xs:sequence>

```

```

<xs:element name="Mode" type="tt:ExposureMode" maxOccurs="unbounded"/>
<xs:element name="Priority" type="tt:ExposurePriority" minOccurs="0"
maxOccurs="unbounded"/>
<xs:element name="MinExposureTime" type="tt:FloatRange" minOccurs="0"/>
<xs:element name="MaxExposureTime" type="tt:FloatRange" minOccurs="0"/>
<xs:element name="MinGain" type="tt:FloatRange" minOccurs="0"/>
<xs:element name="MaxGain" type="tt:FloatRange" minOccurs="0"/>
<xs:element name="MinIris" type="tt:FloatRange" minOccurs="0"/>
<xs:element name="MaxIris" type="tt:FloatRange" minOccurs="0"/>
<xs:element name="ExposureTime" type="tt:FloatRange" minOccurs="0"/>
<xs:element name="Gain" type="tt:FloatRange" minOccurs="0"/>
<xs:element name="Iris" type="tt:FloatRange" minOccurs="0"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="MoveOptions20">
<xs:sequence>
<xs:element name="Absolute" type="tt:AbsoluteFocusOptions" minOccurs="0"/>
<xs:element name="Relative" type="tt:RelativeFocusOptions20" minOccurs="0"/>
<xs:element name="Continuous" type="tt:ContinuousFocusOptions" minOccurs="0"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="RelativeFocusOptions20">
<xs:sequence>
<xs:element name="Distance" type="tt:FloatRange"/>
<xs:element name="Speed" type="tt:FloatRange" minOccurs="0"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="WhiteBalance20">
<xs:sequence>
<xs:element name="Mode" type="tt:WhiteBalanceMode"/>
<xs:element name="CrGain" type="xs:float" minOccurs="0"/>
<xs:element name="CbGain" type="xs:float" minOccurs="0"/>
<xs:element name="Extension" type="tt:WhiteBalance20Extension" minOccurs="0"/>
</xs:sequence>
<xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="WhiteBalance20Extension">
<xs:sequence>
<xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="FocusConfiguration20">
<xs:sequence>
<xs:element name="AutoFocusMode" type="tt:AutoFocusMode"/>
<xs:element name="DefaultSpeed" type="xs:float" minOccurs="0"/>
<xs:element name="NearLimit" type="xs:float" minOccurs="0"/>
<xs:element name="FarLimit" type="xs:float" minOccurs="0"/>
<xs:element name="Extension" type="tt:FocusConfiguration20Extension"
minOccurs="0"/>
</xs:sequence>
<xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="FocusConfiguration20Extension">
<xs:sequence>
<xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="WhiteBalanceOptions20">
<xs:sequence>
<xs:element name="Mode" type="tt:WhiteBalanceMode" maxOccurs="unbounded"/>
<xs:element name="YrGain" type="tt:FloatRange" minOccurs="0"/>
<xs:element name="YbGain" type="tt:FloatRange" minOccurs="0"/>
<xs:element name="Extension" type="tt:WhiteBalanceOptions20Extension"
minOccurs="0"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="WhiteBalanceOptions20Extension">
<xs:sequence>
<xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="FocusOptions20">
<xs:sequence>
<xs:element name="AutoFocusModes" type="tt:AutoFocusMode" minOccurs="0"
maxOccurs="unbounded"/>

```

```

        <xs:element name="DefaultSpeed" type="tt:FloatRange" minOccurs="0"/>
        <xs:element name="NearLimit" type="tt:FloatRange" minOccurs="0"/>
        <xs:element name="FarLimit" type="tt:FloatRange" minOccurs="0"/>
        <xs:element name="Extension" type="tt:FocusOptions20Extension" minOccurs="0"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="FocusOptions20Extension">
    <xs:sequence>
        <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="ToneCompensationOptions">
    <xs:sequence>
        <xs:element name="Mode" type="xs:string" maxOccurs="unbounded"/>
        <xs:element name="Level" type="xs:boolean"/>
        <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="DefoggingOptions">
    <xs:sequence>
        <xs:element name="Mode" type="xs:string" maxOccurs="unbounded"/>
        <xs:element name="Level" type="xs:boolean"/>
        <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="NoiseReductionOptions">
    <xs:sequence>
        <xs:element name="Level" type="xs:boolean"/>
        <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="Polyline">
    <xs:sequence>
        <xs:element name="Point" type="tt:Vector" minOccurs="2" maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
<xs:element name="Polyline" type="tt:Polyline"/>
<xs:simpleType name="Direction">
    <xs:restriction base="xs:string">
        <xs:enumeration value="Left"/>
        <xs:enumeration value="Right"/>
        <xs:enumeration value="Any"/>
    </xs:restriction>
</xs:simpleType>
<xs:complexType name="AnalyticsEngineConfiguration">
    <xs:sequence>
        <xs:element name="AnalyticsModule" type="tt:Config" minOccurs="0"
maxOccurs="unbounded"/>
        <xs:element name="Extension" type="tt:AnalyticsEngineConfigurationExtension"
minOccurs="0"/>
    </xs:sequence>
    <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="AnalyticsEngineConfigurationExtension">
    <xs:sequence>
        <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="RuleEngineConfiguration">
    <xs:sequence>
        <xs:element name="Rule" type="tt:Config" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="Extension" type="tt:RuleEngineConfigurationExtension"
minOccurs="0"/>
    </xs:sequence>
    <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="RuleEngineConfigurationExtension">
    <xs:sequence>
        <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>

```



```

</xs:sequence>
</xs:complexType>
<xs:complexType name="Config">
  <xs:sequence>
    <xs:element name="Parameters" type="tt:ItemList"/>
  </xs:sequence>
  <xs:attribute name="Name" type="xs:string" use="required"/>
  <xs:attribute name="Type" type="xs:QName" use="required"/>
</xs:complexType>
<xs:complexType name="ConfigDescription">
  <xs:sequence>
    <xs:element name="Parameters" type="tt:ItemListDescription"/>
    <xs:element name="Messages" minOccurs="0" maxOccurs="unbounded">
      <xs:complexType>
        <xs:complexContent>
          <xs:extension base="tt:MessageDescription">
            <xs:sequence>
              <xs:element name="ParentTopic" type="xs:string"/>
            </xs:sequence>
          </xs:extension>
        </xs:complexContent>
      </xs:complexType>
    </xs:element>
    <xs:element name="Extension" type="tt:ConfigDescriptionExtension" minOccurs="0"/>
  </xs:sequence>
  <xs:attribute name="Name" type="xs:QName" use="required"/>
  <xs:attribute name="fixed" type="xs:boolean"/>
  <xs:attribute name="maxInstances" type="xs:integer"/>
  <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="ConfigDescriptionExtension">
  <xs:sequence>
    <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="SupportedRules">
  <xs:sequence>
    <xs:element name="RuleContentSchemaLocation" type="xs:anyURI" minOccurs="0"
maxOccurs="unbounded"/>
    <xs:element name="RuleDescription" type="tt:ConfigDescription" minOccurs="0"
maxOccurs="unbounded"/>
    <xs:element name="Extension" type="tt:SupportedRulesExtension" minOccurs="0"/>
  </xs:sequence>
  <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="SupportedRulesExtension">
  <xs:sequence>
    <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="SupportedAnalyticsModules">
  <xs:sequence>
    <xs:element name="AnalyticsModuleContentSchemaLocation" type="xs:anyURI"
minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="AnalyticsModuleDescription" type="tt:ConfigDescription"
minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="Extension" type="tt:SupportedAnalyticsModulesExtension"
minOccurs="0"/>
  </xs:sequence>
  <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="SupportedAnalyticsModulesExtension">
  <xs:sequence>
    <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="PolygonConfiguration">
  <xs:sequence>
    <xs:element name="Polygon" type="tt:Polygon"/>
    <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="PolylineArray">

```



```

    <xs:sequence>
      <xs:element name="Segment" type="tt:Polyline" maxOccurs="unbounded"/>
      <xs:element name="Extension" type="tt:PolylineArrayExtension" minOccurs="0"/>
    </xs:sequence>
    <xs:anyAttribute processContents="lax"/>
  </xs:complexType>
  <xs:complexType name="PolylineArrayExtension">
    <xs:sequence>
      <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="PolylineArrayConfiguration">
    <xs:sequence>
      <xs:element name="PolylineArray" type="tt:PolylineArray"/>
      <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:anyAttribute processContents="lax"/>
  </xs:complexType>
  <xs:complexType name="MotionExpression">
    <xs:sequence>
      <xs:element name="Expression" type="xs:string"/>
      <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="Type" type="xs:string"/>
    <xs:anyAttribute processContents="lax"/>
  </xs:complexType>
  <xs:complexType name="MotionExpressionConfiguration">
    <xs:sequence>
      <xs:element name="MotionExpression" type="tt:MotionExpression"/>
      <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:anyAttribute processContents="lax"/>
  </xs:complexType>
  <xs:complexType name="CellLayout">
    <xs:sequence>
      <xs:element name="Transformation" type="tt:Transformation"/>
      <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="Columns" type="xs:integer" use="required"/>
    <xs:attribute name="Rows" type="xs:integer" use="required"/>
    <xs:anyAttribute processContents="lax"/>
  </xs:complexType>
  <xs:complexType name="PaneConfiguration">
    <xs:sequence>
      <xs:element name="PaneName" type="xs:string" minOccurs="0"/>
      <xs:element name="AudioOutputToken" type="tt:ReferenceToken" minOccurs="0"/>
      <xs:element name="AudioSourceToken" type="tt:ReferenceToken" minOccurs="0"/>
      <xs:element name="AudioEncoderConfiguration" type="tt:AudioEncoderConfiguration"
minOccurs="0"/>
      <xs:element name="ReceiverToken" type="tt:ReferenceToken" minOccurs="0"/>
      <xs:element name="Token" type="tt:Token"/>
      <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:anyAttribute processContents="lax"/>
  </xs:complexType>
  <xs:complexType name="PaneLayout">
    <xs:sequence>
      <xs:element name="Pane" type="tt:ReferenceToken"/>
      <xs:element name="Area" type="tt:Rectangle"/>
      <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:anyAttribute processContents="lax"/>
  </xs:complexType>
  <xs:complexType name="Layout">
    <xs:sequence>
      <xs:element name="PaneLayout" type="tt:PaneLayout" maxOccurs="unbounded"/>
      <xs:element name="Extension" type="tt:LayoutExtension" minOccurs="0"/>
    </xs:sequence>
    <xs:anyAttribute processContents="lax"/>
  </xs:complexType>
  <xs:complexType name="LayoutExtension">

```

```

        <xs:sequence>
            <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="CodingCapabilities">
        <xs:sequence>
            <xs:element name="AudioEncodingCapabilities"
type="tt:AudioEncoderConfigurationOptions" minOccurs="0"/>
            <xs:element name="AudioDecodingCapabilities"
type="tt:AudioDecoderConfigurationOptions" minOccurs="0"/>
            <xs:element name="VideoDecodingCapabilities"
type="tt:VideoDecoderConfigurationOptions"/>
            <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:anyAttribute processContents="lax"/>
    </xs:complexType>
    <xs:complexType name="LayoutOptions">
        <xs:sequence>
            <xs:element name="PaneLayoutOptions" type="tt:PaneLayoutOptions"
maxOccurs="unbounded"/>
            <xs:element name="Extension" type="tt:LayoutOptionsExtension" minOccurs="0"/>
        </xs:sequence>
        <xs:anyAttribute processContents="lax"/>
    </xs:complexType>
    <xs:complexType name="LayoutOptionsExtension">
        <xs:sequence>
            <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="PaneLayoutOptions">
        <xs:sequence>
            <xs:element name="Area" type="tt:Rectangle" maxOccurs="unbounded"/>
            <xs:element name="Extension" type="tt:PaneOptionExtension" minOccurs="0"/>
        </xs:sequence>
        <xs:anyAttribute processContents="lax"/>
    </xs:complexType>
    <xs:complexType name="PaneOptionExtension">
        <xs:sequence>
            <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="EventFilter">
        <xs:complexContent>
            <xs:extension base="wsnt:FilterType">
                <xs:anyAttribute processContents="lax"/>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
    <xs:complexType name="PTZPositionFilter">
        <xs:sequence>
            <xs:element name="MinPosition" type="tt:PTZVector"/>
            <xs:element name="MaxPosition" type="tt:PTZVector"/>
            <xs:element name="EnterOrExit" type="xs:boolean"/>
            <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:anyAttribute processContents="lax"/>
    </xs:complexType>
    <xs:complexType name="MetadataFilter">
        <xs:sequence>
            <xs:element name="MetadataStreamFilter" type="tt:XPathExpression"/>
            <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:anyAttribute processContents="lax"/>
    </xs:complexType>
    <xs:simpleType name="XPathExpression">
        <xs:restriction base="xs:string"/>
    </xs:simpleType>
    <xs:complexType name="AnalyticsEngine">
        <xs:complexContent>
            <xs:extension base="tt:ConfigurationEntity">
                <xs:sequence>
                    <xs:element name="AnalyticsEngineConfiguration"

```

```

type="tt:AnalyticsDeviceEngineConfiguration"/>
  <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:anyAttribute processContents="lax"/>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="AnalyticsDeviceEngineConfiguration">
  <xs:sequence>
    <xs:element name="EngineConfiguration" type="tt:EngineConfiguration"
maxOccurs="unbounded"/>
    <xs:element name="Extension"
type="tt:AnalyticsDeviceEngineConfigurationExtension" minOccurs="0"/>
  </xs:sequence>
  <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="AnalyticsDeviceEngineConfigurationExtension">
  <xs:sequence>
    <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="EngineConfiguration">
  <xs:sequence>
    <xs:element name="VideoAnalyticsConfiguration"
type="tt:VideoAnalyticsConfiguration"/>
    <xs:element name="AnalyticsEngineInputInfo" type="tt:AnalyticsEngineInputInfo"/>
    <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="AnalyticsEngineInputInfo">
  <xs:sequence>
    <xs:element name="InputInfo" type="tt:Config" minOccurs="0"/>
    <xs:element name="Extension" type="tt:AnalyticsEngineInputInfoExtension"
minOccurs="0"/>
  </xs:sequence>
  <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="AnalyticsEngineInputInfoExtension">
  <xs:sequence>
    <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="AnalyticsEngineInput">
  <xs:complexContent>
    <xs:extension base="tt:ConfigurationEntity">
      <xs:sequence>
        <xs:element name="SourceIdentification" type="tt:SourceIdentification"/>
        <xs:element name="VideoInput" type="tt:VideoEncoderConfiguration"/>
        <xs:element name="MetadataInput" type="tt:MetadataInput"/>
        <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:anyAttribute processContents="lax"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="SourceIdentification">
  <xs:sequence>
    <xs:element name="Name" type="xs:string"/>
    <xs:element name="Token" type="tt:ReferenceToken" maxOccurs="unbounded"/>
    <xs:element name="Extension" type="tt:SourceIdentificationExtension"
minOccurs="0"/>
  </xs:sequence>
  <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="SourceIdentificationExtension">
  <xs:sequence>
    <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="MetadataInput">
  <xs:sequence>

```

```

        <xs:element name="MetadataConfig" type="tt:Config" minOccurs="0"
maxOccurs="unbounded"/>
        <xs:element name="Extension" type="tt:MetadataInputExtension" minOccurs="0"/>
    </xs:sequence>
    <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="MetadataInputExtension">
    <xs:sequence>
        <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="AnalyticsEngineControl">
    <xs:complexContent>
        <xs:extension base="tt:ConfigurationEntity">
            <xs:sequence>
                <xs:element name="EngineToken" type="tt:ReferenceToken"/>
                <xs:element name="EngineConfigToken" type="tt:ReferenceToken"/>
                <xs:element name="InputToken" type="tt:ReferenceToken"
maxOccurs="unbounded"/>
                <xs:element name="ReceiverToken" type="tt:ReferenceToken"
maxOccurs="unbounded"/>
                <xs:element name="Multicast" type="tt:MulticastConfiguration" minOccurs="0"/>
                <xs:element name="Subscription" type="tt:Config"/>
                <xs:element name="Mode" type="tt:ModeOfOperation"/>
                <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
            </xs:sequence>
            <xs:anyAttribute processContents="lax"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:simpleType name="ModeOfOperation">
    <xs:restriction base="xs:string">
        <xs:enumeration value="Idle"/>
        <xs:enumeration value="Active"/>
        <xs:enumeration value="Unknown"/>
    </xs:restriction>
</xs:simpleType>
<xs:complexType name="AnalyticsStateInformation">
    <xs:sequence>
        <xs:element name="AnalyticsEngineControlToken" type="tt:ReferenceToken"/>
        <xs:element name="State" type="tt:AnalyticsState"/>
        <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="AnalyticsState">
    <xs:sequence>
        <xs:element name="Error" type="xs:string" minOccurs="0"/>
        <xs:element name="State" type="xs:string"/>
        <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="ActionEngineEventPayload">
    <xs:sequence>
        <xs:element name="RequestInfo" type="soapenv:Envelope" minOccurs="0"
maxOccurs="1"/>
        <xs:element name="ResponseInfo" type="soapenv:Envelope" minOccurs="0"
maxOccurs="1"/>
        <xs:element name="Fault" type="soapenv:Fault" minOccurs="0" maxOccurs="1"/>
        <xs:element name="Extension" type="tt:ActionEngineEventPayloadExtension"
minOccurs="0"/>
    </xs:sequence>
    <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="ActionEngineEventPayloadExtension">
    <xs:sequence>
        <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:simpleType name="AudioClassType">
    <xs:restriction base="xs:string"/>

```

```

</xs:simpleType>
<xs:complexType name="AudioClassCandidate">
  <xs:sequence>
    <xs:element name="Type" type="tt:AudioClassType"/>
    <xs:element name="Likelihood" type="xs:float"/>
    <xs:any namespace="##targetNamespace" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="AudioClassDescriptor">
  <xs:sequence>
    <xs:element name="ClassCandidate" type="tt:AudioClassCandidate" minOccurs="0"
maxOccurs="unbounded"/>
    <xs:element name="Extension" type="tt:AudioClassDescriptorExtension"
minOccurs="0"/>
  </xs:sequence>
  <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="AudioClassDescriptorExtension">
  <xs:sequence>
    <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="ActiveConnection">
  <xs:sequence>
    <xs:element name="CurrentBitrate" type="xs:float"/>
    <xs:element name="CurrentFps" type="xs:float"/>
    <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="ProfileStatus">
  <xs:sequence>
    <xs:element name="ActiveConnections" type="tt:ActiveConnection" minOccurs="0"
maxOccurs="unbounded"/>
    <xs:element name="Extension" type="tt:ProfileStatusExtension" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="ProfileStatusExtension">
  <xs:sequence>
    <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="OSDReference">
  <xs:simpleContent>
    <xs:extension base="tt:ReferenceToken">
      <xs:anyAttribute processContents="lax"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
<xs:simpleType name="OSDType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Text"/>
    <xs:enumeration value="Image"/>
    <xs:enumeration value="Extended"/>
  </xs:restriction>
</xs:simpleType>
<xs:complexType name="OSDPosConfiguration">
  <xs:sequence>
    <xs:element name="Type" type="xs:string"/>
    <xs:element name="Pos" type="tt:Vector" minOccurs="0"/>
    <xs:element name="Extension" type="tt:OSDPosConfigurationExtension"
minOccurs="0"/>
  </xs:sequence>
  <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="OSDPosConfigurationExtension">
  <xs:sequence>
    <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:anyAttribute processContents="lax"/>

```

```

</xs:complexType>
<xs:complexType name="OSDColor">
  <xs:sequence>
    <xs:element name="Color" type="tt:Color"/>
  </xs:sequence>
  <xs:attribute name="Transparent" type="xs:int" use="optional"/>
  <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="OSDTextConfiguration">
  <xs:sequence>
    <xs:element name="Type" type="xs:string"/>
    <xs:element name="DateFormat" type="xs:string" minOccurs="0"/>
    <xs:element name="TimeFormat" type="xs:string" minOccurs="0"/>
    <xs:element name="FontSize" type="xs:int" minOccurs="0"/>
    <xs:element name="FontColor" type="tt:OSDColor" minOccurs="0"/>
    <xs:element name="BackgroundColor" type="tt:OSDColor" minOccurs="0"/>
    <xs:element name="PlainText" type="xs:string" minOccurs="0"/>
    <xs:element name="Extension" type="tt:OSDTextConfigurationExtension"
minOccurs="0"/>
  </xs:sequence>
  <xs:attribute name="IsPersistentText" type="xs:boolean" use="optional"/>
  <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="OSDTextConfigurationExtension">
  <xs:sequence>
    <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="OSDImgConfiguration">
  <xs:sequence>
    <xs:element name="ImgPath" type="xs:anyURI"/>
    <xs:element name="Extension" type="tt:OSDImgConfigurationExtension"
minOccurs="0"/>
  </xs:sequence>
  <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="OSDImgConfigurationExtension">
  <xs:sequence>
    <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="ColorspaceRange">
  <xs:sequence>
    <xs:element name="X" type="tt:FloatRange"/>
    <xs:element name="Y" type="tt:FloatRange"/>
    <xs:element name="Z" type="tt:FloatRange"/>
    <xs:element name="Colorspace" type="xs:anyURI"/>
  </xs:sequence>
  <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="ColorOptions">
  <xs:choice>
    <xs:element name="ColorList" type="tt:Color" maxOccurs="unbounded"/>
    <xs:element name="ColorspaceRange" type="tt:ColorspaceRange"
maxOccurs="unbounded"/>
  </xs:choice>
  <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="OSDColorOptions">
  <xs:sequence>
    <xs:element name="Color" type="tt:ColorOptions" minOccurs="0"/>
    <xs:element name="Transparent" type="tt:IntRange" minOccurs="0"/>
    <xs:element name="Extension" type="tt:OSDColorOptionsExtension" minOccurs="0"/>
  </xs:sequence>
  <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="OSDColorOptionsExtension">
  <xs:sequence>
    <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="OSDTextOptions">

```

```

    <xs:sequence>
      <xs:element name="Type" type="xs:string" maxOccurs="unbounded"/>
      <xs:element name="FontSizeRange" type="tt:IntRange" minOccurs="0"/>
      <xs:element name="DateFormat" type="xs:string" minOccurs="0"
maxOccurs="unbounded"/>
      <xs:element name="TimeFormat" type="xs:string" minOccurs="0"
maxOccurs="unbounded"/>
      <xs:element name="FontColor" type="tt:OSDColorOptions" minOccurs="0"/>
      <xs:element name="BackgroundColor" type="tt:OSDColorOptions" minOccurs="0"/>
      <xs:element name="Extension" type="tt:OSDTextOptionsExtension" minOccurs="0"/>
    </xs:sequence>
    <xs:anyAttribute processContents="lax"/>
  </xs:complexType>
  <xs:complexType name="OSDTextOptionsExtension">
    <xs:sequence>
      <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:anyAttribute processContents="lax"/>
  </xs:complexType>
  <xs:complexType name="OSDImgOptions">
    <xs:sequence>
      <xs:element name="ImagePath" type="xs:anyURI" maxOccurs="unbounded"/>
      <xs:element name="Extension" type="tt:OSDImgOptionsExtension" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="FormatsSupported" type="tt:StringAttrList"/>
    <xs:attribute name="MaxSize" type="xs:int"/>
    <xs:attribute name="MaxWidth" type="xs:int"/>
    <xs:attribute name="MaxHeight" type="xs:int"/>
    <xs:anyAttribute processContents="lax"/>
  </xs:complexType>
  <xs:complexType name="OSDImgOptionsExtension">
    <xs:sequence>
      <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:anyAttribute processContents="lax"/>
  </xs:complexType>
  <xs:complexType name="OSDConfiguration">
    <xs:complexContent>
      <xs:extension base="tt:DeviceEntity">
        <xs:sequence>
          <xs:element name="VideoSourceConfigurationToken" type="tt:OSDReference"/>
          <xs:element name="Type" type="tt:OSDType"/>
          <xs:element name="Position" type="tt:OSDPosConfiguration"/>
          <xs:element name="TextString" type="tt:OSDTextConfiguration" minOccurs="0"/>
          <xs:element name="Image" type="tt:OSDImgConfiguration" minOccurs="0"/>
          <xs:element name="Extension" type="tt:OSDConfigurationExtension"
minOccurs="0"/>
        </xs:sequence>
        <xs:anyAttribute processContents="lax"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="OSDConfigurationExtension">
    <xs:sequence>
      <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:anyAttribute processContents="lax"/>
  </xs:complexType>
  <xs:complexType name="MaximumNumberOfOSDs">
    <xs:attribute name="Total" type="xs:int" use="required"/>
    <xs:attribute name="Image" type="xs:int"/>
    <xs:attribute name="PlainText" type="xs:int"/>
    <xs:attribute name="Date" type="xs:int"/>
    <xs:attribute name="Time" type="xs:int"/>
    <xs:attribute name="DateAndTime" type="xs:int"/>
    <xs:anyAttribute processContents="lax"/>
  </xs:complexType>
  <xs:complexType name="OSDConfigurationOptions">
    <xs:sequence>
      <xs:element name="MaximumNumberOfOSDs" type="tt:MaximumNumberOfOSDs"/>
      <xs:element name="Type" type="tt:OSDType" maxOccurs="unbounded"/>
      <xs:element name="PositionOption" type="xs:string" maxOccurs="unbounded"/>
      <xs:element name="TextOption" type="tt:OSDTextOptions" minOccurs="0"/>
      <xs:element name="ImageOption" type="tt:OSDImgOptions" minOccurs="0"/>
      <xs:element name="Extension" type="tt:OSDConfigurationOptionsExtension"

```

```
minOccurs="0"/>
</xs:sequence>
<xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="OSDConfigurationOptionsExtension">
  <xs:sequence>
    <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:anyAttribute processContents="lax"/>
</xs:complexType>
</xs:schema>
```

IECNORM.COM : Click to view the full PDF of IEC 62676-2-31:2019

F.8 Streaming metadata schema

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:tt="http://www.onvif.org/ver10/schema" xmlns:wsnt="http://docs.oasis-
  open.org/wsn/b-2" targetNamespace="http://www.onvif.org/ver10/schema"
  elementFormDefault="qualified" version="18.05">
  <xs:include schemaLocation="common.xsd"/>
  <xs:import namespace="http://docs.oasis-open.org/wsn/b-2"
  schemaLocation="http://docs.oasis-open.org/wsn/b-2.xsd"/>
  <xs:complexType name="Appearance">
    <xs:sequence>
      <xs:element name="Transformation" type="tt:Transformation" minOccurs="0"/>
      <xs:element name="Shape" type="tt:ShapeDescriptor" minOccurs="0"/>
      <xs:element name="Color" type="tt:ColorDescriptor" minOccurs="0"/>
      <xs:element name="Class" type="tt:ClassDescriptor" minOccurs="0"/>
      <xs:element name="Extension" type="tt:AppearanceExtension" minOccurs="0"/>
      <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:anyAttribute processContents="lax"/>
  </xs:complexType>
  <xs:complexType name="AppearanceExtension">
    <xs:sequence>
      <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="ShapeDescriptor">
    <xs:sequence>
      <xs:element name="BoundingBox" type="tt:Rectangle"/>
      <xs:element name="CenterOfGravity" type="tt:Vector"/>
      <xs:element name="Polygon" type="tt:Polygon" minOccurs="0"
maxOccurs="unbounded"/>
      <xs:element name="Extension" type="tt:ShapeDescriptorExtension" minOccurs="0"/>
      <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:anyAttribute processContents="lax"/>
  </xs:complexType>
  <xs:complexType name="ShapeDescriptorExtension">
    <xs:sequence>
      <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="ColorDescriptor">
    <xs:sequence>
      <xs:element name="ColorCluster" minOccurs="0" maxOccurs="unbounded">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Color" type="tt:Color"/>
            <xs:element name="Weight" type="xs:float" minOccurs="0"/>
            <xs:element name="Covariance" type="tt:ColorCovariance" minOccurs="0"/>
            <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="Extension" type="tt:ColorDescriptorExtension" minOccurs="0"/>
      <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:anyAttribute processContents="lax"/>
  </xs:complexType>
  <xs:complexType name="ColorDescriptorExtension">
    <xs:sequence>
      <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:simpleType name="ClassType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="Animal"/>
      <xs:enumeration value="Face"/>
      <xs:enumeration value="Human"/>
      <xs:enumeration value="Vehical"/>
    </xs:restriction>
  </xs:simpleType>
</xs:schema>
```

```

        <xs:enumeration value="Other"/>
    </xs:restriction>
</xs:simpleType>
<xs:complexType name="StringLikelihood">
    <xs:simpleContent>
        <xs:extension base="xs:string">
            <xs:attribute name="Likelihood" type="xs:float"/>
            <xs:anyAttribute processContents="lax"/>
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>
<xs:complexType name="ClassDescriptor">
    <xs:sequence>
        <xs:element name="ClassCandidate" minOccurs="0" maxOccurs="unbounded">
            <xs:complexType>
                <xs:sequence>
                    <xs:element name="Type" type="tt:ClassType"/>
                    <xs:element name="Likelihood" type="xs:float"/>
                    <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
                </xs:sequence>
            </xs:complexType>
        </xs:element>
        <xs:element name="Extension" type="tt:ClassDescriptorExtension" minOccurs="0"/>
        <xs:element name="Type" type="tt:StringLikelihood" minOccurs="0"
maxOccurs="unbounded"/>
        <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="ClassDescriptorExtension">
    <xs:sequence>
        <xs:any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
        <xs:element name="OtherTypes" type="tt:OtherType" maxOccurs="unbounded"/>
        <xs:element name="Extension" type="tt:ClassDescriptorExtension2" minOccurs="0"/>
        <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="ClassDescriptorExtension2">
    <xs:sequence>
        <xs:any namespace="##targetNamespace" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="OtherType">
    <xs:sequence>
        <xs:element name="Type" type="xs:string"/>
        <xs:element name="Likelihood" type="xs:float"/>
        <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="Object">
    <xs:complexContent>
        <xs:extension base="tt:ObjectId">
            <xs:sequence>
                <xs:element name="Appearance" type="tt:Appearance" minOccurs="0"/>
                <xs:element name="Behaviour" type="tt:Behaviour" minOccurs="0"/>
                <xs:element name="Extension" type="tt:ObjectExtension" minOccurs="0"/>
                <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
            </xs:sequence>
            <xs:anyAttribute processContents="lax"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="ObjectExtension">
    <xs:sequence>
        <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="Frame">
    <xs:sequence>

```

```

    <xs:element name="PTZStatus" type="tt:PTZStatus" minOccurs="0"/>
    <xs:element name="Transformation" type="tt:Transformation" minOccurs="0"/>
    <xs:element name="Object" type="tt:Object" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="ObjectTree" type="tt:ObjectTree" minOccurs="0"/>
    <xs:element name="Extension" type="tt:FrameExtension" minOccurs="0"/>
    <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="UtcTime" type="xs:dateTime" use="required"/>
  <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="FrameExtension">
  <xs:sequence>
    <xs:any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    <xs:element name="MotionInCells" type="tt:MotionInCells" minOccurs="0"/>
    <xs:element name="Extension" type="tt:FrameExtension2" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="FrameExtension2">
  <xs:sequence>
    <xs:any namespace="##targetNamespace" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="Merge">
  <xs:sequence>
    <xs:element name="from" type="tt:ObjectId" minOccurs="2" maxOccurs="unbounded"/>
    <xs:element name="to" type="tt:ObjectId"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="Split">
  <xs:sequence>
    <xs:element name="from" type="tt:ObjectId"/>
    <xs:element name="to" type="tt:ObjectId" minOccurs="2" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="Rename">
  <xs:sequence>
    <xs:element name="from" type="tt:ObjectId"/>
    <xs:element name="to" type="tt:ObjectId"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="ObjectId">
  <xs:attribute name="ObjectId" type="xs:integer"/>
</xs:complexType>
<xs:complexType name="Behaviour">
  <xs:sequence>
    <xs:element name="Removed" minOccurs="0">
      <xs:complexType>
        <xs:sequence>
          <xs:any namespace="##other" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="Idle" minOccurs="0">
      <xs:complexType>
        <xs:sequence>
          <xs:any namespace="##other" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="Extension" type="tt:BehaviourExtension" minOccurs="0"/>
    <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="BehaviourExtension">
  <xs:sequence>
    <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="ObjectTree">
  <xs:sequence>
    <xs:element name="Rename" type="tt:Rename" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="Split" type="tt:Split" minOccurs="0" maxOccurs="unbounded"/>

```

```

        <xs:element name="Merge" type="tt:Merge" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="Delete" type="tt:ObjectId" minOccurs="0"
maxOccurs="unbounded"/>
        <xs:element name="Extension" type="tt:ObjectTreeExtension" minOccurs="0"/>
        <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="ObjectTreeExtension">
    <xs:sequence>
        <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="MotionInCells">
    <xs:sequence>
        <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="Columns" type="xs:integer" use="required"/>
    <xs:attribute name="Rows" type="xs:integer" use="required"/>
    <xs:attribute name="Cells" type="xs:base64Binary" use="required"/>
    <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="MetadataStream">
    <xs:sequence>
        <xs:choice minOccurs="0" maxOccurs="unbounded">
            <xs:element name="VideoAnalytics" type="tt:VideoAnalyticsStream"/>
            <xs:element name="PTZ" type="tt:PTZStream"/>
            <xs:element name="Event" type="tt:EventStream"/>
            <xs:element name="Extension" type="tt:MetadataStreamExtension"/>
            <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
        </xs:choice>
    </xs:sequence>
    <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="MetadataStreamExtension">
    <xs:sequence>
        <xs:any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
        <xs:element name="AudioAnalyticsStream" type="tt:AudioAnalyticsStream"
minOccurs="0"/>
        <xs:element name="Extension" type="tt:MetadataStreamExtension2" minOccurs="0"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="MetadataStreamExtension2">
    <xs:sequence>
        <xs:any namespace="##targetNamespace" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="AudioAnalyticsStream">
    <xs:sequence>
        <xs:element name="AudioDescriptor" type="tt:AudioDescriptor" minOccurs="0"
maxOccurs="unbounded"/>
        <xs:element name="Extension" type="tt:AudioAnalyticsStreamExtension"
minOccurs="0"/>
        <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="AudioDescriptor">
    <xs:sequence>
        <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="UtcTime" type="xs:dateTime" use="required"/>
    <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="AudioAnalyticsStreamExtension">
    <xs:sequence>
        <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>

```

```
<xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:element name="MetadataStream" type="tt:MetadataStream"/>
<xs:complexType name="VideoAnalyticsStream">
  <xs:choice minOccurs="0" maxOccurs="unbounded">
    <xs:element name="Frame" type="tt:Frame"/>
    <xs:element name="Extension" type="tt:VideoAnalyticsStreamExtension"/>
    <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:choice>
</xs:complexType>
<xs:complexType name="VideoAnalyticsStreamExtension">
  <xs:sequence>
    <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="PTZStream">
  <xs:choice minOccurs="0" maxOccurs="unbounded">
    <xs:element name="PTZStatus" type="tt:PTZStatus"/>
    <xs:element name="Extension" type="tt:PTZStreamExtension"/>
    <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:choice>
</xs:complexType>
<xs:complexType name="PTZStreamExtension">
  <xs:sequence>
    <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="EventStream">
  <xs:choice minOccurs="0" maxOccurs="unbounded">
    <xs:element ref="wsnt:NotificationMessage"/>
    <xs:element name="Extension" type="tt:EventStreamExtension"/>
    <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:choice>
</xs:complexType>
<xs:complexType name="EventStreamExtension">
  <xs:sequence>
    <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
</xs:schema>
```

Bibliography

ISO/IEC 10918-1, *Information technology – Digital compression and coding of continuous-tone still images: Requirements and guidelines*

ISO/IEC 14496-2, *Information technology – Coding of audio-visual objects – Part 2: Visual*

ISO/IEC 14496-3, *Information technology – Coding of audio-visual objects – Part 3: Audio*

ISO/IEC 14496-10, *Information technology – Coding of audio-visual objects – Part 10: Advanced Video Coding*

ISO/IEC 23008-2, *Information technology – High efficiency coding and media delivery in heterogeneous environments – Part 2: High efficiency video coding*

IETF RFC 4571, Framing Real-time Transport Protocol (RTP) and RTP Control Protocol (RTCP) Packets over Connection-Oriented Transport
<<http://www.ietf.org/rfc/rfc4571.txt>>

IETF RFC 4572, Connection-Oriented Media Transport over the Transport Layer Security (TLS) Protocol in the Session Description Protocol (SDP)
<http://www.ietf.org/rfc/rfc4572.txt>

INTERNET ENGINEERING TASK FORCE (IETF), RFC 4585: *Extended RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback (RTP/AVPF)* [online]. Edited by J. Ott et al. July 2006 [viewed 2019-01-08]. Available at
<http://www.ietf.org/rfc/rfc4585.txt>

INTERNET ENGINEERING TASK FORCE (IETF), RFC 5104: *Codec Control Messages in the RTP Audio-Visual Profile with Feedback (AVPF)* [online]. Edited by S. Wenger et al. February 2008 [viewed 2019-01-08]. Available at
<http://www.ietf.org/rfc/rfc5104.txt>

ITU-T Recommendation G.711, *Pulse code modulation (PCM) of voice frequencies* [online]. November 1988 [viewed 2019-01-08]. Available at
http://www.itu.int/rec/dologin_pub.asp?lang=e&id=T-REC-G.711-198811-!!!PDF-E&type=items

ITU-T G.726, *40, 32, 24, 16 kbit/s Adaptive Differential Pulse Code Modulation (ADPCM)* [online]. December 1990 [viewed 2019-01-08]. Available at
http://www.itu.int/rec/dologin_pub.asp?lang=e&id=T-REC-G.726-199012-!!!PDF-E&type=items

ONVIF Profile S Specification
< <https://www.onvif.org/profiles/profile-s> >

ONVIF Profile T Specification
< <https://www.onvif.org/profiles/profile-t> >

IECNORM.COM : Click to view the full PDF of IEC 62676-2-31:2019

SOMMAIRE

AVANT-PROPOS	293
INTRODUCTION.....	295
1 Domaine d'application	296
2 Références normatives	296
3 Termes et définitions	297
4 Présentation	298
4.1 Généralités	298
4.2 Dispositif ES	298
4.3 Configuration d'imagerie	298
4.4 Configuration multimédia	299
4.4.1 Profils multimédias	299
4.4.2 Mode de source vidéo.....	301
4.5 Transmission en continu en temps réel	301
4.6 Contrôle PTZ	303
4.7 Analyse.....	304
4.8 Interfaces.....	305
5 Service ES de dispositif.....	306
5.1 Généralités	306
5.2 VideoOutputs	306
5.2.1 Généralités	306
5.2.2 GetVideoOutputs	306
5.3 VideoOutputConfiguration	306
5.3.1 GetVideoOutputConfiguration	306
5.3.2 SetVideoOutputConfiguration	307
5.3.3 GetVideoOutputConfigurationOptions	307
5.4 VideoSources	308
5.4.1 Généralités	308
5.4.2 GetVideoSources.....	308
5.5 AudioOutputs	308
5.5.1 Généralités	308
5.5.2 GetAudioOutputs	308
5.6 AudioSources	309
5.6.1 Généralités	309
5.6.2 GetAudioSources.....	309
5.7 Fonctionnalités	309
6 Service multimédia	310
6.1 Généralités	310
6.2 Méthodes de profil multimédia	310
6.2.1 Création de profil multimédia	310
6.2.2 Obtention de profils multimédias.....	311
6.2.3 Ajout d'une ou de plusieurs configurations à un profil	312
6.2.4 Suppression d'une ou de plusieurs configurations d'un profil	313
6.2.5 Suppression de profil multimédia	313
6.3 Configurations multimédias	314
6.3.1 Généralités	314
6.3.2 Configuration de source vidéo	314

6.3.3	Configuration de codeur vidéo	315
6.3.4	Configuration de source audio	315
6.3.5	Configuration de codeur audio	315
6.3.6	Configuration PTZ	315
6.3.7	Configuration d'analyse	315
6.3.8	Configuration de métadonnées	316
6.3.9	Configuration de sortie audio	316
6.3.10	Configuration de décodeur audio	317
6.4	Méthodes de configuration multimédia	317
6.4.1	Généralités	317
6.4.2	Obtention des configurations	317
6.4.3	Modification d'une configuration	318
6.4.4	Obtention d'options de configuration	319
6.4.5	GetVideoEncoderInstances	319
6.5	GetStreamUri	320
6.6	GetSnapshotUri	321
6.7	Multidiffusion	322
6.7.1	Généralités	322
6.7.2	Démarrage de transmission en continu en multidiffusion	322
6.7.3	Interruption de transmission en continu en multidiffusion	323
6.8	SetSynchronizationPoint	323
6.9	Mode de source vidéo	324
6.9.1	Généralités	324
6.9.2	GetVideoSourceModes	324
6.9.3	SetVideoSourceMode	325
6.10	Affichage écran (OSD – <i>On-Screen Display</i>)	325
6.10.1	Généralités	325
6.10.2	CreateOSD	326
6.10.3	DeleteOSD	327
6.10.4	GetOSDs	327
6.10.5	SetOSD	328
6.10.6	GetOSDOptions	328
6.11	Masques de confidentialité	329
6.11.1	Généralités	329
6.11.2	CreateMask	330
6.11.3	DeleteMask	330
6.11.4	GetMasks	331
6.11.5	SetMask	331
6.11.6	GetMaskOptions	332
6.12	Fonctionnalités	332
6.13	Événements	333
6.13.1	ProfileChange	333
6.13.2	ConfigurationChange	334
6.13.3	ActiveConnections	334
6.14	Écarts du service multimédia version 1	334
6.14.1	Généralités	334
6.14.2	Gestion des profils	335
6.14.3	Liste des configurations	335
6.14.4	Masques de confidentialité	335

7	Service d'imagerie	335
7.1	Généralités	335
7.2	Réglages d'imagerie	335
7.2.1	Paramètres	335
7.2.2	GetImagingSettings	337
7.2.3	SetImagingSettings	338
7.2.4	GetOptions	339
7.3	Préréglages d'imagerie	339
7.3.1	Généralités	339
7.3.2	GetPresets	339
7.3.3	GetCurrentPreset	340
7.3.4	SetCurrentPreset	341
7.4	Opérations de mise au point	341
7.4.1	Move	341
7.4.2	GetMoveOptions	342
7.4.3	Stop	343
7.4.4	GetImagingStatus	343
7.5	Fonctionnalités	344
8	Service PTZ	344
8.1	Généralités	344
8.2	Nœud PTZ	344
8.2.1	Généralités	344
8.2.2	GetNodes	345
8.2.3	GetNode	345
8.3	Configuration PTZ	346
8.3.1	Généralités	346
8.3.2	GetConfigurations	347
8.3.3	GetConfiguration	347
8.3.4	GetConfigurationOptions	348
8.3.5	SetConfiguration	348
8.3.6	GetCompatibleConfigurations	349
8.4	Opérations de déplacement	349
8.4.1	Généralités	349
8.4.2	AbsoluteMove	350
8.4.3	RelativeMove	351
8.4.4	ContinuousMove	352
8.4.5	GeoMove	353
8.4.6	Stop (Interruption)	354
8.4.7	GetStatus	355
8.5	Opérations de préréglage	356
8.5.1	Généralités	356
8.5.2	SetPreset	356
8.5.3	GetPresets	357
8.5.4	GotoPreset	358
8.5.5	RemovePreset	358
8.6	Opérations de position de départ	359
8.6.1	Généralités	359
8.6.2	GotoHomePosition	359
8.6.3	SetHomePosition	360

8.7	Opérations auxiliaires	360
8.7.1	Généralités	360
8.7.2	SendAuxiliaryCommand	361
8.8	Espaces PTZ prédéfinis	361
8.8.1	Généralités	361
8.8.2	Espaces de position absolue	361
8.8.3	Espaces de translation relative	366
8.8.4	Espaces de vitesse continue	367
8.8.5	Espaces de vitesse	368
8.9	Opérations de tours de préréglage	370
8.9.1	Généralités	370
8.9.2	GetPresetTours	370
8.9.3	GetPresetTour	371
8.9.4	GetPresetTourOptions	371
8.9.5	CreatePresetTour	372
8.9.6	ModifyPresetTour	372
8.9.7	OperatePresetTour	373
8.9.8	RemovePresetTour	374
8.9.9	Paramètres de tour de préréglage	374
8.10	Configuration de direction de contrôle de panoramique/inclinaison	375
8.11	Fonctionnalités	377
8.12	Événements	378
8.12.1	Généralités	378
8.12.2	PréréglagesPTZ	378
8.12.3	PresetTours	379
9	Service d'analyse	379
9.1	Généralités	379
9.2	Interface description de scène	379
9.2.1	Présentation	379
9.2.2	Contenu associé aux trames	379
9.2.3	Éléments de scène	382
9.3	Interface de règles	390
9.3.1	Généralités	390
9.3.2	Représentation des règles	390
9.3.3	Langage de description de règles	391
9.3.4	Opérations sur les règles	392
9.4	Interface de modules d'analyse	396
9.4.1	Généralités	396
9.4.2	Configuration de module d'analyse	396
9.4.3	Langage de description de module d'analyse	396
9.4.4	Opérations sur les modules d'analyse	397
9.5	GetAnalyticsModuleOptions	400
9.6	Fonctionnalités	400
9.7	Événements – Détection audio (Audio Detected)	401
10	Transmission en continu en temps réel	401
10.1	Généralités	401
10.2	Protocole de flux multimédia	401
10.2.1	Format de transport	401
10.2.2	Transport multimédia	403

10.2.3	Points de synchronisation	407
10.2.4	JPEG par protocole RTP	408
10.3	Protocole de contrôle multimédia	411
10.3.1	Contrôle de flux RTSP	411
10.3.2	Méthode de maintien "keep-alive " pour session RTSP	412
10.3.3	Synchronisation audio et vidéo RTSP	413
10.3.4	Session RTSP pour un flux de métadonnées	413
10.3.5	Transmission en continu en multidiffusion.....	414
10.3.6	Exemple de message RTSP	414
10.3.7	RTSP par le protocole HTTP.....	415
10.4	Connexion de voie de retour	415
10.4.1	Généralités	415
10.4.2	Balise RTSP Require	415
10.4.3	Configuration de connexion pour une connexion bidirectionnelle	416
10.4.4	Exemple de description d'un serveur sans prise en charge de voie de retour:	416
10.4.5	Exemple de description d'un serveur avec prise en charge de voie de retour ONVIF:	416
10.4.6	Transmission en continu en multidiffusion.....	418
10.5	Gestion des erreurs	418
Annexe A (normative)	Échange XML effectif (EXI)	419
Annexe B (normative)	Description de la lentille	420
Annexe C (informative)	Spécification de règles	422
C.1	Généralités	422
C.2	LineDetector	422
C.3	FieldDetector	422
C.4	LoiteringDetector	423
C.5	Détecteur de mouvement déclaratif.....	424
C.6	Règle de comptage	425
C.7	Règle de requête.....	426
Annexe D (informative)	Détection de mouvement de cellules	427
D.1	Détecteur de mouvement de cellules.....	427
D.2	Moteur d'analyse de mouvement de cellules	428
D.2.1	Généralités	428
D.2.2	Configuration de module.....	429
Annexe E (normative)	Détection de mouvement.....	432
Annexe F (normative)	Fichiers de schémas.....	434
F.1	Dispositif ES	434
F.2	Imagerie	449
F.3	Multimédia	455
F.4	Media 2.....	492
F.5	PTZ	514
F.6	Analyse.....	527
F.7	Schéma commun	533
F.8	Schéma de métadonnées de transmission en continu	572
Bibliographie.....		577

Figure 1 – Profil multimédia	299
Figure 2 – Configuration de profil complète.....	300
Figure 3 – Structure de couches	302
Figure 4 – Architecture d'analyse.....	304
Figure 5 – Exemple avec quatre configurations OSD	326
Figure 6 – Exemple d'écran avec masque et système de coordonnées	329
Figure 7 – Espace de position de panoramique/inclinaison sphérique en degrés pour une caméra fixée au plafond	363
Figure 8 – Exemple de modifications de la direction de contrôle de panoramique/inclinaison par les fonctionnalités E-Flip et Reverse	377
Figure 9 – Système de coordonnées de trame par défaut	381
Figure 10 – En-tête RTP	403
Figure 11 – Séquence RTCP	406
Figure 12 – Rapport expéditeur RTCP	407
Figure 13 – Synchronisation multimédia.....	407
Figure 14 – Structure de paquet RTP/JPEG	408
Figure 15 – Contrôle de flux.....	411
Figure 16 – Méthode Keep alive.....	413
Figure B.1 – Mise en correspondance optique de l'angle (α) par le rayon (R) avec les coordonnées x/y normalisées.....	420
Figure B.2 – Mise en correspondance lisse avec approximations B-spline	420
Figure B.3 – Compensation du décalage de l'axe vertical.....	421
Figure D.1 – CellLayout d'un CellMotionEngine 8 x 6	430
Tableau 1 – Espaces de noms référencés (avec préfixe)	305
Tableau 2 – Valeurs relatives à l'espace de nom Colourspace	387
Tableau 3 – Description des attributs du type MotionInCells	390
Tableau 4 – Valeur d'en-tête RTP	404
Tableau 5 – Méthodes RTSP	412
Tableau A.1 – Réglages d'en-tête EXI définis par l'ONVIF	419
Tableau A.2 – Réglages de configuration EXI définis par l'ONVIF	419
Tableau C.1 – Paramètres de configuration des règles de l'élément LoiteringDetector	424
Tableau C.2 – Description des champs d'un événement d'errance	424
Tableau C.3 – Paramètres de configuration des règles du détecteur de mouvement déclaratif.....	425
Tableau C.4 – Description des champs d'un événement de mouvement déclaratif	425
Tableau C.5 – Paramètres de configuration de la règle de comptage	426
Tableau C.6 – Description des champs d'un événement de comptage.....	426
Tableau C.7 – Paramètres de configuration de la règle de requête	426
Tableau D.1 – Paramètres de configuration des règles du détecteur de mouvement de cellules	428
Tableau D.2 – Description des champs d'un événement de détection de mouvement de cellules	428

Tableau D.3 – Paramètres de configuration de module	429
Tableau D.4 – Description des champs CellLayout.....	430
Tableau E.1 – Paramètres de configuration des règles du détecteur de zone de mouvement.....	432
Tableau E.2 – Options de configuration des règles du détecteur de zone de mouvement.....	433
Tableau E.3 – Description des champs d'un événement de détecteur de zone de mouvement.....	433

IECNORM.COM : Click to view the full PDF of IEC 62676-2-31:2019

COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

**SYSTÈMES DE VIDÉOSURVEILLANCE DESTINÉS
À ÊTRE UTILISÉS DANS LES APPLICATIONS DE SÉCURITÉ –****Partie 2-31: Transmission en continu en direct
et contrôle basé sur les services web**

AVANT-PROPOS

- 1) La Commission Electrotechnique Internationale (IEC) est une organisation mondiale de normalisation composée de l'ensemble des comités électrotechniques nationaux (Comités nationaux de l'IEC). L'IEC a pour objet de favoriser la coopération internationale pour toutes les questions de normalisation dans les domaines de l'électricité et de l'électronique. À cet effet, l'IEC – entre autres activités – publie des Normes internationales, des Spécifications techniques, des Rapports techniques, des Spécifications accessibles au public (PAS) et des Guides (ci-après dénommés "Publication(s) de l'IEC"). Leur élaboration est confiée à des comités d'études, aux travaux desquels tout Comité national intéressé par le sujet traité peut participer. Les organisations internationales, gouvernementales et non gouvernementales, en liaison avec l'IEC, participent également aux travaux. L'IEC collabore étroitement avec l'Organisation Internationale de Normalisation (ISO), selon des conditions fixées par accord entre les deux organisations.
- 2) Les décisions ou accords officiels de l'IEC concernant les questions techniques représentent, dans la mesure du possible, un accord international sur les sujets étudiés, étant donné que les Comités nationaux de l'IEC intéressés sont représentés dans chaque comité d'études.
- 3) Les Publications de l'IEC se présentent sous la forme de recommandations internationales et sont agréées comme telles par les Comités nationaux de l'IEC. Tous les efforts raisonnables sont entrepris afin que l'IEC s'assure de l'exactitude du contenu technique de ses publications; l'IEC ne peut pas être tenue responsable de l'éventuelle mauvaise utilisation ou interprétation qui en est faite par un quelconque utilisateur final.
- 4) Dans le but d'encourager l'uniformité internationale, les Comités nationaux de l'IEC s'engagent, dans toute la mesure possible, à appliquer de façon transparente les Publications de l'IEC dans leurs publications nationales et régionales. Toutes divergences entre toutes Publications de l'IEC et toutes publications nationales ou régionales correspondantes doivent être indiquées en termes clairs dans ces dernières.
- 5) L'IEC elle-même ne fournit aucune attestation de conformité. Des organismes de certification indépendants fournissent des services d'évaluation de conformité et, dans certains secteurs, accèdent aux marques de conformité de l'IEC. L'IEC n'est responsable d'aucun des services effectués par les organismes de certification indépendants.
- 6) Tous les utilisateurs doivent s'assurer qu'ils sont en possession de la dernière édition de cette publication.
- 7) Aucune responsabilité ne doit être imputée à l'IEC, à ses administrateurs, employés, auxiliaires ou mandataires, y compris ses experts particuliers et les membres de ses comités d'études et des Comités nationaux de l'IEC, pour tout préjudice causé en cas de dommages corporels et matériels, ou de tout autre dommage de quelque nature que ce soit, directe ou indirecte, ou pour supporter les coûts (y compris les frais de justice) et les dépenses découlant de la publication ou de l'utilisation de cette Publication de l'IEC ou de toute autre Publication de l'IEC, ou au crédit qui lui est accordé.
- 8) L'attention est attirée sur les références normatives citées dans cette publication. L'utilisation de publications référencées est obligatoire pour une application correcte de la présente publication.
- 9) L'attention est attirée sur le fait que certains des éléments de la présente Publication de l'IEC peuvent faire l'objet de droits de brevet. L'IEC ne saurait être tenue pour responsable de ne pas avoir identifié de tels droits de brevets et de ne pas avoir signalé leur existence.

La Norme internationale IEC 62676-2-31 a été établie par le comité d'études 79 de l'IEC: Systèmes d'alarme et de sécurité électroniques.

Cette première édition, conjointement avec l'IEC 60839-11-31 et l'IEC 62676-2-32, annule et remplace l'IEC 62676-2-3:2013.

Cette édition inclut les modifications techniques majeures suivantes par rapport à l'IEC 62676-2-3:2013:

- a) ajout du service Media2;
- b) méthodes supplémentaires applicables au service d'imagerie;

- c) suppression des méthodes de duplication du service ES de dispositif;
- d) retrait définitif du service de dispositif d'analyse et du service d'affichage.

Cette publication contient des fichiers de schémas joints. Ces fichiers sont destinés à être utilisés comme complément et ne font pas partie intégrante de la norme.

Le texte de cette Norme internationale est issu des documents suivants:

FDIS	Rapport de vote
79/620/FDIS	79/622/RVD

Le rapport de vote indiqué dans le tableau ci-dessus donne toute information sur le vote ayant abouti à l'approbation de cette Norme internationale.

Ce document a été rédigé selon les Directives ISO/IEC, Partie 2.

Une liste de toutes les parties de la série IEC 62676, publiées sous le titre général *Systèmes de vidéosurveillance destinés à être utilisés dans les applications de sécurité*, peut être consultée sur le site web de l'IEC.

Le comité a décidé que le contenu de ce document ne sera pas modifié avant la date de stabilité indiquée sur le site web de l'IEC sous "<http://webstore.iec.ch>" dans les données relatives au document recherché. À cette date, le document sera

- reconduit,
- supprimé,
- remplacé par une édition révisée, ou
- amendé.

IMPORTANT – Le logo "colour inside" qui se trouve sur la page de couverture de cette publication indique qu'elle contient des couleurs qui sont considérées comme utiles à une bonne compréhension de son contenu. Les utilisateurs devraient, par conséquent, imprimer cette publication en utilisant une imprimante couleur.

INTRODUCTION

L'objet du présent document est d'assurer une mise en œuvre vidéo en réseau avec interopérabilité complète, constituée de produits provenant de différents fournisseurs de vidéo en réseau. Le présent document décrit le modèle vidéo en réseau, les interfaces, les types de données et les schémas d'échange de données. Le document réutilise les normes existantes pertinentes lorsqu'elles sont disponibles et présente de nouvelles spécifications, uniquement lorsque cela est nécessaire pour prendre en charge les exigences spécifiques de la vidéosurveillance en réseau.

IECNORM.COM : Click to view the full PDF of IEC 62676-2-31:2019

SYSTÈMES DE VIDÉOSURVEILLANCE DESTINÉS À ÊTRE UTILISÉS DANS LES APPLICATIONS DE SÉCURITÉ –

Partie 2-31: Transmission en continu en direct et contrôle basé sur les services web

1 Domaine d'application

Le présent document définit les procédures de communication entre les clients vidéo en réseau et les dispositifs de transmission vidéo. Ce nouvel ensemble de spécifications permet de concevoir des systèmes vidéo en réseau avec des dispositifs et des récepteurs de différents fabricants utilisant des interfaces communes et bien définies. Ces interfaces couvrent des fonctions telles que la configuration multimédia et d'imagerie, la transmission en continu en temps réel de contenu audio et vidéo, le contrôle panoramique, inclinaison et zoom (PTZ – *Pan, Tilt and Zoom*) et l'analyse.

Les interfaces de gestion et de contrôle définies dans le présent document sont décrites sous forme de services Web. L'Annexe F contient les définitions du schéma XML et du langage de description de services web (WSDL – *Web Service Description Language*) pour les services de réseaux présentés.

2 Références normatives

Les documents suivants sont cités dans le texte de sorte qu'ils constituent, pour tout ou partie de leur contenu, des exigences du présent document. Pour les références datées, seule l'édition citée s'applique. Pour les références non datées, la dernière édition du document de référence s'applique (y compris les éventuels amendements).

IEC 60839-11-31, *Systèmes d'alarme et de sécurité électroniques – Partie 11-31: Systèmes de contrôle d'accès électronique – Protocole de base d'interopérabilité en fonction des services Web*

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

INTERNET ENGINEERING TASK FORCE (IETF). RFC 1952: *GZIP file format specification version 4.3* [en ligne]. Édité par P. Deutsch. Mai 1996 [consulté le 2019-01-08]. Disponible à l'adresse <http://tools.ietf.org/html/rfc1952>

INTERNET ENGINEERING TASK FORCE (IETF). RFC 2326: *Real Time Streaming Protocol (RTSP)* [en ligne]. Édité par H. Schulzrinne et al. Avril 1998 [consulté le 2019-01-08]. Disponible à l'adresse <http://www.ietf.org/rfc/rfc2326.txt>

INTERNET ENGINEERING TASK FORCE (IETF). RFC 2435: *RTP Payload Format for JPEG-compressed Video* [en ligne]. Édité par L. Berc et al. Octobre 1998 [consulté le 2019-01-08]. Disponible à l'adresse <http://www.ietf.org/rfc/rfc2435.txt>

INTERNET ENGINEERING TASK FORCE (IETF). RFC 2818: *HTTP over TLS* [en ligne]. Édité par E. Rescorla. Mai 2000 [consulté le 2019-01-08]. Disponible à l'adresse <http://www.ietf.org/rfc/rfc2818.txt>

INTERNET ENGINEERING TASK FORCE (IETF), RFC 3016: *RTP Payload Format for MPEG-4 Audio/Visual Streams* [en ligne]. Édité par Y. Kikuchi et al. Novembre 2000 [consulté le 2019-01-08]. Disponible à l'adresse <http://www.ietf.org/rfc/rfc3016.txt>

INTERNET ENGINEERING TASK FORCE (IETF), RFC 3550: *RTP: A Transport Protocol for Real-Time Applications* [en ligne]. Édité par H. Schulzrinne et al. Juillet 2003 [consulté le 2019-01-08]. Disponible à l'adresse <http://www.ietf.org/rfc/rfc3550.txt>

INTERNET ENGINEERING TASK FORCE (IETF), RFC 3551: *RTP Profile for Audio and Video Conferences with Minimal Control* [en ligne]. Édité par H. Schulzrinne et al. Juillet 2003 [consulté le 2019-01-08]. Disponible à l'adresse <http://www.ietf.org/rfc/rfc3551.txt>

INTERNET ENGINEERING TASK FORCE (IETF), RFC 3640: *RTP Payload Format for Transport of MPEG-4 Elementary Streams* [en ligne]. Édité par J. van der Meer et al. Novembre 2003 [consulté le 2019-01-08]. Disponible à l'adresse <http://www.ietf.org/rfc/rfc3640.txt>

INTERNET ENGINEERING TASK FORCE (IETF), RFC 3984: *RTP Payload Format for H.264 Video* [en ligne]. Édité par T. Stockhammer et al. Février 2005 [consulté le 2019-01-08]. Disponible à l'adresse <http://www.ietf.org/rfc/rfc3984>

INTERNET ENGINEERING TASK FORCE (IETF), RFC 4566: *SDP: Session Description Protocol* [en ligne]. Édité par M. Handley et al. Juillet 2006 [consulté le 2019-01-08]. Disponible à l'adresse <http://www.ietf.org/rfc/rfc4566.txt>

INTERNET ENGINEERING TASK FORCE (IETF), RFC 6455: *The WebSocket Protocol* [en ligne]. Édité par I. Fette et al. Décembre 2011 [consulté le 2019-01-08]. Disponible à l'adresse <http://www.ietf.org/rfc/rfc6455.txt>

INTERNET ENGINEERING TASK FORCE (IETF), RFC 7798: *RTP Payload Format for High Efficiency Video Coding (HEVC)* [en ligne]. Édité par Y.-K. Wang et al. Mars 2016 [consulté le 2019-01-08]. Disponible à l'adresse <http://www.ietf.org/rfc/rfc7798.txt>

Apple, *Tunneling QuickTime RTSP and RTP over HTTP*

Disponible à l'adresse

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

3 Termes et définitions

Pour les besoins du présent document, les termes et définitions suivants s'appliquent.

L'ISO et l'IEC tiennent à jour des bases de données terminologiques destinées à être utilisées en normalisation, consultables aux adresses suivantes:

- IEC Electropedia: disponible à l'adresse <http://www.electropedia.org/>
- ISO Online browsing platform: disponible à l'adresse <http://www.iso.org/obp>

¹ QuickTime est l'appellation commerciale d'un produit distribué par Apple Inc. Cette information est donnée à l'intention des utilisateurs du présent document et ne signifie nullement que l'IEC approuve ou recommande l'emploi exclusif du produit ainsi désigné. Des produits équivalents peuvent être utilisés s'il est démontré qu'ils conduisent aux mêmes résultats.

3.1

profil multimédia

profil qui mappe une source vidéo ou audio, ou une sortie audio à un codeur vidéo ou audio, une configuration de décodeur audio et des configurations PTZ et analyse

3.2

métadonnées

ensemble des données de transmission en continu, sauf la vidéo et l'audio, comprenant les résultats d'analyse vidéo, les données de position PTZ et d'autres métadonnées (telles que des données textuelles provenant d'applications POS)

3.3

nœud PTZ

entité PTZ de niveau inférieur avec mappage pour le dispositif PTZ et ses fonctionnalités

3.4

analyse

algorithmes ou programmes utilisés pour analyser les données vidéo et audio afin de générer une description des données de niveau supérieur

4 Présentation

4.1 Généralités

La présente spécification définit une interface normalisée applicable aux caméras de vidéosurveillance et aux codeurs en fonction du mécanisme de communication de services Web de base défini dans l'IEC 60839-11-31. Les services présentés dans cette spécification définissent les informations détaillées techniques d'interface pour les dispositifs et les clients afin d'assurer la conformité aux profils S et T ONVIF.

4.2 Dispositif ES

La présente spécification étend les définitions du dispositif ES tirées de l'IEC 60839-11-31 par l'ajout d'interfaces destinées à extraire les entités suivantes:

- VideoOutputs
- VideoSources
- AudioOutputs
- AudioSources

4.3 Configuration d'imagerie

Le service d'imagerie fournit les données de configuration et de contrôle pour les propriétés spécifiques à l'imagerie. WSDL fait partie du cadre d'application et est fourni dans le fichier WSDL d'imagerie.

Le service comprend les opérations suivantes:

- Obtenir et définir les configurations d'imagerie telles que le temps d'exposition, le gain et l'équilibre des blancs.
- Obtenir les options de configuration d'imagerie qui comprennent des plages valides pour les paramètres d'imagerie.
- Déplacer la lentille de mise au point;
- Obtenir la position actuelle et modifier le statut d'une lentille de mise au point.
- Récupérer et configurer les préséglages d'imagerie.

4.4 Configuration multimédia

4.4.1 Profils multimédias

Les configurations multimédias sont traitées par l'intermédiaire du service multimédia. Les configurations multimédias servent à déterminer les propriétés de transmission en continu des flux multimédias demandés tels que définis dans la présente spécification. Les ensembles de configuration multimédias sont appelés profils multimédias. Voir la Figure 1 pour un exemple abstrait.

Les configurations de transmission vidéo et audio en continu en temps réel sont contrôlées à l'aide de profils multimédias. Un profil multimédia mappe une source vidéo et/ou audio à un codeur vidéo et/ou audio, des configurations PTZ et Analyse. Un dispositif satisfaisant à l'ONVIF et prenant en charge le service multimédia présente différents profils disponibles en fonction de ses fonctionnalités (l'ensemble de profils disponibles peut toutefois varier de manière dynamique).

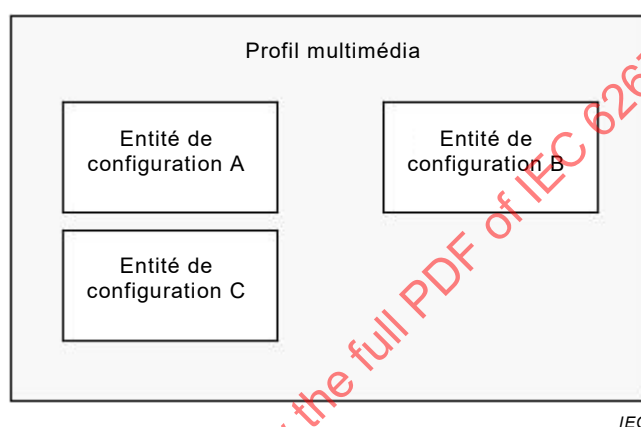


Figure 1 – Profil multimédia

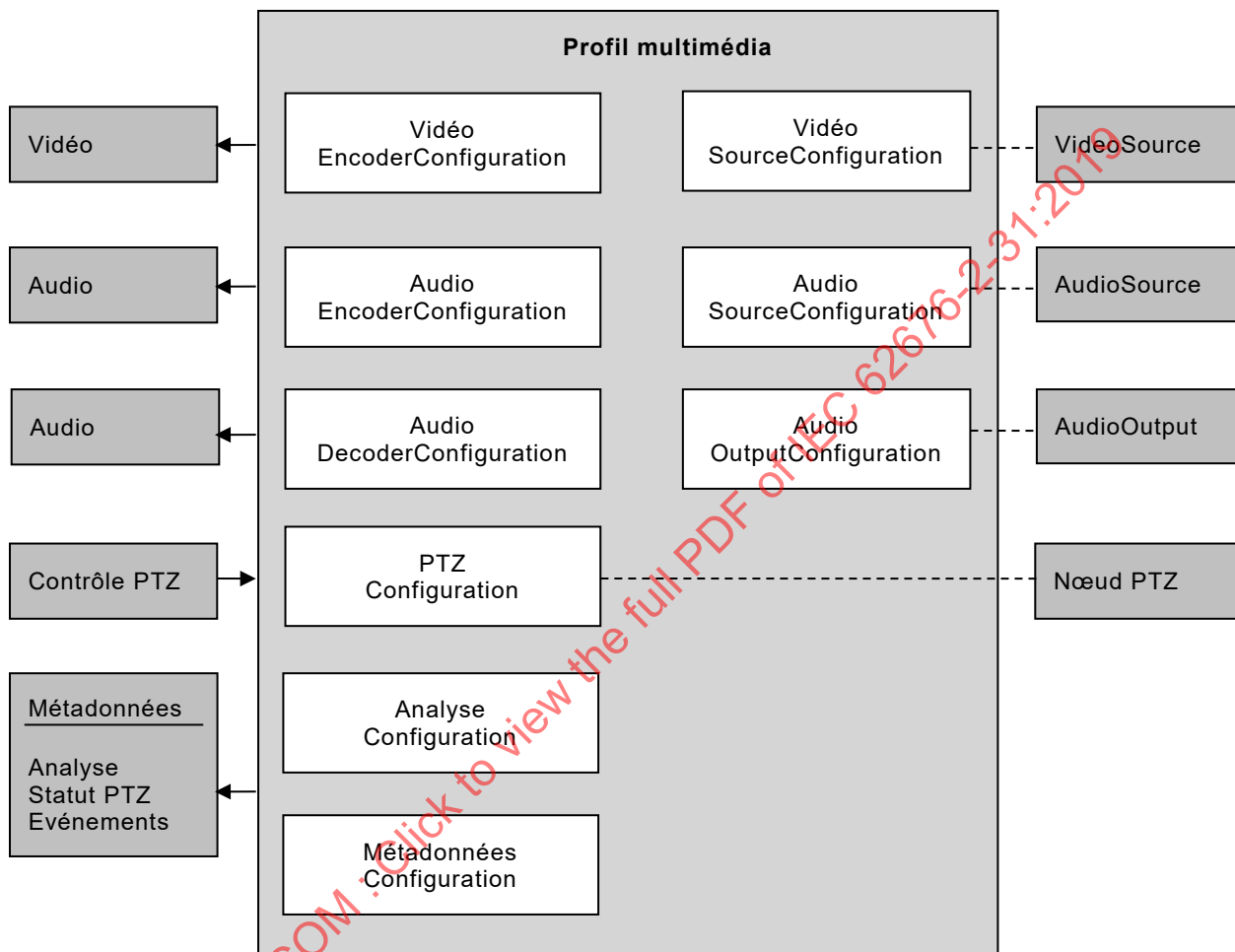
Un profil multimédia est composé d'un ensemble d'entités de configuration interconnectées comme représenté à la Figure 1. Les configurations sont fournies par le dispositif et peuvent être statiques ou créées de manière dynamique par celui-ci. Par exemple, les configurations dynamiques peuvent être créées par le dispositif en fonction des ressources de codage disponibles actuelles. Une entité de configuration appartient à l'un des types suivants:

- configuration de source vidéo;
- configuration de source audio;
- configuration de codeur vidéo;
- configuration de codeur audio;
- configuration PTZ;
- configuration d'analyse;
- configuration de métadonnées;
- configuration de sortie audio;
- configuration de décodeur audio.

Un profil est composé de l'ensemble ou d'un sous-ensemble de ces entités de configuration. Suivant les fonctionnalités du dispositif, une entité de configuration particulière peut faire partie ou non d'un profil. Par exemple, un profil avec une source audio et une configuration de codeur audio peut exister uniquement dans un dispositif avec support audio.

Un dispositif peut fournir des profils "prêts à l'emploi" pour les configurations multimédias les plus courantes qu'il propose. Le profil contient un attribut "fixe" qui indique si un profil peut être supprimé ou non. L'attribut fixe n'indique pas si un profil est immuable. De fait, des configurations doivent pouvoir être ajoutées à ou supprimées d'un profil fixe. Le caractère fixe ou pas d'un profil est défini par le dispositif.

Un exemple de configuration de profil complète est représenté à la Figure 2.



IEC

Figure 2 – Configuration de profil complète

Un profil multimédia décrit les éléments à présenter au client dans un flux multimédia, et comment les présenter, ainsi que la manière de traiter les entrées PTZ et analyse.

Les commandes suivantes gèrent les profils multimédias:

- CreateProfile – Crée un nouveau profil multimédia.
- GetProfiles – Obtient un ou tous les profils multimédias existants.
- DeleteProfile – Supprime un profil multimédia spécifique.
- AddConfiguration – Ajoute ou remplace les entités de configuration d'un profil multimédia.
- RemoveConfiguration – Supprime une ou plusieurs entités de configuration d'un profil multimédia.

Les commandes suivantes gèrent les entités de configuration:

- `Get<entity>ConfigurationOptions` – Obtient les valeurs de propriété valides pour une entité de configuration spécifique.
- `Get<entity>Configurations` – Obtient une ou plusieurs entités de configuration. Le client peut demander une configuration spécifique en fournissant le jeton de configuration. Il peut également spécifier un jeton de profil pour obtenir toutes les configurations compatibles. L'absence de spécification de jeton entraîne le renvoi de toutes les configurations existantes.
- `Set<entity>Configuration` – Met à jour les réglages d'une entité de configuration.

Où `<entity>` est le type d'entité de configuration. Par exemple, la commande complète permettant d'obtenir une configuration de codeur vidéo est `GetVideoEncoderConfiguration`.

Les commandes suivantes initient et manipulent un flux vidéo/audio:

- `GetStreamUri` – Demande un URI de transmission en continu valide pour un profil multimédia et un protocole spécifiques.
- `StartMulticastStreaming` – Commence la transmission en continu en multidiffusion à l'aide d'un profil multimédia spécifié.
- `StopMulticastStreaming` – Interrompt un flux de multidiffusion.
- `SetSynchronizationPoint` – Insère un point de synchronisation (trame I, etc.) dans les flux actifs.
- `GetSnapshotUri` – Demande un URI pour un profil multimédia spécifique qui peut être utilisé pour obtenir des instantanés JPEG.

4.4.2 Mode de source vidéo

Un dispositif peut prendre en charge la modification du mode de source vidéo. Une telle modification exige généralement un redémarrage du dispositif en raison de dépendances matérielles. Par exemple, la sélection de chaque mode de source vidéo permet de modifier la fonctionnalité de résolution maximale (1920 x 1080 @ 16:9 ou 2048 x 1536 @ 4:3) et de taux de trame (20 fps ou 30 fps) du dispositif.

Les commandes suivantes gèrent le mode de source vidéo:

- `GetVideoSourceModes` – Obtenir une liste des modes de source vidéo.
- `SetVideoSourceMode` – Définir le mode de source vidéo sur le mode spécifié.

4.5 Transmission en continu en temps réel

Le présent document définit les options et les formats de transmission en continu multimédia. Une distinction est faite entre plan multimédia et plan de contrôle, comme représenté à la Figure 3. Un ensemble d'options de transmission en continu multimédia (données audio, données vidéo et métadonnées), toutes basées sur le protocole RTP [RFC 3550], est décrit afin d'assurer l'interopérabilité des services de transmission en continu multimédia.

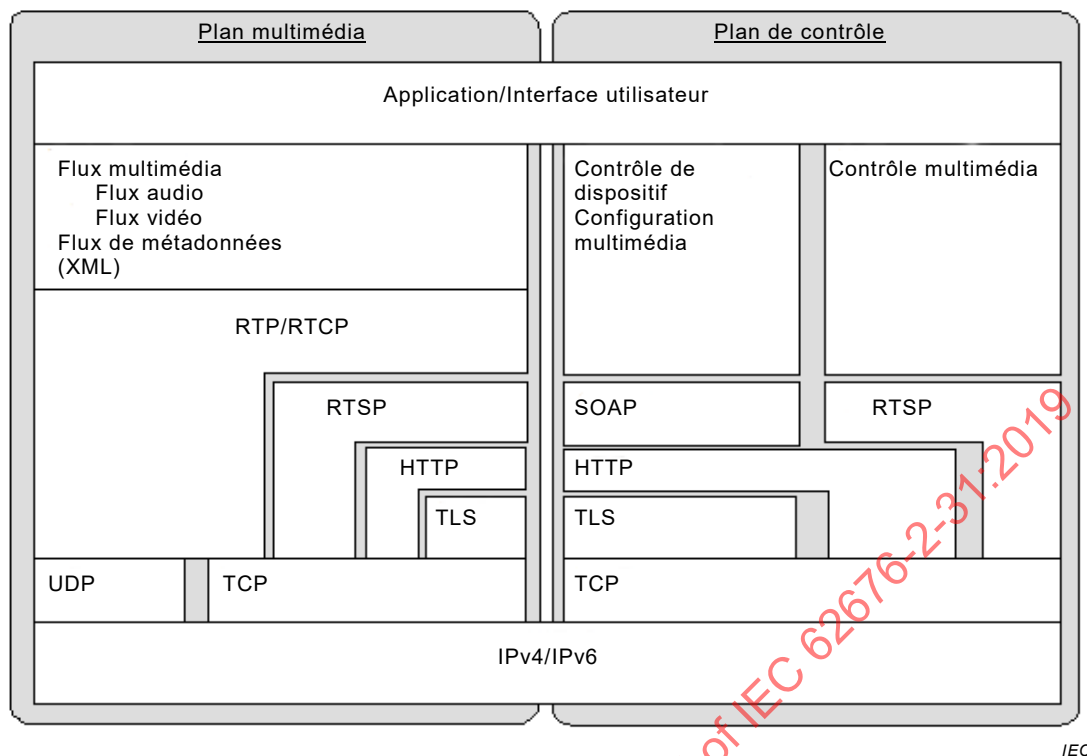


Figure 3 – Structure de couches

Le format de transmission en continu de métadonnées permet une transmission en temps réel bien définie des données analytiques, de statut PTZ et de notification.

La configuration multimédia est exécutée par le protocole SOAP/HTTP et est couverte par le service de configuration multimédia explicité en 4.6.

Le contrôle multimédia est assuré par le protocole RTSP tel que défini dans RFC 2326. Le présent document utilise le profilage RTP, RTCP et RTSP, ainsi que JPEG sur des extensions RTP et des mécanismes de contrôle multidiffusion.

Le présent document introduit des extensions à la norme RFC 2326 pour permettre les connexions de transmission en continu bidirectionnelles.

Les configurations de transmission en continu des codecs vidéo suivants sont proposées:

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

et pour les codecs audio suivants:

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

4.6 Contrôle PTZ

Le modèle PTZ regroupe les mouvements possibles de l'unité PTZ dans une composante panoramique/inclinaison et une composante Zoom. Pour diriger l'unité PTZ, le service permet d'effectuer des opérations de déplacement absolu, déplacement relatif et déplacement continu. Différents systèmes de coordonnées et unités sont utilisés pour effectuer ces opérations.

Le service PTZ permet d'effectuer une opération `AbsoluteMove` pour déplacer le dispositif PTZ vers une position absolue. Le service attend la position absolue sous la forme d'un argument faisant référence à un système de coordonnées absolu. La vitesse du mouvement panoramique/inclinaison et du mouvement zoom peut être éventuellement spécifiée. Les valeurs de vitesse sont des scalaires positifs et ne contiennent pas d'information directionnelle. Les vitesses du mouvement panoramique/inclinaison ne peuvent pas être spécifiées séparément sans connaître la position actuelle. Cette approche de spécification d'une position souhaitée produit généralement une action qui n'est ni fluide ni intuitive.

Une opération `RelativeMove` est introduite par le service PTZ afin de diriger le dôme par rapport à la position actuelle, mais sans qu'il soit nécessaire de connaître cette dernière. L'opération attend une translation de position sous la forme d'un argument faisant référence à un système de coordonnées relatif. La présente spécification différencie les systèmes de coordonnées relatif et absolu, étant donné que dans certains cas, il n'existe pas de système de coordonnées absolu pour un système de coordonnées relatif bien défini. Un argument de vitesse facultatif peut être ajouté à l'opération `RelativeMove` avec la même définition que pour l'opération `AbsoluteMove`.

Enfin, le dispositif PTZ peut être déplacé en continu à l'aide de la commande `ContinuousMove` dans une certaine direction et selon une certaine vitesse. Ainsi, un vecteur de vélocité représente à la fois les informations de direction et de vitesse. Cette dernière est exprimée par la longueur du vecteur.

Les coordonnées de panoramique/inclinaison et de zoom peuvent être spécifiées de manière unique en ajoutant aux coordonnées les URI spatiaux appropriés. Un URI spatial représente de manière unique le système de coordonnées sous-jacent. Le paragraphe 8.8 définit un ensemble normalisé de systèmes de coordonnées. Un nœud PTZ doit mettre en œuvre ces systèmes de coordonnées si le type de mouvement correspondant est pris en charge par le nœud PTZ. Dans la plupart des cas, la position de panoramique/inclinaison est représentée par des angles de panoramique/inclinaison dans un système de coordonnées sphérique. Un PTZ numérique, fonctionnant sur une caméra mégapixel fixe, peut représenter la direction de visée de la caméra par une position de pixel sur un plan de projection statique. Par conséquent, différents systèmes de coordonnées sont nécessaires dans ce cas pour capturer les mouvements physiques ou virtuels du dispositif PTZ. Le nœud PTZ peut éventuellement définir ses propres systèmes de coordonnées spécifiques au dispositif afin de permettre aux clients de tirer profit des propriétés spécifiques du nœud PTZ en question.

La description du nœud PTZ extraite par l'opération `GetNode` ou `GetNodes` contient tous les systèmes de coordonnées pris en charge par un nœud PTZ spécifique. Chaque système de coordonnées appartient à l'un des groupes suivants:

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

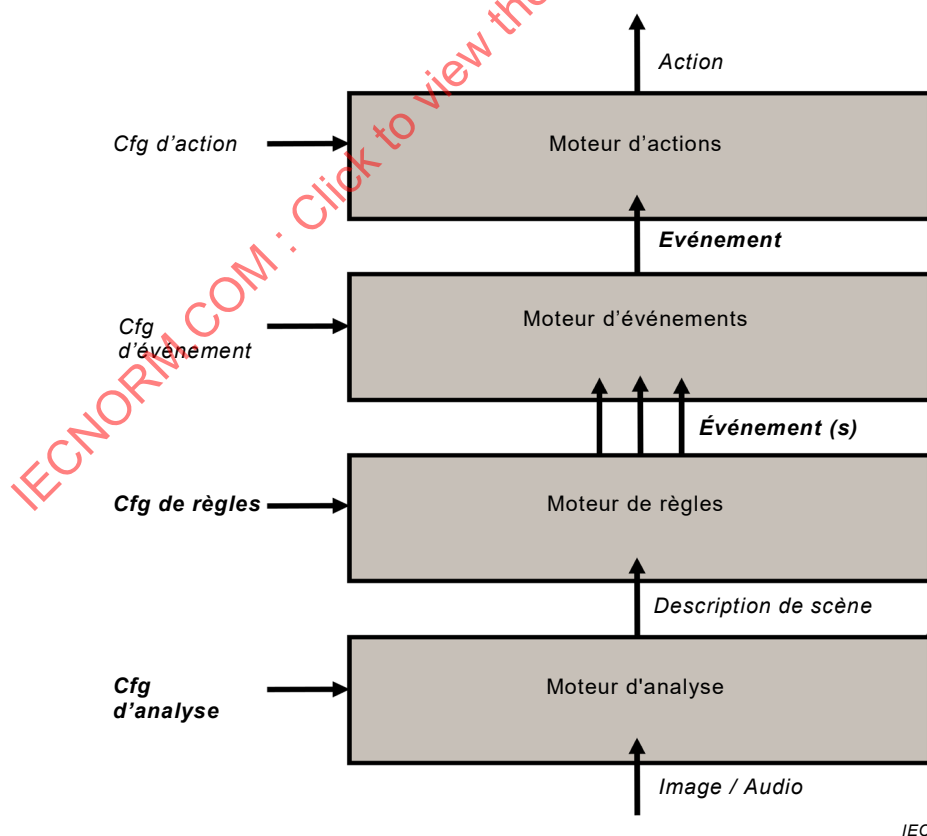
Si le nœud PTZ ne prend pas en charge les systèmes de coordonnées d'un certain groupe, l'opération de déplacement correspondante n'est pas disponible pour ce nœud PTZ. Par exemple, si la liste ne contient pas un groupe AbsolutePanTiltPositionSpace, l'opération AbsoluteMove doit échouer lorsqu'une position de panoramique/inclinaison absolue est spécifiée. La section de commande correspondante décrit les espaces exigés pour une commande de déplacement spécifique.

4.7 Analyse

Les applications d'analyse sont divisées en parties d'analyse d'image/audio et spécifiques à l'application. L'interface entre ces deux parties produit une abstraction qui décrit la scène sur la base des objets présents. La partie spécifique à l'application compare les descriptions et les règles de la scène (comme les lignes virtuelles qu'il est interdit de franchir ou les polygones qui définissent une zone protégée). D'autres règles peuvent représenter un comportement à l'intérieur des objets, comme des objets suivant d'autres objets (pour former une détection par talonnage). Ces règles peuvent également permettre de décrire un mouvement d'objet interdit, qui peut être utilisé pour établir une limite de vitesse.

Ces deux parties séparées, appelées "moteur d'analyse vidéo/audio" et "moteur de règles", forment avec les événements et les actions, l'architecture d'analyse selon la présente spécification, comme représenté à la Figure 4.

L'architecture d'analyse est composée d'éléments et d'interfaces. Chaque élément donne une fonctionnalité correspondant à une entité unique d'un point de vue sémantique de la solution complète d'analyse vidéo/audio. Les interfaces sont unidirectionnelles et définissent une entité d'information avec un contenu unique. Seules les interfaces sont soumises à cette spécification. Le point central de cette architecture est la capacité à répartir les éléments ou ensembles d'éléments adjacents vers un dispositif du réseau.



IEC

Figure 4 – Architecture d'analyse

Les interfaces suivantes sont définies dans le présent document:

- interface de configuration d'analyse;
- description de scène;
- interface de configuration de règles.

La présente spécification définit un cadre de configuration pour le moteur d'analyse appelé Service d'analyse ONVIF. Ce cadre permet à un client de contacter un dispositif qui met en oeuvre le service d'analyse ONVIF pour les modules d'analyse pris en charge et leurs configurations. Les configurations de ces modules peuvent être ajoutées, supprimées ou modifiées de manière dynamique par un client, lui permettant d'exécuter plusieurs modules d'analyse en parallèle, s'ils sont pris en charge par le dispositif.

La sortie du moteur d'analyse est appelée description de scène. La description de scène représente l'abstraction de la scène en matière de métadonnées pour les objets, statiques ou dynamiques, qui font partie de la scène. La présente spécification définit une interface de description de scène XML comprenant les types de données.

Les règles décrivent la manière d'interpréter la description de scène et de réagir à cette information. La spécification définit la syntaxe de règle et les méthodes normalisées pour communiquer la configuration de ces règles de l'application au dispositif.

Un dispositif qui prend en charge le service d'analyse ONVIF doit mettre en oeuvre l'interface de description de scène et permettre l'expédition des événements à l'aide du service d'événements. Si le dispositif prend également en charge un moteur de règles, il doit alors mettre en oeuvre l'interface de modules analytiques et de règles.

Les interfaces et la configuration du moteur d'événements et d'actions ne relèvent pas du domaine d'application de la présente spécification. L'interface d'événements est gérée par l'intermédiaire du service d'événements décrit dans l'IEC 60839-11-31.

Une configuration d'analyse vidéo peut être jointe à un profil multimédia si le service multimédia ONVIF existe. Dans ce cas, la configuration d'analyse vidéo est connectée à une source vidéo spécifique.

4.8 Interfaces

Le Tableau 1 répertorie les préfixes et espaces de noms utilisés dans la présente spécification. Le tableau donne les articles de l'Annexe F qui correspondent aux interfaces définies par le présent document. Les préfixes répertoriés ne font pas partie du présent document, et une mise en oeuvre peut utiliser le préfixe de son choix.

Tableau 1 – Espaces de noms référencés (avec préfixe)

Préfixe	URI d'espace de nom	Référence
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	Article F.7
tmd	http://www.onvif.org/ver10/deviceIO/wsdl	Article F.1
tr2	http://www.onvif.org/ver20/media/wsdl	Article F.4
timg	http://www.onvif.org/ver20/imaging/wsdl	Article F.2
tptz	http://www.onvif.org/ver20/ptz/wsdl	Article F.5
axt	http://www.onvif.org/ver20/analytics	Article F.6

5 Service ES de dispositif

5.1 Généralités

Ce service propose des commandes permettant d'extraire et de configurer les entrées et sorties physiques d'un dispositif.

Un dispositif comportant des sources et des sorties physiques doit prendre en charge ce service. Pour la définition du schéma, se reporter à l'Article F.1.

5.2 VideoOutputs

5.2.1 Généralités

Le type VideoOutput représente les sorties vidéo physiques d'un dispositif qui peuvent être connectées à un moniteur pour afficher le signal vidéo. La structure contient les réglages de présentation qui peuvent être configurés à l'aide du service d'affichage.

5.2.2 GetVideoOutputs

Cette commande répertorie toutes les sorties vidéo disponibles d'un dispositif. Un dispositif comportant une ou plusieurs sorties vidéo physiques doit prendre en charge la liste des sorties vidéo disponibles au moyen de la commande GetVideoOutputs.

DEMANDE:

Ceci est un message vide.

RÉPONSE:

- **Token – facultatif, non limité [tt:ReferenceToken]**
Liste des jetons énumérant toutes les sorties vidéo disponibles du dispositif.

DÉFAUTS:

Aucun

CLASSE D'ACCÈS:

READ_MEDIA

5.3 VideoOutputConfiguration

5.3.1 GetVideoOutputConfiguration

Cette opération demande la configuration d'une sortie vidéo. Un dispositif comportant une ou plusieurs sorties vidéo doit prendre en charge l'extraction de VideoOutputConfiguration au moyen de cette commande.

DEMANDE:

- **VideoOutputToken [tt:ReferenceToken]**
Référence existante d'une sortie vidéo.

RÉPONSE:

- **Configuration [tt:VideoOutputConfiguration]**
Configuration de la sortie vidéo.

DÉFAUTS:

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

La sortie vidéo demandée n'existe pas.

CLASSE D'ACCÈS:

READ_MEDIA**5.3.2 SetVideoOutputConfiguration**

Cette opération modifie une configuration de sortie vidéo. Un dispositif comportant une ou plusieurs sorties vidéo doit prendre en charge le réglage de sa configuration de sortie vidéo au moyen de cette commande.

DEMANDE:

- **Configuration [tt:VideoOutputConfiguration]**
Configuration de la sortie vidéo.
- **ForcePersistence [xs:boolean]**
Modificateur déconseillé des réglages temporaires si prise en charge par le dispositif.

RÉPONSE:

Ceci est un message vide.

DÉFAUTS:

- **env:Sender – ter:InvalidArgVal – ter:NoVideoOutput**
La sortie vidéo demandée n'existe pas.
- **env:Sender – ter:InvalidArgVal – ter:ConfigModify**
Les paramètres de configuration ne peuvent pas être définis.

CLASSE D'ACCÈS:

ACTUATE**5.3.3 GetVideoOutputConfigurationOptions**

Cette opération demande les VideoOutputConfigurationOptions d'un VideoOutput. Un dispositif comportant une ou plusieurs sorties vidéo doit prendre en charge l'extraction de VideoOutputConfigurationOptions au moyen de cette commande.

DEMANDE:

- **VideoOutputToken [tt:ReferenceToken]**
Référence existante d'une sortie vidéo.

RÉPONSE:

- **VideoOutputOptions [tt:VideoOutputConfigurationOptions]**
Options de configuration de la sortie vidéo.

DÉFAUTS:

- **env:Sender – ter:InvalidArgVal – ter:NoVideoOutput**
La sortie vidéo demandée n'existe pas.

5.4 VideoSources

5.4.1 Généralités

Une VideoSource représente une entrée vidéo physique. La structure contient la résolution en pixels de la vidéo, le taux de trame et les réglages d'imagerie. Les réglages d'imagerie peuvent être manipulés par l'intermédiaire de la commande ImagingService, si elle est prise en charge et contient des paramètres de mise au point, d'exposition et de luminosité, par exemple.

5.4.2 GetVideoSources

Cette opération répertorie toutes les sources vidéo disponibles pour le dispositif. Le dispositif qui comporte une ou plusieurs entrées vidéo doit prendre en charge la liste des sources vidéo disponibles au moyen de la commande GetVideoSources.

DEMANDE:

Ceci est un message vide.

RÉPONSE:

- **Token – facultatif, non limité [tt:ReferenceToken]**

Liste des jetons énumérant toutes les sources vidéo disponibles du dispositif.

DÉFAUTS:

Aucun

CLASSE D'ACCÈS:

READ_MEDIA

5.5 AudioOutputs

5.5.1 Généralités

Une AudioOutput représente les sorties audio physiques qui peuvent être connectées à un haut-parleur.

5.5.2 GetAudioOutputs

Cette commande répertorie toutes les sorties audio disponibles d'un dispositif. Un dispositif comportant une ou plusieurs sorties audio physiques doit prendre en charge la liste des sorties audio disponibles au moyen de la commande GetAudioOutputs.

DEMANDE:

Ceci est un message vide.

RÉPONSE:

- **Token – facultatif, non limité [tt:ReferenceToken]**

Liste des jetons énumérant toutes les sorties audio disponibles du dispositif.

DÉFAUTS:

Aucun

CLASSE D'ACCÈS:

READ_MEDIA

5.6 AudioSources

5.6.1 Généralités

Une AudioSource représente une entrée audio non codée et déclare le nombre de voies d'entrée.

5.6.2 GetAudioSources

Cette opération répertorie toutes les sources audio disponibles pour le dispositif. Le dispositif qui comporte une ou plusieurs sources audio doit prendre en charge la liste des entrées audio disponibles au moyen de la commande GetAudioSources.

DEMANDE:

Ceci est un message vide.

RÉPONSE:

- **Token – facultatif, non limité [tt:ReferenceToken]**

Liste des jetons énumérant toutes les sources audio disponibles du dispositif.

DÉFAUTS:

Aucun

CLASSE D'ACCÈS:

READ_MEDIA

5.7 Fonctionnalités

Les fonctionnalités reflètent les fonctions facultatives et la fonctionnalité d'un service. Les informations sont statiques et ne varient pas pendant le fonctionnement du dispositif. Les fonctionnalités suivantes sont disponibles:

VideoSources:	nombre de sources vidéo (par défaut aucune).
VideoOutputs:	nombre de sorties vidéo (par défaut aucune).
AudioSources:	nombre de sources audio (par défaut aucune).
AudioOutputs:	nombre de sorties audio (par défaut aucune).
RelayOutputs:	nombre de sorties de relais (par défaut aucune).
DigitalInputs:	nombre d'entrées numériques (par défaut aucune).
SerialPorts:	nombre de ports série (par défaut aucune).
DigitalInputOptions:	Indique la prise en charge de la configuration DigitalInput de l'état de veille (par défaut définie sur 'false')

DEMANDE:

Ceci est un message vide.

RÉPONSE:

- **Capabilities [tmd:Capabilities]**

Liste des fonctionnalités définies ci-dessus.

DÉFAUTS:

Aucun

CLASSE D'ACCÈS:

PRE_AUTH

6 Service multimédia

6.1 Généralités

Le service multimédia est utilisé pour configurer les propriétés de transmission en continu multimédia du dispositif.

Le service multimédia permet à un client de procéder à des configurations de transmission en continu multimédia et toute autre transmission en continu en temps réel. Les configurations multimédias sont traitées par l'intermédiaire de profils multimédias. Une vue d'ensemble du modèle de configuration multimédia de l'ONVIF est présentée en 4.4.1.

Les commandes de service multimédia sont divisées en deux catégories principales:

- Configuration multimédia:
 - commandes de profil multimédia,
 - commandes de source vidéo,
 - commandes de codeur vidéo,
 - commandes de source audio,
 - commandes de codeur audio,
 - commandes d'analyse vidéo,
 - commandes de métadonnées,
 - commandes de sortie audio,
 - commandes de décodeur audio.
- Transmission en continu multimédia:
 - demande d'URI de flux,
 - obtention d'URI d'instantané,
 - commandes de contrôle de multidiffusion,
 - point de synchronisation multimédia.

Un ensemble basique d'opérations est exigé pour le service multimédia. Il est recommandé de prendre en charge d'autres opérations. Les exigences détaillées sont répertoriées sous les descriptions de commande.

Les dispositions de 6.2 à 6.13 décrivent le service Media2. Pour le schéma associé, se reporter aux définitions de l'Article F.4. Pour les écarts du service Media1, voir 6.14.

6.2 Méthodes de profil multimédia

6.2.1 Création de profil multimédia

Cette opération crée un nouveau profil multimédia. Le profil multimédia doit être créé dans le dispositif.

Un dispositif qui met en oeuvre ce service doit prendre en charge la création des profils multimédias tant que le nombre de profils existants ne dépasse pas la valeur de fonctionnalité `MaximumNumberOfProfiles`.

Un profil créé doit pouvoir être supprimé et la valeur "false" doit être attribuée à l'attribut "fixe" d'un dispositif dans le profil retourné.

DEMANDE:

- **Name (Nom) [tt:Name]**
Nom du nouveau profil.
- **Configuration – facultatif, non limité [tr2:ConfigurationRef]**
Liste facultative des configurations à ajouter au nouveau profil.

RÉPONSE:

- **Token [tt:ReferenceToken]**
Jeton attribué au profil nouvellement créé.

DÉFAUTS:

- **env:Receiver – ter:Action – ter:MaxNVTPProfiles**
Le nombre maximal de profils pris en charge par le dispositif a été atteint.
- **env:Receiver – ter:Action – ter:ConfigurationConflict**
D'autres configurations du profil multimédia sont en conflit avec celle à ajouter et son ajout génère un conflit de profil multimédia.
- **env:Sender – ter:InvalidArgVal – ter:NoConfig**
Une des configurations fournies indiquée par le ConfigurationToken n'existe pas.

CLASSE D'ACCÈS:

ACTUATE**6.2.2 Obtention de profils multimédias**

Un point terminal peut demander les profils multimédias existants d'un dispositif à l'aide de la commande GetProfiles. Cette commande permet d'extraire à la fois les profils préconfigurés et les profils créés de manière dynamique.

Le paramètre Jeton contrôle les profils retournés:

- En l'absence de ce paramètre, cette commande répertorie tous les profils configurés d'un dispositif.
- Si ce paramètre est fourni, la commande soit répertorie le profil référencé, soit répond par la transmission d'une erreur.

Le paramètre Type contrôle les configurations retournées et n'influe pas sur le nombre de profils retournés:

- En l'absence de ce paramètre, les profils retournés ne doivent contenir aucune information de configuration.
- Si un paramètre Type unique avec la valeur 'All' est fourni, les profils retournés doivent inclure toutes les configurations associées.
- À défaut, la liste des configurations demandée doit inclure pour chaque profil les configurations présentes sous la forme d'un paramètre Type.

Un dispositif qui met en oeuvre ce service doit prendre en charge l'extraction de profils multimédias au moyen de la commande GetProfiles.

DEMANDE:

- **Token – facultatif [tt:ReferenceToken]**
Jeton facultatif destiné à extraire exactement un profil.
- **Type – facultatif, non limité [xs:string]**
En cas de transmission d'un ou de plusieurs types, seules les configurations correspondantes sont retournées.

RÉPONSE:

- **Profils – facultatif, non limité [tr2:MediaProfile]**
Liste des profils. Chaque profil contient un ensemble d'entités de configuration définissant une configuration spécifique qui peut être utilisée pour la transmission en continu multimédia, l'analyse, la transmission en continu de métadonnées, etc.

DÉFAUTS:

- **env:Sender – ter:InvalidArgVal – ter:NoProfile**
Le jeton de profil demandé n'existe pas.

CLASSE D'ACCÈS:

READ_MEDIA

NOTE Un client peut demander une énumération de profils sans aucune information détaillée de configuration, en ne transmettant pas de paramètre Type.

6.2.3 Ajout d'une ou de plusieurs configurations à un profil

Cette opération ajoute une ou plusieurs configurations à un profil multimédia existant. Si l'une des configurations existe déjà dans le profil multimédia, elle est remplacée. Un dispositif qui prend en charge le service Media2 doit prendre en charge cette commande.

DEMANDE:

- **ProfileToken [tt:ReferenceToken]**
Jeton d'un profil existant.
- **Name – facultatif [tt:Name]**
Le dispositif doit mettre à jour le nom du profil lorsque cette option est fournie.
- **Configuration – facultatif, non limité [tr2:ConfigurationRef]**
Liste facultative des configurations à ajouter au profil.

RÉPONSE:

Ceci est un message vide.

DÉFAUTS:

- **env:Sender – ter:InvalidArgVal – ter:NoProfile**
Le jeton de profil demandé n'existe pas.
- **env:Sender – ter:InvalidArgVal – ter:NoConfig**
Une des configurations fournies indiquée par le ConfigurationToken n'existe pas.
- **env:Receiver – ter:Action – ter:ConfigurationConflict**
D'autres configurations du profil multimédia sont en conflit avec celle à ajouter et son ajout génère un conflit de profil multimédia.

CLASSE D'ACCÈS:

ACTUATE

6.2.4 Suppression d'une ou de plusieurs configurations d'un profil

Cette opération supprime une ou plusieurs configurations d'un profil multimédia existant. La suppression d'une configuration non existante doit être ignorée et ne pas générer d'erreur. Un dispositif qui prend en charge le service Media2 doit prendre en charge cette commande.

Un dispositif doit retourner une erreur si un client tente de supprimer VideoSourceConfiguration du profil comportant un élément VideoEncoderConfiguration. Pour éviter cette condition d'erreur, un client peut tout d'abord supprimer le VideoEncoderConfiguration, ou peut supprimer VideoSourceConfiguration et VideoEncoderConfiguration ensemble au moyen d'une commande RemoveConfiguration unique.

Un dispositif doit retourner une erreur si un client tente de supprimer AudioSourceConfiguration du profil comportant un AudioEncoderConfiguration. Pour éviter cette condition d'erreur, un client peut tout d'abord supprimer le AudioEncoderConfiguration, ou peut supprimer AudioSourceConfiguration et AudioEncoderConfiguration ensemble au moyen d'une commande RemoveConfiguration unique.

DEMANDE:

- **ProfileToken [tt:ReferenceToken]**
Jeton d'un profil existant.
- **Configuration – non limité [tr2:ConfigurationRef]**
Liste des configurations à supprimer du profil.

RÉPONSE:

Ceci est un message vide.

DÉFAUTS:

- **env:Sender – ter:InvalidArgVal – ter:NoProfile**
Le jeton de profil demandé n'existe pas.
- **env:Receiver – ter:Action – ter:ConfigurationConflict**
D'autres configurations du profil multimédia sont dépendantes de cette configuration et sa suppression génère un état conflictuel du profil multimédia.

CLASSE D'ACCÈS:

ACTUATE

6.2.5 Suppression de profil multimédia

Cette opération supprime un profil. Le dispositif doit prendre en charge la suppression d'un profil multimédia au moyen de la commande DeleteProfile.

DEMANDE:

- **Token [tt:ReferenceToken]**
Jeton d'un profil existant.

RÉPONSE:

Ceci est un message vide.

DÉFAUTS:

- **env:Sender – ter:InvalidArgVal – ter:NoProfile**
Le jeton de profil demandé n'existe pas.
- **env:Sender – ter:Action – ter:DeletionOfFixedProfile**
Un profil fixe ne peut pas être supprimé.

CLASSE D'ACCÈS:

ACTUATE**6.3 Configurations multimédias****6.3.1 Généralités**

Un profil multimédia est composé d'un ensemble de configurations multimédias. Les profils multimédias sont utilisés par un client pour configurer les propriétés d'un flux multimédia provenant d'un dispositif.

Un dispositif doit fournir au moins un profil multimédia au démarrage. Il convient qu'un dispositif fournisse des profils "prêts à l'emploi" pour les configurations multimédias les plus courantes qu'il propose.

Un profil est composé d'un ensemble d'entités de configuration interconnectées. Les configurations sont fournies par le dispositif et peuvent être soit statiques, soit créées de manière dynamique par celui-ci. Par exemple, les configurations dynamiques peuvent être créées par le dispositif en fonction des ressources de codage disponibles actuelles. Les paragraphes 6.3.2 à 6.3.10 énumèrent les configurations disponibles, chaque configuration définissant les contraintes spécifiques à l'entité. Un profil est composé de l'ensemble ou d'un sous-ensemble de ces entités de configuration. Suivant les fonctionnalités du dispositif, une entité de configuration particulière peut faire partie ou non d'un profil. Par exemple, un profil avec une source audio et une configuration de codeur audio peut exister uniquement dans un dispositif avec support audio.

Toutes les configurations comportent les paramètres de base suivants

- **Token** – Identifiant de la configuration. Ce paramètre est en lecture seule et ne peut pas être modifié par les clients.
- **Name** – Nom configurable comportant jusqu'à 64 caractères au maximum.
- **UseCount** – Propriété en lecture seule indiquant le nombre de profils référencés. Déconseillé pour le service Media2; les dispositifs peuvent prendre en charge cette valeur.

6.3.2 Configuration de source vidéo

Une VideoSourceConfiguration contient une référence à une VideoSource et une structure Bounds (bornes) contenant soit la zone de pixels totale de la VideoSource, soit une sous-portion de celle-ci. Les éléments Bounds et VideoSource définissent l'image qui est transmise en continu à un client.

L'option Rotate (Rotation) spécifie une rotation facultative de la zone définie par les paramètres Bounds. Noter que dans le cas, par exemple, d'une rotation à 90 degrés, le paramètre de largeur correspond à la hauteur de l'élément Video et inversement.

Tous les systèmes de coordonnées (par exemple, Privacy Masks (Masques de confidentialité) dans le service Media2 et Motion Regions (Zones de mouvement) dans le service d'analyse) qui s'appliquent à une configuration de source vidéo sont basés sur l'image résultante après application de la structure Bounds, puis de l'option Rotate à l'image source.

L'option `LensDescription` (Description de la lentille) permet de décrire la déformation géométrique de la source vidéo. Pour des informations détaillées, voir l'Annexe B.

Les options `SceneOrientation` (Orientation de scène) permettent de décrire l'orientation de la scène capturée par la source vidéo. Cette orientation peut être la suivante: `Below` (au-dessous) (par rapport au plafond), `Above` (au-dessus) (par rapport au sol ou à une table) et `Horizon` (horizontale) (sur une paroi ou un mur). Certains dispositifs peuvent prendre en charge la détection automatique des options `SceneOrientation`.

6.3.3 Configuration de codeur vidéo

Une `VideoEncoderConfiguration` contient les paramètres suivants de configuration du codage de données vidéo:

- `Encoding` – Codage utilisé pour la compression de la vidéo. Voir la référence `Type Name` (Nom de type) IANA MIME pour une liste complète de valeurs.
- `Resolution` – Résolution de pixel des données vidéo codées.
- `Quality` – Détermine la qualité de la vidéo. Une valeur élevée dans la plage de qualité prise en charge indique une qualité supérieure.
- `RateControl` – Définit les paramètres de configuration du débit binaire [kbps], un paramètre `EncodingInterval` (intervalle de codage et de transmission des images) et un paramètre `TargetFrameRate` [fps] de configuration du taux de trame de sortie.
- Profil de codage et longueur GOV [trame].

La structure `VideoEncoderConfiguration` contient également des paramètres de multidiffusion.

En l'absence de la structure complète du paramètre `RateControl`, l'état actuel du contrôle de débit n'est pas défini et spécifique au fournisseur. Un dispositif qui prend en charge la désactivation des mécanismes de contrôle du débit doit refléter cette prise en charge par omission de l'élément `RateControl` lorsqu'il est désactivé; à défaut, il doit retourner les valeurs actuelles utilisées pour `RateControl`. En l'absence de `RateControl`, les options respectives spécifient si un élément `RateControl` peut être (r)ajouté.

6.3.4 Configuration de source audio

`AudioSourceConfiguration` contient une référence à `AudioSource`.

6.3.5 Configuration de codeur audio

`AudioEncoderConfiguration` contient les paramètres suivants de codage des données audio:

- `Encoding` – Codage utilisé pour la compression des données audio. Voir la référence `Type Name` (Nom de type) IANA MIME pour une liste complète de valeurs.
- `Bitrate` – Débit binaire de sortie [kbps].
- `SampleRate` – Fréquence d'échantillonnage de sortie [kHz].

La structure `AudioEncoderConfiguration` contient également des paramètres de multidiffusion.

6.3.6 Configuration PTZ

Un profil comportant une configuration PTZ permet la transmission en continu du statut PTZ dans le flux de métadonnées. De plus, le profil multimédia peut être utilisé pour contrôler le mouvement PTZ tel que défini dans la spécification du service PTZ.

6.3.7 Configuration d'analyse

Un profil contenant une `AnalyticsConfiguration` active des flux utilisant ce profil multimédia pour contenir des données d'analyse (dans les métadonnées) telles que définies par la référence de

configuration soumise. Pour la configuration d'analyse, se reporter à la spécification du service d'analyse ONVIF.

Un dispositif doit retourner une erreur (ter:NoConfig) si un client tente d'ajouter VideoAnalyticsConfiguration au profil en l'absence de la configuration de source exigée (par exemple, VideoSourceConfiguration). Pour éviter cette condition d'erreur, un client peut tout d'abord ajouter une configuration de source exigée (par exemple, VideoSourceConfiguration), ou peut ajouter une configuration de source exigée (par exemple, VideoSourceConfiguration) et VideoAnalyticsConfiguration ensemble dans une commande AddConfiguration unique. Un dispositif peut refuser l'ajout d'une configuration d'analyse vidéo avant une configuration de source vidéo. Dans ce cas, il convient qu'il réponde avec un défaut ConfigurationConflict.

Lorsqu'une configuration d'analyse est présente dans un profil, la configuration de métadonnées peut activer la transmission continue de la description de scène dans les flux RTP (voir 6.3.8).

Un dispositif peut ne pas permettre de référencer exactement la même AnalyticsConfiguration de plusieurs profils multimédias avec différentes VideoSourceConfigurations. Si le dispositif le permet, il doit générer des descriptions de scène individuelles pour chaque profil, le système de coordonnées d'une description de scène étant associé à une VideoSourceConfiguration spécifique. Des règles de masquage et géométriques concernent en outre le système de coordonnées de la VideoSourceConfiguration. Cela peut nécessiter de traiter séparément l'ensemble des analyses vidéo pour chaque VideoSourceConfiguration, même si elles font référence exactement à la même VideoSource.

6.3.8 Configuration de métadonnées

Un profil contenant une MetadataConfiguration active la transmission en continu des métadonnées. Les métadonnées peuvent être composées d'événements, du statut PTZ et/ou de données d'analyse.

Une MetadataConfiguration contient des paramètres de sélection des données à inclure dans le flux de métadonnées. Les choix comprennent le statut PTZ, la position PTZ, les événements tels que définis par un abonnement et les données d'analyse. Les données d'abonnement d'événement sont décrites dans la section "Traitement des événements" de la spécification principale de l'ONVIF. Les paramètres d'analyse définissent les données à inclure depuis la partie de moteur d'analyse du profil (voir 6.3.7).

La structure contient également les paramètres de multidiffusion utilisés pour configurer et commander la multidiffusion du flux de métadonnées. Les dispositifs qui prennent en charge les métadonnées comprimées doivent indiquer l'algorithme de compression disponible tel que défini dans le MetadataCompressionType. Les types de compression actuellement définis sont "GZIP" et "EXI". Pour des informations détaillées concernant la configuration EXI, voir l'Annexe A.

6.3.9 Configuration de sortie audio

6.3.9.1 Généralités

La configuration de sortie audio contient les paramètres suivants:

- SourceToken: référence à une sortie audio existante.
- OutputLevel: paramètre de configuration du volume de la sortie.
- SendPrimacy: paramètre qui peut être utilisé pour les dispositifs équipés d'une entrée/sortie audio bidirectionnelle non simultanée pour configurer la direction de transmission active (voir 6.3.9.2).

6.3.9.2 Modes de voie audio

Une voie audio peut prendre en charge différents types de transmission audio. En mode de fonctionnement bidirectionnel simultané, aucun traitement particulier n'est exigé. En mode de fonctionnement bidirectionnel non simultané, il est nécessaire de permuter la direction de transmission.

Un paramètre Send-Primacy facultatif à l'intérieur de AudioOutputConfiguration indique la direction actuellement active. Un client peut basculer d'un mode à l'autre en attribuant une valeur à AudioOutputConfiguration.

Les modes suivants sont définis pour Send-Primacy:

- www.onvif.org/ver20/HalfDuplex/Server
Le serveur peut envoyer des données audio au client. Dans ce mode, le client ne doit pas envoyer de données audio par voie de retour au dispositif.
- www.onvif.org/ver20/HalfDuplex/Client
Le client peut envoyer des données audio au serveur par la voie de retour. Dans ce mode, le dispositif ne doit pas envoyer de données audio au client.
- www.onvif.org/ver20/HalfDuplex/Auto
La manière de traiter l'envoi et la réception de données audio revient au dispositif.

L'annulation de l'écho acoustique ne relève pas du domaine d'application du présent document.

6.3.10 Configuration de décodeur audio

La configuration de décodeur audio ne contient aucun paramètre de configuration du décodage. Les codages pris en charge par une configuration de décodeur audio peuvent être extraits par la méthode GetAudioDecoderConfigurationOptions.

6.4 Méthodes de configuration multimédia

6.4.1 Généralités

Un dispositif doit, pour chaque entité prise en charge, proposer l'ensemble de commandes Get<entity>Configuration, Set<entity>Configuration et Get<entity>ConfigurationOptions.

6.4.2 Obtention des configurations

L'opération Get<entity>Configuration permet d'extraire les réglages réels d'une ou plusieurs configurations. La syntaxe et la sémantique du message de demande sont identiques pour toutes les entités de configuration:

- Lorsqu'un jeton de configuration est fourni, le dispositif doit répondre avec la configuration demandée ou indiquer une erreur en l'absence de celle-ci.
- Dans le cas où seul un jeton de profil est fourni, le dispositif doit répondre avec toutes les configurations compatibles avec le profil multimédia fourni.
- En l'absence de jeton, le dispositif doit répondre avec toutes les configurations disponibles.

DEMANDE:

- **ProfileToken – facultatif [tt:ReferenceToken]**
Jeton de profil facultatif destiné à extraire toutes les configurations compatibles.
- **ConfigurationToken – facultatif [tt:ReferenceToken]**
Jeton facultatif destiné à extraire exactement une configuration.

RÉPONSE:

- **Configurations – facultatif, non limité [tt:<entity>Configuration]**

Liste des configurations.

DÉFAUTS:

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

Le jeton de profil demandé n'existe pas.

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

La configuration demandée n'existe pas dans le contexte indiqué.

CLASSE D'ACCÈS:

READ_MEDIA**6.4.3 Modification d'une configuration**

L'opération Set<entity>Configuration modifie une configuration. La modification peut avoir un effet immédiat sur les flux en cours d'exécution, mais il n'est pas certain que les modifications soient effectives à moins que le client ne relance tout flux concerné. Les méthodes du client de modification d'un flux en cours d'exécution ne relèvent pas du domaine d'application de la présente spécification.

DEMANDE:

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

L'élément Configuration contient la configuration modifiée. La configuration doit exister dans le dispositif.

RÉPONSE:

Ce message est vide.

DÉFAUTS:

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

La configuration demandée n'existe pas dans le contexte indiqué.

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

Les paramètres de configuration ne peuvent pas être définis.

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

Les nouveaux réglages entrent en conflit avec d'autres utilisations de la configuration.

CLASSE D'ACCÈS:

ACTUATE

Note pour la VideoEncoderConfiguration: si nécessaire, le dispositif peut adapter les valeurs de paramètres pour les éléments Quality et RateControl sans retourner d'erreur. Un dispositif doit adapter une BitrateLimit hors de toute plage et non retourner un défaut.

Note pour la AudioSourceConfiguration: si les nouveaux réglages invalident les paramètres déjà négociés à l'aide du protocole RTSP (en modifiant le type de codec, par exemple), le dispositif ne doit pas les appliquer aux flux existants. Il doit plutôt continuer le flux à l'aide des anciens paramètres ou arrêter l'envoi de données sur les flux concernés.

6.4.4 Obtention d'options de configuration

L'opération `Get<entity>ConfigurationOptions` retourne les paramètres disponibles et leurs plages valides au client. Toute combinaison des paramètres obtenus avec un profil multimédia et une configuration donnés doit constituer une entrée valide pour la commande de définition de configuration correspondante.

En présence d'un jeton de configuration, le dispositif doit retourner les options compatibles avec cette configuration. Si un jeton de profil multimédia est spécifié, le dispositif doit retourner les options compatibles avec ce profil multimédia. Si un jeton de profil multimédia et un jeton de configuration sont tous deux spécifiés, le dispositif doit retourner les options compatibles tant avec ce profil multimédia qu'avec cette configuration. En l'absence de spécification de jetons, les options doivent être considérées comme génériques pour le dispositif.

DEMANDE:

- **ConfigurationToken – facultatif [tt:ReferenceToken]**
Jeton facultatif destiné à extraire exactement une configuration.
- **ProfileToken – facultatif [tt:ReferenceToken]**
Jeton de profil facultatif destiné à extraire toutes les configurations compatibles.

RÉPONSE:

- **ConfigurationsOptions – non limité [tt:<entity>Configuration]**
Selon la structure de `tr2:Get<entity>ConfigurationOptionsResponse`, les options retournent un ou plusieurs éléments.

NOTE Les entités `AudioOutput`, `AudioSource`, `Metadata` et `VideoSource` retournent un seul élément.

DÉFAUTS:

- **env:Sender – ter:InvalidArgVal – ter:NoProfile**
Le jeton de profil demandé n'existe pas.
- **env:Sender – ter:InvalidArgVal – ter:NoConfig**
La configuration demandée n'existe pas dans le contexte indiqué.

CLASSE D'ACCÈS:

READ_MEDIA

6.4.5 GetVideoEncoderInstances

Cette commande fournit des informations concernant le nombre de codeurs vidéo que peut instancier simultanément un dispositif pour une `VideoSourceConfiguration`. Un dispositif doit prendre en charge cette commande.

La réponse d'information contient les informations suivantes:

- **Total:** Nombre total d'instances de codeur indépendant du codec;
- **Codec:** Nombre d'instances de codeur pour chaque codec pris en charge.

Un dispositif doit garantir l'instanciation simultanée du nombre indiqué d'instances. Si un dispositif limite le nombre d'instances de chaque type de codage vidéo particulier, la réponse doit contenir les informations par codec vidéo.

La somme totale d'instances de codeur vidéo sur toutes les sources vidéo d'un dispositif ne doit pas dépasser la valeur indiquée par l'intermédiaire de `MaximumNumberOfProfiles`.

Par exemple, si un dispositif comprend deux VideoSourceConfigurations et si la première configuration permet un nombre total de deux instances simultanées et la seconde configuration permet une seule instance, il doit permettre la création d'au moins trois profils multimédia.

DEMANDE:

- **ConfigurationToken [tt:ReferenceToken]**

Jeton de la configuration.

RÉPONSE:

- **Info [tr2:EncoderInstanceInfo]**

Ce message contient le nombre total minimal garanti d'instances de codeur (applications) par VideoSourceConfiguration.

DÉFAUTS:

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

La configuration demandée n'existe pas dans le contexte indiqué.

CLASSE D'ACCÈS:

READ_MEDIA

6.5 GetStreamUri

Cette opération demande un URI qui peut être utilisé pour initier un flux multimédia en direct en utilisant le protocole RTSP en tant que protocole de commande. Il convient que l'URI retourné reste définitivement valide, même en cas de modification des paramètres du profil.

Les types de flux suivants sont définis:

- RtpUnicast – transmission en continu de RTP par RTSP par l'intermédiaire de UDP Unicast.
- RtpMulticast – transmission en continu de RTP par RTSP par l'intermédiaire de UDP Multicast.
- RTSP – transmission en continu de RTP par RTSP par le protocole TCP.
- RtpOverHttp – Tunnelisation de la voie de commande RTSP et du flux RTP par le protocole HTTP ou HTTPS.

Pour assurer une compatibilité totale avec d'autres services ONVIF, un dispositif ne doit pas générer d'URI comportant plus de 128 octets.

Un dispositif qui indique la fonctionnalité RTSPStreaming doit prendre en charge cette commande. Lors de la demande du protocole de transport RtpOverHttp, un dispositif doit retourner un URI qui utilise le même port que le service web. Cette action active une traversée NAT continue.

DEMANDE:

- **Protocol [xs:string]**

Le protocole définit le mode prévu de transmission en continu des données codées au client.

- **ProfileToken [tt:ReferenceToken]**

L'élément ProfileToken indique le profil multimédia à utiliser et définit la configuration du contenu du flux.

RÉPONSE:

- **Uri [xs:anyUri]**

URI stable à utiliser dans le cadre de la demande de flux multimédia.

DÉFAUTS:

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

Le jeton de profil demandé n'existe pas.

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

Le profil spécifié n'est pas pris en charge.

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

Le protocole spécifié provoque un conflit avec un autre flux.

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

Les configurations ne sont pas toutes prévues pour une multidiffusion.

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

Le profil multimédia spécifié contient soit des sources non utilisées, soit des configurations de codeur sans source correspondante.

CLASSE D'ACCÈS:

READ_MEDIA**6.6 GetSnapshotUri**

Un client de réseau utilise la commande GetSnapshotUri pour obtenir un instantané JPEG depuis le dispositif. L'URI retourné doit rester définitivement valide, même en cas de modification des paramètres du profil. L'URI peut être utilisé pour l'acquisition d'une ou de plusieurs images JPEG par l'intermédiaire d'une opération GET HTTP.

Le format d'image est toujours JPEG, quel que soit le réglage de codage dans le profil multimédia. Il convient que les réglages JPEG (tels que la résolution ou la qualité) soient issus du profil si approprié. L'image fournie doit être mise à jour automatiquement et indépendamment des appels en direction de GetSnapshotUri.

Un dispositif doit prendre en charge cette commande lorsque la fonctionnalité SnapshotUri est définie sur la valeur "true".

DEMANDE:

- **ProfileToken [tt:ReferenceToken]**

L'élément ProfileToken indique le profil multimédia à utiliser et définit la configuration du contenu du flux.

RÉPONSE:

- **Uri [xs:anyUri]**

Uri stable à utiliser pour l'acquisition d'un instantané au format JPEG, ainsi que les paramètres définissant la durée de vie de l'Uri.

DÉFAUTS:

- **env:Sender – ter:InvalidArgVal – ter:NoProfile**
Le jeton de profil demandé n'existe pas.
- **env:Receiver – ter:Action – ter:IncompleteConfiguration**
Le profil multimédia spécifié contient soit des sources non utilisées, soit des configurations de codeur sans source correspondante.

CLASSE D'ACCÈS:

READ_MEDIA

6.7 Multidiffusion

6.7.1 Généralités

La présente spécification définit deux mécanismes de contrôle des flux de multidiffusion. Outre l'établissement de sessions de multidiffusion par l'intermédiaire de sessions RTSP telles que définies en 6.5, les dispositions de 6.7 définissent un mécanisme de transmission en continu en multidiffusion par lequel la transmission en continu réelle est contrôlée par le protocole IGMP. Utiliser cette méthode avec prudence, dans la mesure où une configuration de réseau incorrecte peut entraîner une inondation du réseau par des paquets audio et vidéo.

Un dispositif qui indique la prise en charge de la transmission en continu en multidiffusion non contrôlée par un protocole RTSP au moyen de la fonctionnalité AutoStartMulticast doit prendre en charge les méthodes définies en 6.7.

6.7.2 Démarrage de transmission en continu en multidiffusion

Cette commande démarre la transmission en continu en multidiffusion à l'aide du profil multimédia spécifié d'un dispositif. La transmission en continu se poursuit jusqu'à ce que la commande StopMulticastStreaming soit appelée pour le même profil. La transmission doit reprendre après redémarrage. Elle peut être interrompue en appliquant la méthode StopMulticastStreaming. L'adresse, le port et la TTL de multidiffusion sont configurés dans la VideoEncoderConfiguration, AudioEncoderConfiguration et MetadataConfiguration, respectivement.

La transmission en continu en multidiffusion peut s'interrompre lors de la suppression du profil correspondant ou de la modification de l'une de ses configurations par l'une des méthodes de configuration définies.

La mise en œuvre doit assurer que le flux RTP peut être décodé sans établir une connexion de contrôle RTSP. Notamment dans le cas d'un format vidéo H.264, l'en-tête SPS/PPS doit faire l'objet d'un envoi intrabande.

DEMANDE:

- **ProfileToken [tt:ReferenceToken]**
L'élément ProfileToken indique le profil multimédia à utiliser et définit la configuration du contenu du flux.

RÉPONSE:

Ce message est vide.

DÉFAUTS:

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

Le jeton de profil demandé n'existe pas.

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

Le profil multimédia spécifié ne contient pas de référence à une configuration de codeur vidéo ou à une configuration de source vidéo, de référence à une configuration de source ou à une configuration de codeur audio, ou de référence à une configuration de métadonnées

CLASSE D'ACCÈS:

ACTUATE**6.7.3 Interruption de transmission en continu en multidiffusion**

Cette commande interrompt la transmission en continu en multidiffusion à l'aide du profil multimédia spécifié d'un dispositif. Dans le cas où un dispositif reçoit une demande StopMulticastStreaming dont la transmission en continu en multidiffusion n'a pas commencé, il convient qu'il réponde avec une StopMulticastStreamingResponse satisfaisante.

DEMANDE:

- **ProfileToken [tt:ReferenceToken]**

L'élément ProfileToken indique le profil multimédia à utiliser.

RÉPONSE:

Ce message est vide.

DÉFAUTS:

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

Le jeton de profil demandé n'existe pas.

CLASSE D'ACCÈS:

ACTUATE**6.8 SetSynchronizationPoint**

Les points de synchronisation permettent aux clients de décoder et d'utiliser correctement toutes les données après le point de synchronisation.

Par exemple, si un flux vidéo est configuré avec une distance de trame I élevée et qu'un client perd un seul paquet, le client n'affiche pas la vidéo tant que la trame I suivante n'est pas transmise. Dans de tels cas, le client peut demander un point de synchronisation qui impose au dispositif d'ajouter une trame I dès que possible. Les clients peuvent demander des points de synchronisation pour des profils. Le dispositif doit ajouter des points de synchronisation pour tous les flux associés à ce profil.

De même, un point de synchronisation est utilisé pour obtenir une mise à jour sur le PTZ total ou le statut d'événement par l'intermédiaire du flux de métadonnées.

Si un flux vidéo est associé au profil, une trame I doit être ajoutée à ce flux vidéo. Si un flux d'événement est associé au profil, la demande de point de synchronisation doit être gérée comme indiqué en 10.2.3. Si le profil est configuré pour les métadonnées de PTZ, la position PTZ doit être répétée dans le flux de métadonnées.

Un dispositif doit prendre en charge la demande de trame I au moyen de la commande SetSynchronizationPoint si la fonctionnalité RTSPStreaming est définie.

DEMANDE:

- **ProfileToken [tt:ReferenceToken]**

Référence de profil pour laquelle un point de synchronisation fait l'objet d'une demande.

RÉPONSE:

Ce message est vide.

DÉFAUTS:

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

Le jeton de profil demandé n'existe pas.

CLASSE D'ACCÈS:

ACTUATE

6.9 Mode de source vidéo

6.9.1 Généralités

Un dispositif peut être capable de modifier son mode de source vidéo. Différents modes de source vidéo peuvent influencer sur les options disponibles dans GetVideoSourceConfigurationOptions et GetVideoEncoderConfigurationOptions, comme autoriser uniquement les formats 16 × 9 dans un mode et uniquement les formats 4 × 3 dans un autre, voire uniquement les taux de trames dérivés 30 fps dans un mode et les taux de trames dérivés 25 fps dans un autre. La commande GetVideoSourceModes fournit des informations de synthèse concernant les différents modes pris en charge par le dispositif.

6.9.2 GetVideoSourceModes

Un dispositif retourne les informations concernant le mode de source vidéo actuel et les modes de source définissables d'une source vidéo spécifiée. Un dispositif qui indique une fonctionnalité VideoSourceMode doit prendre en charge cette commande.

DEMANDE:

- **VideoSourceToken [tt:ReferenceToken]**

Source vidéo pour laquelle il convient d'extraire les modes de source.

RÉPONSE:

- **VideoSourceMode – non limité [tt:VideoSourceMode]**

Liste des informations de mode contenant les fonctionnalités de la source vidéo respective.

DÉFAUTS:

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

La sortie vidéo demandée n'existe pas.

CLASSE D'ACCÈS:

READ_SYSTEM

6.9.3 SetVideoSourceMode

SetVideoSourceMode modifie la structure de profil multimédia relative à la source vidéo applicable au mode de source vidéo spécifié. Un dispositif qui indique une fonctionnalité VideoSourceMode doit prendre en charge cette commande. Le comportement après modification du mode n'est pas défini dans la présente spécification.

DEMANDE:

- **VideoSourceToken [tt:ReferenceToken]**
Source vidéo pour laquelle il convient d'extraire les modes de source.
- **VideoSourceModeToken [tt:ReferenceToken]**
Jeton du mode de source vidéo à définir.

RÉPONSE:

- **Reboot [xs:boolean]**
Informations concernant le redémarrage après retour de la réponse. Lorsque la fonctionnalité Reboot est définie comme "true", un dispositif redémarre automatiquement après définition du mode.

DÉFAUTS:

- **env:Sender – ter:InvalidArgVal – ter:NoVideoSource**
La sortie vidéo demandée n'existe pas.
- **env:Sender – ter:InvalidArgVal – ter:NoVideoSourceMode**
Le mode de source vidéo demandé n'existe pas.

CLASSE D'ACCÈS:

WRITE_SYSTEM

6.10 Affichage écran (OSD – *On-Screen Display*)

6.10.1 Généralités

Le service OSD offre des fonctions permettant à un client de contrôler et configurer l'affichage écran d'un dispositif. Le service propose la configuration OSD sous plusieurs formes (par exemple, image, texte, date et heure). Les fonctions d'extraction et de configuration des configurations sont également fournies. Toutes les configurations OSD sont associées à une VideoSourceConfiguration qui affiche le contenu d'un OSD. La Figure 5 présente un exemple d'affichage écran comprenant les coordonnées d'écran normalisées.

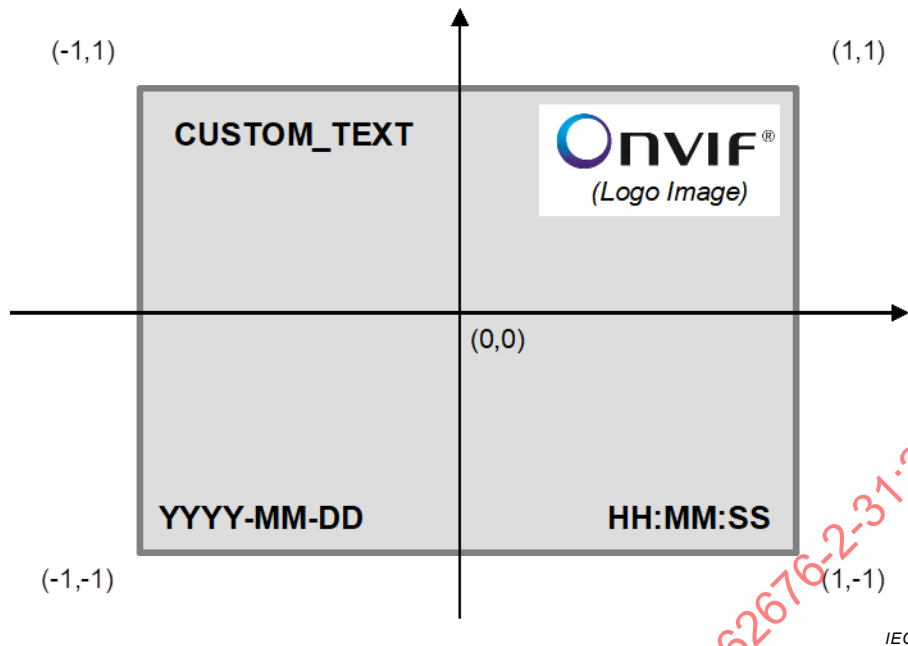


Figure 5 – Exemple avec quatre configurations OSD

Le Paragraphe 6.10 définit les méthodes de création et de suppression des configurations OSD, ainsi que d'obtention, de définition et de requête des options. Un dispositif qui indique une prise en charge d'OSD par la fonctionnalité OSD doit prendre en charge toutes les méthodes OSD définies en 6.10.

Des dispositifs qui prennent en charge une OSDTextConfiguration temporaire, doivent notifier la fonctionnalité TemporaryOSDText telle que définie en 6.12. Par défaut, le dispositif doit rendre toutes les OSDTextConfigurations permanentes lors du redémarrage, mais lorsque l'attribut IsPersistentText de la OSDTextConfiguration est défini comme "false", le contenu de texte OSD doit être effacé après le redémarrage. OSDConfiguration doit toujours être valide après le redémarrage.

6.10.2 CreateOSD

Cette opération crée une nouvelle configuration OSD avec des valeurs spécifiées, et associe également le nouvel OSD et une VideoSourceConfiguration existante identifiée par son VideoSourceConfigurationToken. Toute valeur exigée par un dispositif pour une nouvelle configuration OSD facultative et absente du message CreateOSD peut être adaptée à la valeur appropriée par le dispositif. Un dispositif qui indique la fonctionnalité OSD doit prendre en charge la création d'un OSD tant que le nombre d'OSD existants ne dépasse pas la valeur de MaximumNumberOfOSDs dans les GetOSDOptions. Une configuration OSD créée doit pouvoir être supprimée.

DEMANDE:

- **OSD [tt:OSDConfiguration]**

Contient une nouvelle configuration OSD avec la valeur spécifiée. Le dispositif est chargé de l'attribution du jeton OSD. Le jeton OSD dans la CreateOSDRequest peut être ignoré.

RÉPONSE:

- **OSDToken [tt:ReferenceToken]**

Jeton attribué à la configuration nouvellement créée.

DÉFAUTS:

- **env:Receiver – ter:Action – ter:MaxOSDs**
Le nombre maximal d'OSD pris en charge par la VideoSourceConfiguration spécifique a été atteint.
- **env:Sender – ter:InvalidArgVal – ter:NoConfig**
Une des configurations fournies indiquée par le ConfigurationToken n'existe pas.
- **env:Receiver – ter:InvalidArgVal – ter:ConfigModify**
Les paramètres de configuration ne peuvent pas être définis.

CLASSE D'ACCÈS:

ACTUATE**6.10.3 DeleteOSD**

Cette opération supprime une configuration OSD.

DEMANDE:

- **OSDToken [tt:ReferenceToken]**
Jeton attribué à la configuration nouvellement créée

RÉPONSE:

Ceci est un message vide.

DÉFAUTS:

- **env:Sender – ter:InvalidArgVal – ter:NoConfig**
Le jeton OSD demandé (OSDToken) n'existe pas.

CLASSE D'ACCÈS:

ACTUATE**6.10.4 GetOSDs**

Cette opération répertorie les configurations OSD existantes pour le dispositif. Le dispositif doit prendre en charge la liste des configurations OSD existantes par la commande GetOSD. En l'absence d'un jeton OSD et d'un jeton de configuration de source vidéo, le dispositif doit répondre avec toutes les configurations OSD disponibles.

DEMANDE:

- **OSDToken – facultatif [tt:ReferenceToken]**
Jeton d'une configuration OSD existante. Lorsqu'un jeton OSD est fourni, le dispositif doit répondre avec la configuration demandée ou indiquer une erreur en l'absence de celle-ci.
- **ConfigurationToken – facultatif [tt:ReferenceToken]**
Dans le cas où seul un jeton de configuration de source vidéo est fourni, le dispositif doit répondre avec toutes les configurations existantes pour la configuration de source vidéo.

RÉPONSE:

- **OSD – facultatif, non limité [tt:OSDConfiguration]**
Liste des configurations OSD.

DÉFAUTS:

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

Le jeton OSD demandé (OSDTOKEN) n'existe pas.

CLASSE D'ACCÈS:

READ_MEDIA**6.10.5 SetOSD**

Cette opération modifie une configuration OSD. Les flux en cours d'exécution utilisant cette configuration peuvent immédiatement être mis à jour selon les nouveaux réglages.

Un dispositif doit accepter toute combinaison de paramètres retournés par GetOSDOptions. Si nécessaire, le dispositif peut adapter les valeurs de paramètres pour les éléments FontColor, FontSize et BackgroundColor sans retourner d'erreur. La modification du jeton de configuration n'est pas prise en charge.

DEMANDE:

- **OSD [tt:OSDConfiguration]**

L'élément OSD contient la configuration OSD modifiée. L'élément Configuration contient un élément qui spécifie l'OSD dont la configuration doit être modifiée. L'OSD doit exister dans le dispositif.

RÉPONSE:

Ceci est un message vide.

DÉFAUTS:

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

Le jeton OSD demandé (OSDTOKEN) n'existe pas.

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

Les paramètres de configuration ne peuvent pas être définis.

CLASSE D'ACCÈS:

ACTUATE

Une OSDTextConfiguration sans l'attribut IsPersistentText doit être interprétée comme permanente par le dispositif.

6.10.6 GetOSDOptions

Cette opération retourne les options disponibles lorsque les paramètres OSD sont reconfigurés. Le dispositif doit prendre en charge la liste des options de paramètres OSD disponibles (pour une configuration de source vidéo donnée) au moyen de la commande GetOSDOptions. Toute combinaison des paramètres obtenus avec une configuration de source vidéo donnée doit constituer une entrée valide pour la commande SetOSD correspondante.

DEMANDE:

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

L'élément ConfigurationToken spécifie la configuration de source vidéo dont les options OSD appropriées font l'objet d'une demande. Le jeton de configuration de source vidéo doit exister dans le dispositif.

RÉPONSE:

- **OSDOptions [tt:OSDConfigurationOptions]**

Options OSD appropriées à la configuration de source vidéo spécifiée dans la demande.

DÉFAUTS:

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

Le jeton OSD demandé (OSDTOKEN) n'existe pas.

CLASSE D'ACCÈS:

READ_MEDIA**6.11 Masques de confidentialité****6.11.1 Généralités**

Les masques de confidentialité permettent de couvrir des zones de l'image. Le dispositif indique, par les options disponibles, le nombre de masques et le nombre de bords que chaque masque peut prendre en charge. De plus, les dispositifs peuvent limiter la prise en charge des masques aux seuls masques de forme rectangulaire. La Figure 6 présente le mode de définition du masque par un polygone utilisant des coordonnées normalisées par rapport à la fenêtre VideoSourceConfiguration.

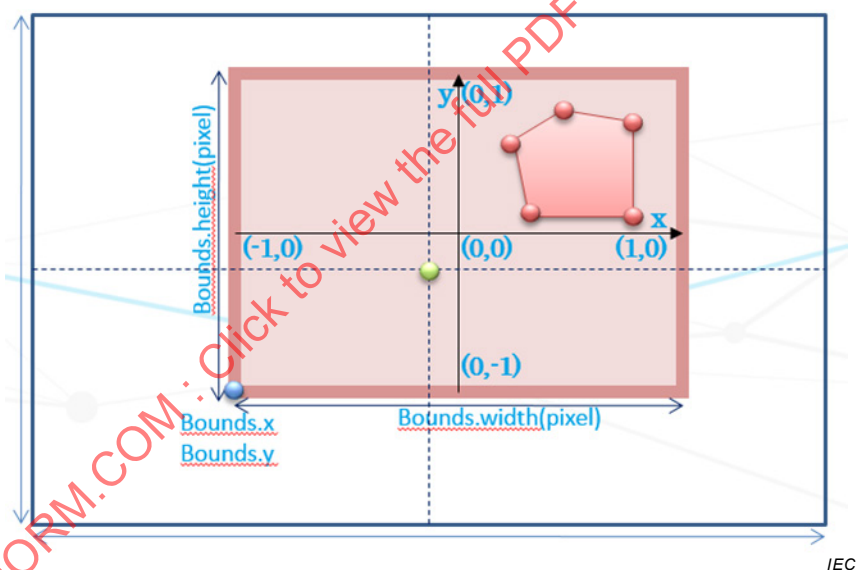


Figure 6 – Exemple d'écran avec masque et système de coordonnées

Un dispositif peut prendre en charge les mises à jour automatiques de l'emplacement et de la forme des masques selon le fonctionnement interne. Les informations détaillées ne relèvent pas du domaine d'application de la présente spécification.

Selon les options prises en charge, un dispositif peut prendre en charge les types de masques suivants:

- **Couleur (Color)** Le dispositif indique quelles valeurs ou plages de couleurs sont prises en charge.
De plus, un dispositif peut indiquer qu'il prend en charge une seule couleur pour tous les masques d'une VideoSourceConfiguration.
- **Flou (Blurred)** La zone masquée n'est pas mise au point.
- **Pixélisé (Pixelized)** La zone masquée est recouverte d'une mosaïque.

6.11.2 CreateMask

Cette opération crée un nouveau masque pour une VideoSourceConfiguration existante. Un dispositif qui indique la prise en charge des masques par la fonctionnalité Mask (Masque) doit prendre en charge la création de masques au moyen de cette fonction tant que le nombre de masques existants ne dépasse pas la valeur de MaxMasks pour la VideoSourceConfiguration donnée.

Un dispositif doit ignorer le jeton de masque transmis dans la commande.

DEMANDE:

- **Mask [tr2:Mask]**
Contient la nouvelle configuration de masque. Le dispositif est chargé de l'attribution du jeton de masque.

RÉPONSE:

- **Token [tt:ReferenceToken]**
Jeton attribué au masque nouvellement créé.

DÉFAUTS:

- **env:Receiver – ter:Action – ter:MaxMasks**
Le nombre maximal de masques pris en charge par la VideoSourceConfiguration spécifique a été atteint.
- **env:Sender – ter:InvalidArgVal – ter:NoConfig**
La configuration de source vidéo fournie n'existe pas.
- **env:Receiver – ter:InvalidArgVal – ter:InvalidPolygon**
Le polygone fourni n'est pas pris en charge.

CLASSE D'ACCÈS:

ACTUATE

6.11.3 DeleteMask

Cette opération supprime une configuration de masque.

DEMANDE:

- **Token [tt:ReferenceToken]**
Jeton attribué au masque nouvellement créé.

RÉPONSE:

Ceci est un message vide.

DÉFAUTS:

- **env:Sender – ter:InvalidArgVal – ter:NoConfig**
Le jeton de masque demandé n'existe pas.

CLASSE D'ACCÈS:

ACTUATE

6.11.4 GetMasks

Cette opération répertorie les configurations de masque existantes pour le dispositif. Un dispositif qui indique la prise en charge de la fonctionnalité Mask doit prendre en charge la liste des configurations de masque existantes au moyen de cette commande. En l'absence de jeton de masque et de jeton de configuration, le dispositif doit répondre avec toutes les configurations de masque disponibles dans le dispositif.

DEMANDE:

- **MaskToken – facultatif [tt:ReferenceToken]**

Lorsqu'un jeton de masque est fourni, le dispositif doit répondre avec la configuration demandée ou indiquer une erreur en l'absence de celle-ci.

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

Lorsque seul un jeton de configuration de source vidéo est fourni, le dispositif doit répondre avec toutes les configurations existantes pour la configuration de source vidéo.

RÉPONSE:

- **Masque – facultatif, non limité [tr2:Mask]**

Liste des masques.

DÉFAUTS:

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

La configuration de source vidéo demandée ou l'élément Mask indiqué par MaskToken n'existe pas.

CLASSE D'ACCÈS:

READ_MEDIA

6.11.5 SetMask

Cette opération modifie une configuration de masque. Les flux en cours d'exécution utilisant cette configuration peuvent immédiatement être mis à jour selon les nouveaux réglages.

Un dispositif qui indique la prise en charge de l'élément Mask par ses fonctionnalités prend en charge cette commande. Il doit accepter toute combinaison de paramètres retournés par GetMaskOptions. Si nécessaire, le dispositif peut adapter les valeurs de paramètres pour les éléments Color et Polygon sans retourner d'erreur.

Noter que les dispositifs qui indiquent SingleColorOnly entraînent une mise à jour de tous les masques de la VideoSource associée.

Un dispositif qui indique RectangleOnly doit accepter tout polygone avec quatre points. Lorsque les quatre sommets ne définissent pas un rectangle exact, le dispositif peut les ajuster.

DEMANDE:

- **Mask [tr2:Mask]**

Contient la nouvelle configuration de masque. Le dispositif est chargé de l'attribution du jeton de masque.

RÉPONSE:

Ceci est un message vide.

DÉFAUTS:

- **env:Sender – ter:InvalidArgVal – ter:NoConfig**
La configuration de source vidéo fournie n'existe pas.
- **env:Receiver – ter:InvalidArgVal – ter:ConfigModify**
Les paramètres de configuration ne peuvent pas être définis.
- **env:Receiver – ter:InvalidArgVal – ter:InvalidPolygon**
Le polygone fourni n'est pas pris en charge.

CLASSE D'ACCÈS:

ACTUATE

6.11.6 GetMaskOptions

Cette opération retourne les options disponibles lorsque les paramètres Mask sont reconfigurés. Un dispositif qui indique la prise en charge de l'élément Mask par ses fonctionnalités doit prendre en charge la liste des options des paramètres Mask disponibles (pour une configuration de source vidéo donnée) au moyen de cette commande. Toute combinaison des paramètres obtenus avec une configuration de source vidéo donnée doit constituer une entrée valide pour la commande SetMask correspondante.

DEMANDE:

- **ConfigurationToken [tt:ReferenceToken]**
Jeton d'une configuration existante pour laquelle il convient de répertorier tous les masques existants.

RÉPONSE:

- **Options [tr2:MaskOptions]**
Ce message contient les options Mask appropriées à la configuration de source vidéo spécifiée dans la demande.

DÉFAUTS:

- **env:Sender – ter:InvalidArgVal – ter:NoConfig**
La configuration de source vidéo demandée n'existe pas.

CLASSE D'ACCÈS:

READ_MEDIA

6.12 Fonctionnalités

Les fonctionnalités reflètent les fonctions facultatives et la fonctionnalité d'un service. Les informations sont statiques et ne varient pas pendant le fonctionnement du dispositif.

Les fonctionnalités de profil suivantes sont disponibles:

MaximumNumberOfProfiles:	Somme des MediaProfiles fixes et dynamiques pris en charge par le dispositif. Un dispositif qui met en oeuvre ce service doit proposer cette fonctionnalité.
ConfigurationsSupported:	Énumère les configurations prises en charge par le dispositif telles que définies par tr2:ConfigurationEnumeration.
SnapshotUri:	Indique la prise en charge de GetSnapshotUri.
Rotation:	Indique la prise en charge de la fonctionnalité Rotation.

VideoSourceMode:	Indique la prise en charge de la modification du mode de source vidéo.
OSD:	Indique la prise en charge de la configuration OSD.
TemporaryOSDText:	Indique la prise en charge de la configuration de texte OSD temporaire.
Mask:	Indique la prise en charge de la configuration de masque.
SourceMask:	Indique que les masques de confidentialité sont uniquement pris en charge au niveau de la source vidéo et non au niveau de la configuration de source vidéo. Si cette disposition est vraie, tout ajout, toute suppression ou toute modification d'un masque de confidentialité effectués pour une configuration de source vidéo sont appliqués automatiquement par le dispositif à un masque de confidentialité correspondant pour toutes les autres configurations de source vidéo associées à la même source vidéo.

Les fonctionnalités de transmission en continu suivantes sont disponibles:

RTSPStreaming:	Indique la prise en charge de la transmission en continu multimédia en direct par le protocole RTSP.
RTPMulticast:	Indique la prise en charge de la multidiffusion UDP telle que décrite en 10.2.1.2.
RTP_RTSP_TCP:	Indique la prise en charge du transport RTP/RTSP/TCP tel que défini en 10.2.1.4.
NonAggregateControl:	Indique la prise en charge du contrôle RTSP non agrégé tel que décrit en 10.2.
RTSPOverWebSocket:	Indique la prise en charge de la transmission en continu RTSP/RTP par WebSocket et fournit l'URI WebSocket URI, tel que décrit en 10.2.1.6.
AutoStartMulticast:	Indique la prise en charge de la transmission en continu multidiffusion non contrôlée par protocole RTSP.

DEMANDE:

Ceci est un message vide.

RÉPONSE:

- **Capabilities [tr2:Capabilities]**
Liste des fonctionnalités définies ci-dessus.

DÉFAUTS:

Aucun

CLASSE D'ACCÈS:

PRE_AUTH

6.13 Événements

6.13.1 ProfileChange

Chaque création ou suppression d'un profil, ou chaque ajout ou suppression d'une ou plusieurs de ses configurations doit produire l'événement suivant.

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

6.13.2 ConfigurationChange

À chaque modification d'une configuration d'un dispositif, ce dernier doit générer l'événement suivant. Pour le paramètre Type, transmettre la valeur ConfigurationEnumeration appropriée.

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

6.13.3 ActiveConnections

Il convient qu'un dispositif qui prend en charge le service multimédia génère l'événement de contrôle "Active Connections" afin d'informer un client de l'utilisation actuelle de ses profils multimédia. Un dispositif satisfaisant à l'ONVIF doit utiliser le format de rubrique et de message suivant:

```
Topic: tns1:Monitoring/Profile/ActiveConnections

<xs:complexType name="ProfileStatus">
  <xs:sequence>
    <xs:element name="ActiveConnections" type="tt:ActiveConnection" minOccurs="0"
      maxOccurs="unbounded"/>
    <xs:element name="Extension" type="tt:ProfileStatusExtension" minOccurs="0" />
  </xs:sequence>
</xs:complexType>

<xs:complexType name="ActiveConnection">
  <xs:sequence>
    <xs:element name="CurrentBitrate" type="xs:float"/>
    <xs:element name="CurrentFps" type="xs:float"/>
    <xs:any namespace="##any" processContents="lax" minOccurs="0"
      maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<tt:MessageDescription IsProperty="true">
  <tt:Source>
    <tt:SimpleItemDescription Name="Profile" Type="tt:ReferenceToken"/>
  </tt:Source>
  <tt>Data>
    <tt:ElementItemDescription Name="Status" Type="tt:ProfileStatus"/>
  </tt>Data>
</tt:MessageDescription>
```

6.14 Écarts du service multimédia version 1

6.14.1 Généralités

Les paragraphes précédents définissent l'interface du service multimédia version 2 qui constitue une extension du service multimédia version 1. Le présent paragraphe définit les écarts de la version 1. Pour les informations détaillées concernant l'interface, se reporter à l'Article F.3.

6.14.2 Gestion des profils

- CreateProfile

La méthode permet de créer uniquement un profil vide qui doit ensuite être rempli avec une séquence d'appels Add<entity>Configuration.

- AddConfiguration

Contrairement à une méthode unique qui prend en charge l'ajout de plusieurs configurations, des méthodes distinctes sont définies pour l'ajout de chacune des configurations.

- RemoveConfiguration

Contrairement à une méthode unique qui prend en charge la suppression de plusieurs configurations, des méthodes distinctes sont définies pour chacune des configurations.

- GetProfiles

La méthode comportant le même nom permet d'extraire tous les profils multimédias du dispositif. Une seconde méthode appelée GetProfile permet d'extraire un profil dédié.

6.14.3 Liste des configurations

- Get<entity>Configuration

Trois méthodes sont mises en oeuvre pour chacune de ces méthodes. Il s'agit des méthodes

- Get<entity>Configurations

Extraction de toutes les configurations de l'entité de type.

- Get<entity>Configuration

Extraction d'une configuration spécifiée par son jeton de référence.

- GetCompatible<entity>Configurations

Extraction de toutes les configurations compatibles avec la définition du profil actuel.

6.14.4 Masques de confidentialité

L'ensemble associé d'opérations n'est pas pris en charge.

7 Service d'imagerie

7.1 Généralités

Le service d'imagerie comporte des opérations utilisées pour contrôler et configurer les propriétés d'imagerie sur un dispositif. Il convient qu'un dispositif comportant une ou plusieurs sources vidéo prenne en charge le service d'imagerie tel que défini en l'Article F.2. Les réglages d'imagerie font partie de l'entité VideoSource. Cela signifie que les paramètres d'imagerie altèrent directement une source vidéo spécifique.

7.2 Réglages d'imagerie

7.2.1 Paramètres

Le service d'imagerie comporte des opérations d'obtention ou de définition des paramètres d'imagerie et des plages valides pour ces paramètres. Certains paramètres n'ont aucun effet si un mode spécifique n'est pas défini. Certains des paramètres inclus dans les réglages exigent une fonctionnalité d'imagerie spécifique qui peut être demandée au moyen de la commande GetOptions. Les réglages suivants sont disponibles par l'intermédiaire des opérations de service d'imagerie:

BacklightCompensation: Active/désactive le mode BLC (actif/inactif)

- Actif
 - Paramètre de niveau facultatif (unité non spécifiée).
- Inactif

Brightness: Ajuste la luminosité de l'image (unité non spécifiée).

ColorSaturation: Ajuste la saturation de couleur de l'image (unité non spécifiée).

Sharpness: Ajuste la netteté dans l'image (unité non spécifiée).

Contrast: Ajuste le contraste de l'image (unité non spécifiée).

Exposure:

- Auto – Active l'algorithme d'exposition sur le dispositif:
 - Priority – Définit le mode de priorité d'exposition (faible bruit/taux de trame).
 - Window – Masque d'exposition rectangulaire.
 - Min/MaxExposureTime – Plage de temps d'exposition que l'algorithme est autorisé à utiliser.
 - Min/MaxGain – Plage de gain de capteur que l'algorithme est autorisé à utiliser.
 - Min/MaxIris – Plage de diaphragme que l'algorithme est autorisé à utiliser.
- Manual – Désactive l'algorithme d'exposition sur le dispositif:
 - ExposureTime – Temps d'exposition fixe utilisé par le capteur d'image (μs).
 - Gain – Gain fixe utilisé par le capteur d'image (dB).
 - Iris – Atténuation fixe de la lumière d'entrée altérée par le diaphragme (dB). 0dB correspond à un diaphragme totalement ouvert.

Focus:

- Auto (paramètres qui s'appliquent au mode automatique uniquement):
 - Near/FarLimit – Limites pour la lentille de mise au point (m).
- Manuel (paramètres qui s'appliquent au mode manuel uniquement):
 - Default speed – Vitesse par défaut pour l'opération de déplacement de mise au point (en l'absence du paramètre de vitesse). Une commande manuelle est effectuée par l'intermédiaire de la commande de mouvement (voir 7.4).

IrCutFilter: Bascule l'état de filtre de coupure IR entre Actif, Inactif et Auto. L'état Auto laisse l'algorithme d'exposition décider s'il convient d'activer/désactiver le filtre de coupure IR.

- Actif: active le filtre de coupure IR. Mode jour typique.
- Inactif: désactive le filtre de coupure IR. Mode nuit typique.
- Auto: Activation automatique du filtre de coupure IR par le dispositif.

IrCutFilterAutoAdjustment: Paramètres facultatifs appliqués uniquement au mode automatique afin d'ajuster la synchronisation du basculement du filtre de coupure IR.

- BoundaryType – Spécifie les limites de basculement automatique du filtre de coupure IR auxquelles s'appliquent les paramètres suivants:
 - Common: Appliqué aux deux limites de basculement automatique du filtre de coupure IR sur le mode actif et inactif
 - ToOn/ToOff: Appliqué individuellement à l'une des limites de basculement automatique du filtre de coupure IR sur le mode actif/inactif
- BoundaryOffset – Ajuste le niveau d'exposition aux limites pour le basculement du filtre de coupure IR sur le mode actif/inactif spécifié, avec une valeur normalisée sans unité comprise entre +1,0 et -1,0. Zéro représente la valeur par défaut et -1,0 représente l'ajustement le plus sombre (sans unité).

- **ResponseTime** – Délai de basculement du filtre de coupure IR sur le mode actif/inactif après croisement des niveaux d'exposition aux limites.

Whitebalance:

- Mode d'équilibrage automatique des blancs (auto/manuel).
- Manuel (paramètres qui s'appliquent au mode manuel uniquement):
 - Rgain (sans unité).
 - Bgain (sans unité).

WideDynamicRange: Plage dynamique étendue (actif/inactif):

- Actif
 - Paramètre de niveau facultatif (sans unité).
- Inactif

Image stabilization: Active/désactive la fonctionnalité de stabilisation d'image (actif/inactif/auto):

- Actif: Active la fonctionnalité Image Stabilization.
 - Paramètre de niveau facultatif (unité non spécifiée).
- Inactif: Désactive la fonctionnalité Image Stabilization.
- Auto: Activation automatique de la fonctionnalité Image Stabilization par le dispositif

ToneCompensation: Active/désactive le mode Tone Compensation (Compensation de ton).

- Actif: Active la fonctionnalité Tone Compensation.
 - Paramètre de niveau facultatif – Paramètre permettant d'ajuster le niveau de compensation spécifié avec une valeur normalisée comprise entre 0,0 et +1,0 (sans unité).
- Inactif: Désactive la fonctionnalité Tone Compensation.
- Auto: Active la fonctionnalité Tone Compensation avec ajustement automatique du niveau.

Defogging: Active/désactive la fonction Defogging (Désembuage).

- Actif: Active la fonctionnalité Defogging.
 - Paramètre de niveau facultatif – Paramètre permettant d'ajuster le niveau de compensation spécifié avec une valeur normalisée comprise entre 0,0 et +1,0 (sans unité).
- Inactif: Désactive la fonctionnalité Defogging.
- Auto: Active la fonctionnalité Defogging avec ajustement automatique du niveau.

Noise Reduction: Ajuste le niveau de réduction du bruit de l'image avec une valeur normalisée comprise entre 0,0 et +1,0 (sans unité). Niveau=0 signifie aucune réduction du bruit ou une réduction du bruit minimale.

7.2.2 GetImagingSettings

Cette opération demande les réglages d'imagerie d'une source vidéo sur le dispositif. Un dispositif qui met en oeuvre le service d'imagerie doit prendre en charge cette commande.

DEMANDE:

- **VideoSourceToken [tt:ReferenceToken]**
Référence à la source vidéo pour laquelle les réglages d'imagerie sont demandés.

RÉPONSE:

- **ImagingSettings [tt:ImagingSettings20]**
Réglages d'imagerie pour la source vidéo tels que définis en 7.2.1.

DÉFAUTS:

- **env:Sender – ter:InvalidArgVal – terNoSource**
La source vidéo demandée n'existe pas.
- **env:Receiver – ter:ActionNotSupported – ter:NoImagingForSource**
La source vidéo demandée ne prend pas en charge les réglages d'imagerie.

CLASSE D'ACCÈS:

READ_MEDIA

Si la source vidéo prend en charge l'un des réglages d'imagerie définis par le type ImagingSettings dans le Schéma de l'ONVIF, il convient alors que les réglages d'imagerie du dispositif puissent être extraits au moyen de la commande GetImagingSettings.

7.2.3 SetImagingSettings

Cette opération définit les réglages d'imagerie d'une source vidéo sur un dispositif. Un dispositif qui met en oeuvre le service d'imagerie doit prendre en charge cette commande.

Les paramètres de réglages d'imagerie configurables possibles sont décrits en 7.2.1. Des options de réglage sont obtenues au moyen de la commande définie en 7.2.4.

DEMANDE:

- **VideoSourceToken [tt:ReferenceToken]**
Référence à la VideoSource.
- **ImagingSettings [tt:ImagingSettings20]**
Réglages d'imagerie à appliquer.
- **ForcePersistence – facultatif [xs:boolean]**
Indication déconseillée de la persistance ou non de la configuration après redémarrage.

RÉPONSE:

Ce message est vide.

DÉFAUTS:

- **env:Sender – ter:InvalidArgVal – terNoSource**
La VideoSource demandée n'existe pas.
- **env:Receiver – ter:ActionNotSupported- terNoImagingForSource**
La VideoSource demandée ne prend pas en charge les réglages d'imagerie.
- **env:Sender – ter:InvalidArgVal – terSettingsInvalid**
Les réglages demandés sont incorrects.

CLASSE D'ACCÈS:

ACTUATE

7.2.4 GetOptions

Cette opération obtient les plages valides pour les paramètres d'imagerie comportant des plages spécifiques au dispositif. Un dispositif qui met en oeuvre le service d'imagerie doit prendre en charge cette commande. La commande doit retourner tous les paramètres pris en charge et leurs plages de sorte qu'ils puissent être appliqués à la commande SetImagingSettings.

Pour les paramètres en lecture seule qui ne peuvent pas être modifiés par la commande SetImagingSettings, une seule option unique ou des valeurs Min et Max identiques doivent être fournies.

DEMANDE:

- **VideoSourceToken [tt:ReferenceToken]**

Référence à la VideoSource

RÉPONSE:

- **ImagingOptions [tt:ImagingOptions20]**

Plages valides pour les paramètres d'imagerie qui sont classés comme spécifiques au dispositif.

DÉFAUTS:

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

La VideoSource demandée n'existe pas.

- **env:Receiver – ter:ActionNotSupported- terNoImagingForSource**

La VideoSource demandée ne prend pas en charge les réglages d'imagerie.

CLASSE D'ACCÈS:

READ_MEDIA

7.3 Préréglages d'imagerie

7.3.1 Généralités

Les fabricants de sources vidéo peuvent proposer des configurations d'imagerie préréglées qui contiennent différents réglages de paramètres, accordés pour des performances d'imagerie optimales dans des conditions de scènes données. La fonctionnalité Imaging Presets (Préréglages d'imagerie) est analogue aux fonctionnalités que proposent les appareils photographiques actuels à l'utilisateur non professionnel, afin de pouvoir obtenir facilement des images optimales dans des conditions de scènes données (par exemple, Portrait, Paysage, Sports, Macro, Silhouette, etc.).

Trois commandes activent l'utilisation de la fonctionnalité Imaging Presets.

7.3.2 GetPresets

La commande GetPresets demande la liste prédéfinie actuelle de paramètres Imaging Settings (Préréglages) proposée par le fabricant pour une source vidéo donnée.

Il en résulte une liste contenant les paramètres Imaging Presets (Préréglages d'imagerie) disponibles. Si le dispositif ne propose aucune prise en charge des paramètres Imaging Presets, GetPresetsResponse doit retourner une liste vide.

DEMANDE:

- **VideoSourceToken**

Jeton de référence de la VideoSource pour laquelle il convient de demander les paramètres Imaging Presets disponibles.

RÉPONSE:

- **Preset – non limité [timg:ImagingPreset]**

Liste disponible des préréglages d'imagerie pris en charge par le dispositif. Tout dispositif comportant une fonctionnalité Imaging Presets doit proposer au moins un paramètre Imaging Preset. Si le dispositif ne propose aucune prise en charge des paramètres Imaging Presets, GetPresetsResponse doit retourner une liste vide.

DÉFAUTS:

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

La VideoSource demandée n'existe pas.

CLASSE D'ACCÈS:

READ_MEDIA

7.3.3 GetCurrentPreset

La commande GetCurrentPreset doit demander le paramètre Imaging Preset actuellement appliqué à la source vidéo spécifiée, c'est-à-dire qu'elle doit demander quel est l'ensemble appliqué en dernier à la source vidéo parmi les ensembles prédéfinis de paramètres Imaging Settings (et/ou Thermal Settings (Réglages thermiques)). Il en résulte le paramètre Imaging Preset actuel.

Si les paramètres Imaging Presets ne sont pas pris en charge, ou si la configuration de source vidéo ne correspond à aucun des paramètres Imaging Presets existants, le résultat de la commande GetCurrentPreset doit être la valeur Empty (Vide).

DEMANDE:

- **VideoSourceToken [tt:ReferenceToken]**

Jeton de référence de la VideoSource pour laquelle il convient de demander les paramètres Imaging Presets disponibles.

RÉPONSE:

- **CurrentPreset – non limité [timg:ImagingPreset]**

Paramètre Imaging Preset actuel (c'est-à-dire le dernier appliqué) effectivement appliqué à la source vidéo spécifiée. La valeur retournée est vide en l'absence de toute fonctionnalité ou si la configuration actuelle ne correspond à aucun paramètre Imaging Preset existant.

DÉFAUTS:

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

La VideoSource demandée n'existe pas.

CLASSE D'ACCÈS:

READ_MEDIA

7.3.4 SetCurrentPreset

La commande SetCurrentPreset doit demander d'appliquer un paramètre Imaging Preset donné à la source vidéo spécifiée. SetCurrentPreset doit être disponible uniquement pour les sources vidéo comportant une fonctionnalité Imaging Presets.

Les paramètres Imaging Presets sont définis par le fabricant, et sont proposés comme outil de simplification des ajustements des paramètres Imaging Settings pour des types de scène spécifiques. Lorsque le nouveau paramètre Imaging Preset est appliqué par la commande SetCurrentPreset, en réponse, le dispositif doit ajuster les réglages de la source vidéo afin de correspondre aux valeurs définies par le paramètre Imaging Preset spécifié.

DEMANDE:

- **VideoSourceToken [tt:ReferenceToken]**

Jeton de référence de la VideoSource pour laquelle il convient de définir les paramètres Imaging Presets

- **PresetToken [tt:ReferenceToken]**

Prééglage à appliquer.

RÉPONSE:

Ce message est vide.

DÉFAUTS:

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

La VideoSource demandée n'existe pas.

CLASSE D'ACCÈS:

ACTUATE

7.4 Opérations de mise au point

7.4.1 Move

La commande Move (Déplacement) déplace la lentille de mise au point de manière absolue, relative ou continue depuis sa position actuelle. L'argument de vitesse est facultatif pour les commandes absolue et relative, mais exigé pour le mode continu. Si aucun argument de vitesse n'est utilisé, la vitesse par défaut est utilisée. Les ajustements de mise au point par l'intermédiaire de cette opération désactivent la mise au point automatique. Il convient qu'un dispositif avec prise en charge de la commande de mise au point à distance prenne en charge la commande absolue, relative ou continue par l'intermédiaire de l'opération de déplacement. Les MoveOptions prises en charge sont indiquées au moyen de la commande GetMoveOptions.

Au moins une fonctionnalité de commande de mise au point est exigée pour que cette opération soit fonctionnelle.

L'opération de déplacement contient les commandes suivantes:

Absolute – Exige le paramètre de position et accepte éventuellement un argument de vitesse. Un type sans unité est utilisé par défaut pour le positionnement et la vitesse de mise au point. Éventuellement, la position peut être demandée en unités m^{-1} , si elle est prise en charge.

Relative – Exige le paramètre de distance et accepte éventuellement un argument de vitesse. Une distance négative désigne une direction négative.

Continuous – Exige un argument de vitesse. Un argument de vitesse négative signifie une direction négative.

DEMANDE:

- **VideoSourceToken [tt:ReferenceToken]**
Jeton de référence à la VideoSource.
- **Focus [tt:FocusMove]**
Opération de mise au point.

RÉPONSE:

Ce message est vide.

DÉFAUTS:

- **env:Sender – ter:InvalidArgVal – terNoSource**
La VideoSource demandée n'existe pas.
- **env:Receiver – ter:ActionNotSupported- terNoImagingForSource**
La VideoSource demandée ne prend pas en charge les réglages d'imagerie.

CLASSE D'ACCÈS:

ACTUATE

7.4.2 GetMoveOptions

La commande GetMoveOptions permet d'extraire les options de déplacement de lentille de mise au point à utiliser dans la commande de déplacement telle que définie en 7.4. Un dispositif qui prend en charge le service d'imagerie doit prendre en charge la commande GetMoveOptions. La réponse à la commande doit inclure toutes les opérations de déplacement prises en charge. Si l'option de déplacement de la lentille de mise au point n'est pas du tout prise en charge, la réponse doit être vide.

DEMANDE:

- **VideoSourceToken [tt:ReferenceToken]**
Jeton de référence à la VideoSource.

RÉPONSE:

- **MoveOptions [tt:MoveOptions20]**
Plages valides pour la commande de déplacement de lentille de mise au point.

DÉFAUTS:

- **env:Sender – ter:InvalidArgVal – terNoSource**
La VideoSource demandée n'existe pas.
- **env:Receiver – ter:ActionNotSupported- terNoImagingForSource**
La VideoSource demandée ne prend pas en charge les réglages d'imagerie.

CLASSE D'ACCÈS:

READ_MEDIA

7.4.3 Stop

La commande Stop (Interruption) interrompt tous les mouvements de mise au point en cours de la lentille. Il convient qu'un dispositif qui prend en charge le contrôle de mise au point à distance tel qu'indiqué au moyen des GetMoveOptions prenne en charge cette commande. Cette opération n'altère pas une mise au point automatique en cours.

DEMANDE:

- **VideoSourceToken [tt:ReferenceToken]**

Jeton de référence à la VideoSource.

RÉPONSE:

Ce message est vide.

DÉFAUTS:

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

La VideoSource demandée n'existe pas.

- **env:Receiver – ter:ActionNotSupported- terNoImagingForSource**

La VideoSource demandée ne prend pas en charge les réglages d'imagerie.

CLASSE D'ACCÈS:

ACTUATE

7.4.4 GetImagingStatus

La commande GetStatus demande le statut d'imagerie actuel du dispositif. Un dispositif qui prend en charge le contrôle de mise au point à distance tel qu'indiqué au moyen des GetMoveOptions doit prendre en charge cette commande.

Le statut d'imagerie contient:

- La position de mise au point, le statut de déplacement et les informations d'erreur.
 - La position de mise au point est représentée par un type sans unité.
 - Le statut de déplacement peut être dans un état MOVING, IDLE ou UNKNOWN.
 - Les informations d'erreur sont présentées sous la forme d'une chaîne (une erreur de positionnement indiquée par le matériel, par exemple).

DEMANDE:

- **VideoSourceToken [tt:ReferenceToken]**

Jeton de référence à la VideoSource.

RÉPONSE:

- **ImagingStatus [tt:ImagingStatus20]**

Informations du statut d'imagerie.

DÉFAUTS:

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

La VideoSource demandée n'existe pas.

- **env:Receiver – ter:ActionNotSupported- terNoImagingForSource**

La VideoSource demandée ne prend pas en charge les réglages d'imagerie.

CLASSE D'ACCÈS:

READ_MEDIA

7.5 Fonctionnalités

Les fonctionnalités reflètent les fonctions facultatives et la fonctionnalité d'un service. Les informations sont statiques et ne varient pas pendant le fonctionnement du dispositif. Les fonctionnalités suivantes sont disponibles:

ImageStabilization: Indique si la fonctionnalité ImageStabilization est prise en charge ou non.

ImagingPresets: Indique si la fonctionnalité ImagingPresets est prise en charge ou non.

DEMANDE:

Ce message est vide.

RÉPONSE:

- **Capabilities [timg:Capabilities]**

Ensemble d'indicateurs pour les groupes de fonctions tels que décrits ci-dessus.

DÉFAUTS:

Aucun

CLASSE D'ACCÈS:

PRE_AUTH

8 Service PTZ

8.1 Généralités

Le service PTZ fournit des interfaces pour la configuration et le fonctionnement des caméras mobiles également appelées dômes. Pour la définition du schéma, voir l'Article F.5.

8.2 Nœud PTZ

8.2.1 Généralités

Un dispositif comportant la fonctionnalité PTZ peut avoir plusieurs nœuds PTZ. Les nœuds PTZ peuvent représenter des pilotes PTZ mécaniques, téléchargés ou numériques. Les nœuds PTZ sont les entités de plus bas niveau dans l'API de contrôle PTZ et reflètent les fonctionnalités PTZ prises en charge. Le nœud PTZ est référencé par son nom ou par son jeton de référence. Le service PTZ ne comporte pas d'opérations de création ou de manipulation des nœuds PTZ.

Les propriétés suivantes doivent être disponibles pour tous les nœuds PTZ:

- Token – Identifiant unique utilisé pour référencer des nœuds PTZ.
- Name – Nom donné par l'installateur.
- SupportedPTZSpaces – Liste des systèmes de coordonnées disponibles pour le nœud PTZ. Pour chaque système de coordonnées, le nœud PTZ doit spécifier sa plage autorisée.
- MaximumNumberOfPresets – Toutes les opérations prédéfinies doivent être disponibles pour ce nœud PTZ si un pré réglage est pris en charge.

- **HomeSupported** – Opérateur booléen spécifiant la disponibilité d'une position de départ. Si sa valeur est "true", les opérations relatives à la position de départ (Home Position) doivent être disponibles pour ce nœud PTZ.
- **AuxiliaryCommands** – Liste des commandes auxiliaires prises en charge. Si la liste n'est pas vide, les opérations auxiliaires doivent être disponibles pour ce nœud PTZ. Un dispositif peut utiliser les commandes auxiliaires qui sont décrites dans l'IEC 60839-11-31.
- **MaximumNumberOfPresetTours** – Indique le nombre de tours pré-réglés qui peuvent être créés. Les opérations de tours pré-réglés exigées doivent être disponibles pour ce nœud PTZ si un ou plusieurs tours pré-réglés sont pris en charge.

8.2.2 GetNodes

Un dispositif comportant la fonctionnalité PTZ doit mettre en œuvre cette opération et retourner tous les nœuds PTZ disponibles sur le dispositif.

DEMANDE:

Ceci est un message vide.

RÉPONSE:

- **PTZNode – facultatif, non limité [tt:PTZNode]**

Liste des nœuds PTZ existants sur le dispositif.

DÉFAUTS:

Aucun

CLASSE D'ACCÈS:

READ_MEDIA

8.2.3 GetNode

Un dispositif comportant la fonctionnalité PTZ doit mettre en œuvre l'opération GetNode et retourner les propriétés du nœud PTZ demandé, s'il existe. Sinon, le dispositif doit répondre par un message de défaut approprié.

DEMANDE:

- **NodeToken [tt:ReferenceToken]**

Référence au nœud PTZ demandé.

RÉPONSE:

- **PTZNode [tt:PTZNode]**

Propriétés de nœud PTZ demandées.

DÉFAUTS:

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

Le nœud PTZ n'est pas présent sur le dispositif.

CLASSE D'ACCÈS:

READ_MEDIA

8.3 Configuration PTZ

8.3.1 Généralités

La PTZConfiguration contient une référence au nœud PTZ auquel elle appartient. Cette référence ne peut pas être modifiée par un client.

Les éléments suivants font partie de la configuration PTZ:

- PTZNodeToken – Référence obligatoire au nœud PTZ auquel la configuration PTZ appartient.
- DefaultAbsolutePanTiltPositionSpace – Si le nœud PTZ prend en charge les mouvements de panoramique/inclinaison absolus, il doit spécifier un espace de position panoramique/inclinaison absolu par défaut.
- DefaultRelativePanTiltTranslationSpace – Si le nœud PTZ prend en charge les mouvements de panoramique/inclinaison relatifs, il doit spécifier un espace de translation panoramique/inclinaison relatif par défaut.
- DefaultContinuousPanTiltVelocitySpace – Si le nœud PTZ prend en charge les mouvements de panoramique/inclinaison continus, il doit spécifier un espace de vitesse de panoramique/inclinaison continu par défaut.
- DefaultPanTiltSpeedSpace – Si le nœud PTZ prend en charge les mouvements absolus ou relatifs, il doit spécifier un espace de vitesse de panoramique/inclinaison par défaut.
- DefaultAbsoluteZoomPositionSpace – Si le nœud PTZ prend en charge les mouvements de zoom absolus, il doit spécifier un espace de position de zoom absolu par défaut.
- DefaultRelativeZoomTranslationSpace – Si le nœud PTZ prend en charge les mouvements de zoom relatifs, il doit spécifier un espace de translation de zoom relatif par défaut.
- DefaultContinuousZoomVelocitySpace – Si le nœud PTZ prend en charge les mouvements de zoom continus, il doit spécifier un espace de vitesse de zoom continu par défaut.
- DefaultPTZSpeed – Si le nœud PTZ prend en charge les mouvements PTZ absolus ou relatifs, il doit spécifier des vitesses de panoramique/inclinaison et de zoom par défaut correspondantes.
- DefaultPTZTimeout – Si le nœud PTZ prend en charge des mouvements continus, il doit spécifier un délai d'attente par défaut, au-delà duquel le mouvement s'interrompt.
- PanTiltLimits – Il convient que l'élément de limites de panoramique/inclinaison soit présent pour un nœud PTZ qui prend en charge un mouvement de panoramique/inclinaison absolu. Si cet élément est présent, il indique la prise en charge de limites de panoramique/inclinaison configurables. Si les limites sont activées, les mouvements de panoramique/inclinaison doivent toujours rester dans la plage spécifiée. Les limites de panoramique/inclinaison sont désactivées en définissant les limites à $-\text{INF}$ ou $+\text{INF}$.
- ZoomLimits – Il convient que l'élément de limites de zoom soit présent pour un nœud PTZ qui prend en charge un zoom absolu. Si l'élément est présent, il indique la prise en charge de limites de zoom configurables. Si les limites sont activées, les mouvements de zoom doivent toujours rester dans la plage spécifiée. Les limites de zoom sont désactivées en définissant les limites à $-\text{INF}$ et $+\text{INF}$.
- MoveRamp – Rampe d'accélération facultative utilisée par le dispositif lors de son déplacement.
- PresetRamp – Rampe d'accélération facultative utilisée par le dispositif lors du rappel des préséglages.
- PresetTourRamp – Rampe d'accélération facultative utilisée par le dispositif lors de l'exécution de PresetTours.

Les espaces de position/translation/vitesse par défaut sont introduits afin de permettre aux clients d'envoyer des demandes de déplacement sans qu'il soit nécessaire de spécifier un certain système de coordonnées. Les vitesses par défaut sont introduites pour contrôler la vitesse des demandes de déplacement (absolu, relatif, prédéfini) si aucune vitesse explicite n'a été définie.

L'intervalle de panoramique et inclinaison admis pour les limites de panoramique/inclinaison est défini par une plage d'espace bidimensionnel qui correspond à un espace de position de panoramique/inclinaison absolu spécifique. Au moins un espace de panoramique/inclinaison est exigé par le PTZNode pour prendre en charge les limites de panoramique/inclinaison. Les limites s'appliquent à tous les mouvements de panoramique/inclinaison absolus, relatifs et continus pris en charge. Les limites doivent être vérifiées dans le système de coordonnées pour lequel elles ont été spécifiées. Dans la pratique, cela signifie que même si les mouvements sont spécifiés dans un système de coordonnées différent, les mouvements demandés doivent être transformés dans le système de coordonnées des limites dans lequel ces dernières peuvent être contrôlées. Lorsqu'un mouvement relatif ou continu est spécifié, qui est susceptible de sortir des limites spécifiées, l'unité PTZ doit se déplacer selon les limites spécifiées. Les limites de zoom doivent être interprétées en conséquence.

8.3.2 GetConfigurations

Un dispositif comportant la fonctionnalité PTZ doit retourner toutes les PTZConfigurations par l'intermédiaire de l'opération GetConfigurations.

DEMANDE:

Ceci est un message vide.

RÉPONSE:

- **PTZConfiguration – facultatif, non limité [tt:PTZConfiguration]**

Liste de toutes les PTZConfigurations existantes sur le dispositif.

DÉFAUTS:

Aucun

CLASSE D'ACCÈS:

READ_MEDIA

8.3.3 GetConfiguration

Un dispositif comportant la fonctionnalité PTZ doit retourner la configuration PTZ demandée, si elle existe, par l'intermédiaire de l'opération GetConfiguration.

DEMANDE:

- **ConfigurationToken [tt:ReferenceToken]**
Référence à la configuration PTZ demandée.

RÉPONSE:

- **PTZConfiguration[tt:PTZConfiguration]**
Configuration de nœud PTZ demandée.

DÉFAUTS:

- **env:Sender – ter:InvalidArgVal – ter:NoConfig**
La configuration demandée n'existe pas.

CLASSE D'ACCÈS:

READ_MEDIA

8.3.4 GetConfigurationOptions

Un dispositif comportant la fonctionnalité PTZ doit mettre en œuvre l'opération GetConfigurationOptions. Celle-ci retourne la liste des systèmes de coordonnées pris en charge comprenant leurs limitations d'intervalle. Par conséquent, les options peuvent différer suivant que la configuration PTZ est assignée à un profil (voir 6.3.6) contenant une VideoSourceConfiguration. Dans ce cas, les options peuvent contenir des systèmes de coordonnées supplémentaires faisant référence au système de coordonnées d'image décrit par la VideoSourceConfiguration. Chaque système de coordonnées répertorié appartient à l'un des groupes répertoriés en 8.3.1. Si le nœud PTZ prend en charge les mouvements continus, il doit retourner un intervalle de délai d'attente dans lequel les délais d'attente sont acceptés par le nœud PTZ.

DEMANDE:

- **ConfigurationToken [tt:ReferenceToken]**
Référence à la configuration PTZ demandée.

RÉPONSE:

- **PTZConfigurationOptions[tt:PTZConfigurationOptions]**
Options de configuration de nœud PTZ demandée.

DÉFAUTS:

- **env:Sender – ter:InvalidArgVal – ter:NoConfig**
La configuration demandée n'existe pas.

CLASSE D'ACCÈS:

READ_MEDIA

8.3.5 SetConfiguration

Un dispositif comportant la fonctionnalité PTZ doit mettre en œuvre l'opération SetConfiguration. Le drapeau ForcePersistence indique si les modifications persistent après le redémarrage du dispositif.

DEMANDE:

- **ConfigurationToken [tt:ReferenceToken]**
Référence à la configuration PTZ à modifier.
- **PTZConfigurationOptions[tt:PTZConfigurationOptions]**
Options de configuration de nœud PTZ demandée.
- **ForcePersistence [xs:boolean]**
Modificateur déconseillé des réglages temporaires si prise en charge par le dispositif.

RÉPONSE:

Ceci est un message vide.

DÉFAUTS:

- **env:Sender – ter:InvalidArgVal – ter:NoConfig**
La configuration demandée n'existe pas.
- **env:Sender – ter:InvalidArgVal – ter:ConfigModify**
Les paramètres de configuration ne peuvent pas être définis.
- **env:Receiver – ter:Action – ter:ConfigurationConflict**
Les nouveaux réglages entrent en conflit avec d'autres utilisations de la configuration.

CLASSE D'ACCÈS:

ACTUATE**8.3.6 GetCompatibleConfigurations**

Un dispositif qui indique la prise en charge des GetCompatibleConfigurations par la fonctionnalité GetCompatibleConfigurations doit retourner toutes les PTZConfigurations disponibles qui peuvent être ajoutées au profil multimédia référencé par l'intermédiaire de l'opération GetCompatibleConfigurations.

Il convient qu'un dispositif qui propose deux ou plusieurs PTZConfiguration ou deux ou plusieurs VideoSourceConfiguration ou qui comprend toute autre interdépendance de ressources entre les entités PTZConfiguration et les autres ressources qui peuvent être répertoriées dans un profil multimédia mette en oeuvre cette opération. Les entités PTZConfiguration retournées par cette opération doivent pouvoir les ajouter avec succès au profil multimédia référencé.

DEMANDE:

- **ProfileToken [tt:ReferenceToken]**
Jeton d'un profil multimédia existant.

RÉPONSE:

- **PTZConfiguration – facultatif, non limité [tt:PTZConfiguration]**
Liste des configurations de noeuds PTZ compatibles.

DÉFAUTS:

- **env:Receiver – ter:ActionNotSupported – ter:OptionalActionNotSupported**
Cette méthode n'est pas mise en oeuvre par le dispositif.

CLASSE D'ACCÈS:

READ_MEDIA**8.4 Opérations de déplacement****8.4.1 Généralités**

Le présent paragraphe décrit trois opérations de mouvement de l'unité PTZ: absolu, relatif ou continu. Toutes les opérations exigent un *ProfileToken* référençant un profil multimédia comprenant une PTZConfiguration.

Toutes les commandes de déplacement doivent être mises en oeuvre de manière à être non bloquantes, ce qui signifie qu'elles ne doivent pas attendre que l'opération de déplacement demandée soit terminée. La dernière opération de déplacement peut être remplacée par l'envoi d'une autre demande de déplacement.

En raison de la large gamme de dispositifs physiques dont traite la présente spécification, elle n'exige pas un temps de réponse spécifique aux opérations de déplacement PTZ. Cependant, un dispositif doit rechercher une latence minimale concernant le temps de demande-réponse des commandes de déplacement PTZ. Noter que le dispositif contrôle le délai des commandes et du codage vidéo; le délai du réseau et celui du client altèrent également l'expérience de l'utilisateur, mais ne peuvent pas être contrôlés par le dispositif.

8.4.2 AbsoluteMove

Si un nœud PTZ prend en charge des mouvements de panoramique/inclinaison absolus ou de zoom absolus, il doit prendre en charge l'opération AbsoluteMove. L'argument de position de cette commande spécifie la position absolue vers laquelle l'unité PTZ se déplace. Il comporte un élément de panoramique/inclinaison facultatif et un élément de zoom également facultatif. Si la position de panoramique/inclinaison est omise, le mouvement de panoramique/inclinaison actuel ne doit pas être altéré par cette commande. Le même principe s'applique pour la position de zoom.

Les espaces référencés dans la position doivent être des espaces de position absolue pris en charge par le nœud PTZ. Si l'information d'espace est omise, les espaces par défaut correspondants de la configuration PTZ, qui font partie du profil multimédia spécifié, sont utilisés. Un dispositif peut prendre en charge des mouvements de panoramique/inclinaison absolus, des mouvements de zoom absolus ou des mouvements non absolus en fournissant uniquement des espaces de position absolue pour les cas pris en charge.

Un argument Speed (Vitesse) existant neutralise la DefaultSpeed de la configuration PTZ correspondante pendant le mouvement vers la position demandée. Si des espaces sont référencés dans l'argument Speed, il doit s'agir d'espaces de vitesse pris en charge par le nœud PTZ.

L'opération doit échouer si la position absolue demandée ne peut pas être atteinte.

DEMANDE:

- **ProfileToken [tt:ReferenceToken]**
Référence à un profil multimédia existant.
- **Position [tt:PTZVector]**
Vecteur de spécification de la position cible absolue.
- **Speed – facultatif [tt:PTZSpeed]**
Vecteur de vitesse facultatif.

RÉPONSE:

Ceci est un message vide.

DÉFAUTS:

- **env:Sender – ter:InvalidArgVal – ter:NoProfile**
Le jeton de profil (ProfileToken) demandé n'existe pas.
- **env:Sender – ter:InvalidArgVal – ter:NoPTZProfile**
Le jeton de profil demandé ne référence pas une configuration PTZ.
- **env:Sender – ter:InvalidArgVal – ter:SpaceNotSupported**
Un espace est référencé, mais n'est pas pris en charge par le nœud PTZ.
- **env:Sender – ter:InvalidArgVal – ter:InvalidPosition**
La position demandée est hors limites.
- **env:Sender – ter:InvalidArgVal – ter:InvalidSpeed**
La vitesse demandée est hors limites.

CLASSE D'ACCÈS:

ACTUATE**8.4.3 RelativeMove**

Si un nœud PTZ prend en charge les mouvements de panoramique/inclinaison relatifs ou de zoom relatifs, il doit alors prendre en charge l'opération RelativeMove. L'argument de translation de cette opération spécifie la différence entre la position actuelle et la position à laquelle le dispositif PTZ a reçu l'instruction de se déplacer. L'opération est divisée en un élément de panoramique/inclinaison facultatif et un élément de zoom facultatif. Si l'élément de panoramique/inclinaison est omis, le mouvement de panoramique/inclinaison actuel ne doit pas être altéré par cette commande. Le même principe s'applique pour l'élément de zoom.

Les espaces référencés dans l'élément de translation doivent être des espaces de translation pris en charge par le nœud PTZ. Si l'information d'espace est omise pour l'argument de translation, les espaces par défaut correspondants de la configuration PTZ, qui font partie du profil multimédia spécifié, sont utilisés. Un dispositif peut prendre en charge des mouvements de panoramique/inclinaison relatifs, des mouvements de zoom relatifs ou des mouvements non relatifs en fournissant uniquement des espaces de translation pour les cas pris en charge.

Un argument de vitesse existant neutralise la DefaultSpeed de la configuration PTZ correspondante pendant le mouvement de translation demandé. Si des espaces sont référencés dans l'argument de vitesse, il doit s'agir d'espaces de vitesse pris en charge par le nœud PTZ.

La commande peut être utilisée pour interrompre l'unité PTZ à sa position actuelle en envoyant des valeurs nulles pour les mouvements de panoramique/inclinaison et de zoom. L'interruption doit avoir exactement le même effet, quel que soit l'espace relatif référencé.

Si la translation demandée conduit à une position absolue qui ne peut pas être atteinte, le nœud PTZ doit se déplacer vers une position qui peut être atteinte selon la limite des positions valides.

DEMANDE:

- **ProfileToken [tt:ReferenceToken]**
Référence à un profil multimédia existant.
- **Translation [tt:PTZVector]**
Vecteur de spécification de la translation de position par rapport à la position actuelle.
- **Speed – facultatif [tt:PTZSpeed]**
Vecteur de vitesse facultatif.

RÉPONSE:

Ceci est un message vide.

DÉFAUTS:

- **env:Sender – ter:InvalidArgVal – ter:NoProfile**
Le jeton de profil (ProfileToken) demandé n'existe pas.
- **env:Sender – ter:InvalidArgVal – ter:NoPTZProfile**
Le jeton de profil demandé ne référence pas une configuration PTZ.
- **env:Sender – ter:InvalidArgVal – ter:SpaceNotSupported**
Un espace est référencé, mais n'est pas pris en charge par le nœud PTZ.
- **env:Sender – ter:InvalidArgVal – ter:InvalidTranslation**
La translation demandée est hors limites.
- **env:Sender – ter:InvalidArgVal – ter:InvalidSpeed**
La vitesse demandée est hors limites.

CLASSE D'ACCÈS:**ACTUATE****8.4.4 ContinuousMove**

Un dispositif comportant la fonctionnalité PTZ doit prendre en charge les mouvements continus. L'argument de vitesse de cette commande spécifie une valeur de vitesse signée pour les mouvements de panoramique, d'inclinaison et de zoom. L'élément de panoramique/inclinaison combiné est facultatif et l'élément Zoom lui-même est facultatif. Si l'élément de panoramique/inclinaison est omis, le mouvement de panoramique/inclinaison actuel ne doit pas être altéré par cette commande. Le même principe s'applique pour l'élément de zoom. Les espaces référencés dans l'élément de vitesse doivent être des espaces de vitesse pris en charge par le nœud PTZ. Si l'information d'espace est omise pour l'argument de vitesse, les espaces par défaut correspondants de la configuration PTZ, qui font partie du profil multimédia spécifié, sont utilisés. Un dispositif peut prendre en charge des mouvements de panoramique/inclinaison continus et/ou des mouvements de zoom continus en fournissant uniquement des espaces de vitesse pour les cas pris en charge.

Un argument de délai d'attente existant neutralise le paramètre DefaultPTZTimeout de la configuration PTZ correspondante pendant cette opération de déplacement. Le paramètre de délai d'attente spécifie la durée de déplacement du nœud PTZ.

Un dispositif doit interrompre son mouvement sur un axe particulier (panoramique, inclinaison ou zoom) lorsqu'une valeur nulle est envoyée comme paramètre ContinuousMove pour l'axe concerné. L'interruption doit avoir exactement le même effet, quel que soit l'espace de vitesse référencé. Cette commande a le même effet sur un mouvement continu que la commande d'arrêt décrite en 8.4.6.

Si la vitesse demandée conduit à des positions absolues qui ne peuvent pas être atteintes, le nœud PTZ doit se déplacer vers une position qui peut être atteinte selon la limite de l'intervalle autorisé. Une application typique de l'opération de déplacement continu est le contrôle PTZ à l'aide d'une manette.

DEMANDE:

- **ProfileToken [tt:ReferenceToken]**
Référence à un profil multimédia existant.
- **Velocity [tt:PTZSpeed]**
Vecteur de vitesse qui spécifie la vélocité des mouvements de panoramique, inclinaison et zoom.
- **Timeout – facultatif [xs:duration]**
Délai d'attente facultatif.

RÉPONSE:

Ceci est un message vide.

DÉFAUTS:

- **env:Sender – ter:InvalidArgVal – ter:NoProfile**
Le jeton de profil (ProfileToken) demandé n'existe pas.
- **env:Sender – ter:InvalidArgVal – ter:NoPTZProfile**
Le jeton de profil demandé ne référence pas une configuration PTZ.
- **env:Sender – ter:InvalidArgVal – ter:SpaceNotSupported**
Un espace est référencé, mais n'est pas pris en charge par le nœud PTZ.
- **env:Sender – ter:InvalidArgVal – ter:InvalidTranslation**
La translation demandée est hors limites.
- **env:Sender – ter:InvalidArgVal – ter:TimeoutNotSupported**
L'argument de délai d'attente spécifié ne se situe pas dans l'intervalle de délai d'attente pris en charge.
- **env:Sender – ter:InvalidArgVal – ter:InvalidVelocity**
La vitesse demandée est hors limites.

CLASSE D'ACCÈS:

ACTUATE**8.4.5 GeoMove**

Un dispositif qui indique GeoMove dans l'un de ses nœuds PTZ doit prendre en charge cette commande.

Les paramètres AreaHeight et AreaWidth facultatifs peuvent être ajoutés à la demande, de sorte que le dispositif comportant la fonctionnalité PTZ peut déterminer le facteur zoom en interne. Dans le cas où les paramètres AreaHeight et AreaWidth ne sont pas fournis, l'unité ne modifie pas le zoom. Les paramètres AreaHeight et AreaWidth sont exprimés en mètres.

Un argument de vitesse existant neutralise la DefaultSpeed de la configuration PTZ correspondante pendant le mouvement de translation demandé. Si des espaces sont référencés dans l'argument de vitesse, il doit s'agir d'espaces de vitesse pris en charge par le nœud PTZ.

Si le dispositif comportant la fonctionnalité PTZ ne prend pas en charge l'extraction automatique de la géolocalisation, sa configuration doit être effectuée par la commande SetGeoLocation avant qu'il ne puisse exécuter des commandes géoréférencées. Lorsque le client demande une commande GeoMove avant la configuration de la géolocalisation du dispositif, ce dernier doit retourner une erreur.

Selon la cinématique du dispositif comportant la fonctionnalité PTZ, la position demandée peut ne pas être atteinte. Dans cette situation, le dispositif doit retourner une erreur, en indiquant qu'il ne peut pas exécuter l'action demandée en raison de limites physiques.

DEMANDE:

- **ProfileToken [tt:ReferenceToken]**
Référence à un profil multimédia existant.
- **Target [tt:GeoLocation]**
Coordonnées cibles.
- **Speed – facultatif [tt:PTZSpeed]**
Vecteur de vitesse qui spécifie la vitesse des mouvements de panoramique, inclinaison et zoom.
- **AreaWidth -- facultatif [xs:float]**
Zone facultative à présenter.
- **AreaHeight -- facultatif [xs:float]**
Zone facultative à présenter.

RÉPONSE:

Ceci est un message vide.

DÉFAUTS:

- **env:Sender – ter:InvalidArgVal – ter:NoProfile**
Le jeton de profil (ProfileToken) demandé n'existe pas.
- **env:Sender – ter:InvalidArgVal – ter:NoPTZProfile**
Le jeton de profil demandé ne référence pas une configuration PTZ.
- **env:Sender – ter:InvalidArgVal – ter:GeoMoveNotSupported**
Le dispositif ne prend pas en charge le déplacement géolocalisé.
- **env:Sender – ter:InvalidArgVal – ter:UnreachablePosition**
La translation demandée est hors limites.
- **env:Sender – ter:InvalidArgVal – ter:TimeoutNotSupported**
La position demandée ne peut pas être atteinte.
- **env:Sender – ter:InvalidArgVal – ter:GeoLocationUnknown**
L'unité n'est pas capable d'exécuter GeoMove du fait de la non-configuration de sa géolocalisation ou de l'absence de cette dernière.

CLASSE D'ACCÈS:

ACTUATE

8.4.6 Stop (Interruption)

Un dispositif comportant la fonctionnalité PTZ doit prendre en charge l'opération Stop (interruption). Si aucun argument de filtre d'interruption n'est présent, cette commande interrompt tous les mouvements de panoramique, d'inclinaison et de zoom en cours. L'opération Stop peut être filtrée pour interrompre un mouvement spécifique en définissant l'argument d'interruption correspondant.

DEMANDE:

- **ProfileToken [tt:ReferenceToken]**
Référence à un profil multimédia existant.
- **PanTilt – facultatif [xs:boolean]**
Interrompt l'opération de panoramique et inclinaison (valeurs par défaut vraies).
- **Zoom – facultatif [xs:boolean]**
Interrompt l'opération de zoom (valeurs par défaut vraies).

RÉPONSE:

Ceci est un message vide.

DÉFAUTS:

- **env:Sender – ter:InvalidArgVal – ter:NoProfile**
Le jeton de profil (ProfileToken) demandé n'existe pas.
- **env:Sender – ter:InvalidArgVal – ter:NoPTZProfile**
Le jeton de profil demandé ne référence pas une configuration PTZ.

CLASSE D'ACCÈS:

ACTUATE**8.4.7 GetStatus**

Un dispositif comportant la fonctionnalité PTZ doit être capable de rapporter son statut PTZ au moyen de la commande GetStatus. Le statut PTZ contient les informations suivantes:

- Position (facultatif) – spécifie la position absolue de l'unité PTZ avec les références d'espace. Les espaces absolus par défaut de la configuration PTZ correspondante doivent être référencés dans l'élément de position. Cette information doit être présente si le dispositif indique une prise en charge par la fonctionnalité StatusPosition.
- MoveStatus (facultatif) – indique si l'unité de dispositif de panoramique/inclinaison/zoom est actuellement en mouvement, au repos ou dans un état inconnu. Cette information doit être présente si le dispositif indique une prise en charge par la fonctionnalité MoveStatus. L'état Unknown (Inconnu) ne doit pas être utilisé en fonctionnement normal, mais il est en revanche réservé aux conditions d'initialisation ou d'erreur.
- Error (facultatif) – déclare une condition d'erreur PTZ actuelle. Ce champ doit être présent si le MoveStatus indique Unknown.
- UTC Time – spécifie l'heure TUC à laquelle ce statut a été généré.

DEMANDE:

- **ProfileToken [tt:ReferenceToken]**
Référence à un profil multimédia existant.

RÉPONSE:

- **PTZStatus [tt:PTZStatus]**
PTZStatus pour le profil multimédia demandé.

DÉFAUTS:

- **env:Sender – ter:InvalidArgVal – ter:NoProfile**
Le jeton de profil (ProfileToken) demandé n'existe pas.
- **env:Sender – ter:InvalidArgVal – ter:NoPTZProfile**
Le jeton de profil demandé ne référence pas une configuration PTZ.
- **env:Receiver – ter:Action – ter:NoStatus**
Aucun statut PTZ n'est disponible dans le profil multimédia demandé.

CLASSE D'ACCÈS:

READ_MEDIA**8.5 Opérations de préréglage****8.5.1 Généralités**

Le présent article décrit les opérations de gestion des préréglages d'un nœud PTZ. Ces opérations doivent être mises en œuvre pour les nœuds PTZ prenant en charge les préréglages. Toutes les opérations exigent un ProfileToken référant un profil multimédia comprenant une PTZConfiguration.

8.5.2 SetPreset

La commande SetPreset sauvegarde les paramètres de position de dispositif actuelle, de sorte que le dispositif puisse se déplacer vers la position de préréglage sauvegardée par l'intermédiaire de l'opération GotoPreset.

En l'absence du paramètre PresetToken, le dispositif doit créer un nouveau préréglage. À défaut, il doit mettre à jour la position archivée et éventuellement le nom du préréglage donné. Si la création aboutit avec succès, la réponse contient le PresetToken qui identifie de façon unique le préréglage. Un préréglage existant peut être remplacé en spécifiant le PresetToken du préréglage correspondant. Dans les deux cas (remplacement ou création), un PresetName facultatif peut être spécifié. L'opération échoue si le dispositif PTZ se déplace pendant l'opération SetPreset.

Le dispositif peut sauvegarder en interne les états supplémentaires tels que les propriétés d'imagerie dans le préréglage PTZ, qu'il convient de rappeler ensuite dans l'opération GotoPreset. Un dispositif doit accepter une SetPresetRequest valide qui n'inclut pas l'élément PresetName facultatif.

Les dispositifs peuvent exiger des noms de préréglage uniques et refuser une demande contenant un PresetName déjà existant en répondant par le message d'erreur ter:PresetExist.

DEMANDE:

- **ProfileToken [tt:ReferenceToken]**
Référence à un profil multimédia existant.
- **PresetToken – facultatif [tt:ReferenceToken]**
Jeton de préréglage existant facultatif destiné à mettre à jour une position préréglée.
- **PresetName – facultatif [xs:string]**
Nom facultatif à attribuer à la position préréglée.

RÉPONSE:

- **PresetToken [tt:ReferenceToken]**

Jetton de référence attribué au préréglage par le dispositif.

DÉFAUTS:

- **env:Sender – ter:InvalidArgVal – ter:NoProfile**
Le jeton de profil (ProfileToken) demandé n'existe pas.
- **env:Sender – ter:InvalidArgVal – ter:NoPTZProfile**
Référence à un jeton de préréglage existant.
- **env:Sender – ter:InvalidArgVal – ter:PresetExist**
Le nom demandé existe déjà pour un autre préréglage.
- **env:Sender – ter:InvalidArgVal – ter:InvalidPresetName**
Le PresetName est trop long ou contient des caractères non valides.
- **env:Sender – ter:InvalidArgVal – ter:NoToken**
Le jeton de préréglage demandé n'existe pas.
- **env:Receiver – ter:Action – ter:MovingPTZ**
Le préréglage ne peut pas être défini pendant que l'unité PTZ se déplace.
- **env:Receiver – ter:Action – ter:TooManyPresets**
Nombre maximal de préréglages atteint.

CLASSE D'ACCÈS:

ACTUATE**8.5.3 GetPresets**

L'opération GetPresets retourne les préréglages sauvegardés constitués des éléments suivants:

- Token – Identifiant unique utilisé pour référencer le préréglage.
- Name – Nom mnémonique facultatif.
- PTZ Position – position absolue facultative. Si le nœud PTZ prend en charge les espaces de position de panoramique/inclinaison absolus, la position de panoramique/inclinaison doit être spécifiée. Si le nœud PTZ prend en charge les espaces de position de zoom absolus, la position de zoom doit être spécifiée.

DEMANDE:

- **ProfileToken [tt:ReferenceToken]**

Référence à un profil multimédia existant.

RÉPONSE:

- **Preset – facultatif, non limité [tt:PTZPreset]**

Liste des préréglages pour le profil multimédia demandé.

DÉFAUTS:

- **env:Sender – ter:InvalidArgVal – ter:NoProfile**
Le jeton de profil (ProfileToken) demandé n'existe pas.
- **env:Sender – ter:InvalidArgVal – ter:NoPTZProfile**
Le jeton de profil demandé ne référence pas une configuration PTZ.

CLASSE D'ACCÈS:

READ_MEDIA**8.5.4 GotoPreset**

L'opération GotoPreset rappelle un préréglage déjà défini. Si le paramètre de vitesse est omis, la vitesse par défaut de la configuration PTZ correspondante doit être utilisée. Le paramètre de vitesse peut être spécifié uniquement si des espaces de vitesse sont disponibles pour le nœud PTZ. La commande GotoPreset est une opération non bloquante, qui peut être interrompue par d'autres commandes de déplacement.

DEMANDE:

- **ProfileToken [tt:ReferenceToken]**
Référence à un profil multimédia existant.
- **PresetToken [tt:ReferenceToken]**
Référence à un jeton de préréglage existant.
- **Speed – facultatif [PTZSpeed]**
Vitesse facultative.

RÉPONSE:

Ceci est un message vide.

DÉFAUTS:

- **env:Sender – ter:InvalidArgVal – ter:NoProfile**
Le jeton de profil (ProfileToken) demandé n'existe pas.
- **env:Sender – ter:InvalidArgVal – ter:NoPTZProfile**
Référence à un jeton de préréglage existant.
- **env:Sender – ter:InvalidArgVal – ter:NoToken**
Le jeton de préréglage demandé n'existe pas.
- **env:Sender – ter:InvalidArgVal – ter:SpaceNotSupported**
Un espace qui n'est pas pris en charge par le nœud PTZ est référencé.
- **env:Sender – ter:InvalidArgVal – ter:InvalidSpeed**
La vitesse demandée est hors limites.

CLASSE D'ACCÈS:

ACTUATE**8.5.5 RemovePreset**

L'opération RemovePreset supprime un préréglage déjà défini.

DEMANDE:

- **ProfileToken [tt:ReferenceToken]**
Référence à un profil multimédia existant.
- **PresetToken [tt:ReferenceToken]**
Jeton pré-réglé existant à supprimer.

RÉPONSE:

Ceci est un message vide.

DÉFAUTS:

- **env:Sender – ter:InvalidArgVal – ter:NoProfile**
Le jeton de profil (ProfileToken) demandé n'existe pas.
- **env:Sender – ter:InvalidArgVal – ter:NoPTZProfile**
Référence à un jeton de pré-réglage existant.
- **env:Sender – ter:InvalidArgVal – ter:NoToken**
Le jeton de pré-réglage demandé n'existe pas.

CLASSE D'ACCÈS:

ACTUATE

8.6 Opérations de position de départ

8.6.1 Généralités

Le Paragraphe 8.6 décrit les opérations utilisées pour gérer la position de départ d'un nœud PTZ. Ces opérations doivent être mises en œuvre pour des nœuds PTZ prenant en charge les positions de départ. Toutes les opérations exigent un ProfileToken référençant un profil multimédia comprenant une PTZConfiguration.

La position de "départ" peut être définie par l'opération SetHome ou est une position fixe de l'unité PTZ.

8.6.2 GotoHomePosition

Cette opération déplace l'unité PTZ vers sa position de départ. Si le paramètre de vitesse est omis, la vitesse par défaut de la configuration PTZ correspondante doit être utilisée. Le paramètre de vitesse peut être spécifié uniquement si des espaces de vitesse sont disponibles pour le nœud PTZ. La commande est non bloquante et peut être interrompue par d'autres commandes de déplacement.

DEMANDE:

- **ProfileToken [tt:ReferenceToken]**
Référence à un profil multimédia existant.
- **Speed – facultatif [PTZSpeed]**
Vitesse facultative.

RÉPONSE:

Ceci est un message vide.

DÉFAUTS:

- **env:Sender – ter:InvalidArgVal – ter:NoProfile**
Le jeton de profil (ProfileToken) demandé n'existe pas.
- **env:Sender – ter:InvalidArgVal – ter:NoHomePosition**
Aucune position de départ n'a été définie pour ce profil.
- **env:Sender – ter:InvalidArgVal – ter:NoPTZProfile**
Référence à un jeton de préréglage existant.
- **env:Sender – ter:InvalidArgVal – ter:InvalidSpeed**
La vitesse demandée est hors limites.

CLASSE D'ACCÈS:

ACTUATE

8.6.3 SetHomePosition

L'opération SetHome sauvegarde les paramètres de position actuelle en tant que position de départ, de sorte que l'opération GotoHome puisse demander que le dispositif se déplace vers la position de départ.

La commande SetHomePosition doit retourner un échec si la position de "départ" est fixe et ne peut pas être remplacée. Si la commande SetHomePosition aboutit, il doit être possible de rappeler la position de départ à l'aide de la commande GotoHomePosition.

DEMANDE:

- **ProfileToken [tt:ReferenceToken]**
Référence à un profil multimédia existant.

RÉPONSE:

Ceci est un message vide.

DÉFAUTS:

- **env:Sender – ter:InvalidArgVal – ter:NoProfile**
Le jeton de profil (ProfileToken) demandé n'existe pas.
- **env:Sender – ter:InvalidArgVal – ter:NoPTZProfile**
Référence à un jeton de préréglage existant.
- **env:Receiver – ter:Action – ter:CannotOverwriteHome**
La position de départ est fixe et ne peut pas être remplacée.

CLASSE D'ACCÈS:

ACTUATE

8.7 Opérations auxiliaires

8.7.1 Généralités

Le présent article décrit les opérations de gestion des commandes auxiliaires d'un nœud PTZ (une lampe infrarouge (IR), un chauffage ou un dispositif d'essuyage, par exemple).

Ces opérations doivent être mises en œuvre pour des nœuds PTZ indiquant des commandes auxiliaires dans les propriétés de nœud. Toutes les opérations exigent un ProfileToken référant un profil multimédia comprenant une PTZConfiguration.

8.7.2 SendAuxiliaryCommand

Cette opération est utilisée pour appeler une opération auxiliaire sur le dispositif. Les commandes prises en charge peuvent être extraites au moyen des propriétés de nœud PTZ. Il convient que la AuxiliaryCommand corresponde à la commande prise en charge répertoriée dans le nœud PTZ. Aucune autre syntaxe n'est prise en charge. Si le nœud PTZ répertorie la commande tt:IRLamp, le paramètre de la commande auxiliaire doit alors être conforme à la syntaxe spécifiée dans l'IEC 60839-11-31. L'opération SendAuxiliaryCommand doit être mise en œuvre lorsque le nœud PTZ prend en charge les commandes auxiliaires.

DEMANDE:

- **ProfileToken [tt:ReferenceToken]**
Référence à un profil multimédia existant.
- **AuxiliaryData [tt:AuxiliaryData]**
Commande auxiliaire à appliquer.

RÉPONSE:

- **AuxiliaryResponse[tt:AuxiliaryData]**
Informations concernant les réponses auxiliaires.

DÉFAUTS:

- **env:Sender – ter:InvalidArgVal – ter:NoProfile**
Le jeton de profil (ProfileToken) demandé n'existe pas.
- **env:Sender – ter:InvalidArgVal – ter:NoPTZProfile**
Référence à un jeton de préréglage existant.

CLASSE D'ACCÈS:

ACTUATE

8.8 Espaces PTZ prédéfinis

8.8.1 Généralités

Des espaces sont utilisés pour spécifier des mouvements absolus, relatifs et continus. Les mouvements absolus exigent une position absolue, alors que les mouvements relatifs sont spécifiés par une translation de position. Les mouvements continus exigent la spécification d'une vitesse (mouvement relatif au cours du temps). Pour ces trois cas, différents systèmes de coordonnées sont utilisés décrivant le mouvement souhaité. Les espaces génériques ne spécifient pas de façon absolue le modèle PTZ sous-jacent, de sorte qu'il peut être appliqué à tout matériel PTZ.

8.8.2 Espaces de position absolue

8.8.2.1 Généralités

Les espaces de position absolue sont utilisés lorsque le client souhaite déplacer la caméra selon une certaine position. Le mouvement absolu entre la position A actuelle et une position B choisie de manière arbitraire ne doit pas suivre une trajectoire spécifique. En revanche, le dispositif PTZ peut choisir la trajectoire la plus courte afin d'atteindre la destination cible.

8.8.2.2 Espace de position de panoramique/inclinaison générique

L'espace de panoramique/inclinaison générique doit être fourni par chaque nœud PTZ qui prend en charge un panoramique/inclinaison absolu, étant donné qu'il n'est pas lié à un intervalle physique spécifique. Il convient en revanche de définir l'intervalle comme l'intervalle total de l'unité PTZ normalisé à l'intervalle compris entre -1 et 1, produisant la description d'espace suivante:

```
<tt:AbsolutePanTiltPositionSpace>
  <tt:SpaceURI>
    http://www.onvif.org/ver10/tptz/PanTiltSpaces/PositionGenericSpace
  </tt:SpaceURI>
  <tt:Xrange>
    <tt:Min>-1.0</tt:Min>
    <tt:Max>1.0</tt:Max>
  </tt:Xrange>
  <tt:Yrange>
    <tt:Min>-1.0</tt:Min>
    <tt:Max>1.0</tt:Max>
  </tt:Yrange>
</tt:AbsolutePanTiltPositionSpace>
```

8.8.2.3 Espace de position de panoramique/inclinaison sphérique en degrés

La Figure 7 présente une caméra avec une mécanique de panoramique et d'inclinaison et le système de coordonnées sphériques correspondant. Par hypothèse, la description de l'espace considère que le dôme est fixé au plafond. Un mouvement de panoramique est défini comme la rotation du module caméra autour de l'axe de panoramique. Ainsi, l'axe d'inclinaison tourne également dans le même sens dans le plan orthogonal à l'axe de panoramique, de sorte qu'il soit toujours orthogonal à l'axe de la lentille de la caméra. Un mouvement d'inclinaison est défini comme la rotation du module caméra autour de l'axe d'inclinaison. L'axe d'inclinaison permet de modifier la direction d'observation de la caméra, à savoir d'une position descendante vers une position horizontale. Certains dispositifs peuvent prendre en charge une caméra qui peut observer au-dessus de la position horizontale.

Les angles qui décrivent la rotation autour des axes de panoramique et d'inclinaison sont appelés angles de panoramique et d'inclinaison, avec lesquels le panoramique est représenté par la coordonnée X du vecteur de position et l'inclinaison est représentée par la coordonnée Y de ce même vecteur. Les deux angles sont spécifiés en degrés. La position initiale de ce système de coordonnées correspond à une position parallèle entre la direction de la lentille de la caméra et le plafond. Les coordonnées des angles de panoramique et d'inclinaison dans cette direction d'origine sont (0,0).

En partant de la direction d'origine et en augmentant l'angle de panoramique, les objets situés précédemment au centre de l'image se déplacent vers sa gauche. En partant de la direction d'origine et en augmentant l'angle d'inclinaison, les objets situés précédemment au milieu de l'image se déplacent vers sa partie inférieure.

La plage maximale des angles de panoramique et d'inclinaison est comprise entre -180° et $+180^\circ$ ². Le dispositif peut limiter la plage d'inclinaison de manière arbitraire. L'angle d'inclinaison d'une caméra peut modifier sa direction dans l'espace d'un hémisphère tel qu'une caméra dôme généralement limité par des angles compris entre 0° et -90° . Si un dispositif ne peut pas effectuer un balayage panoramique sur toute la plage possible, il peut limiter la plage de panoramique à un intervalle approprié.

² -180° et $+180^\circ$ correspondent à la même position de panoramique dans l'espace et la caméra peut choisir de représenter la position de panoramique en utilisant l'un quelconque de ces angles.

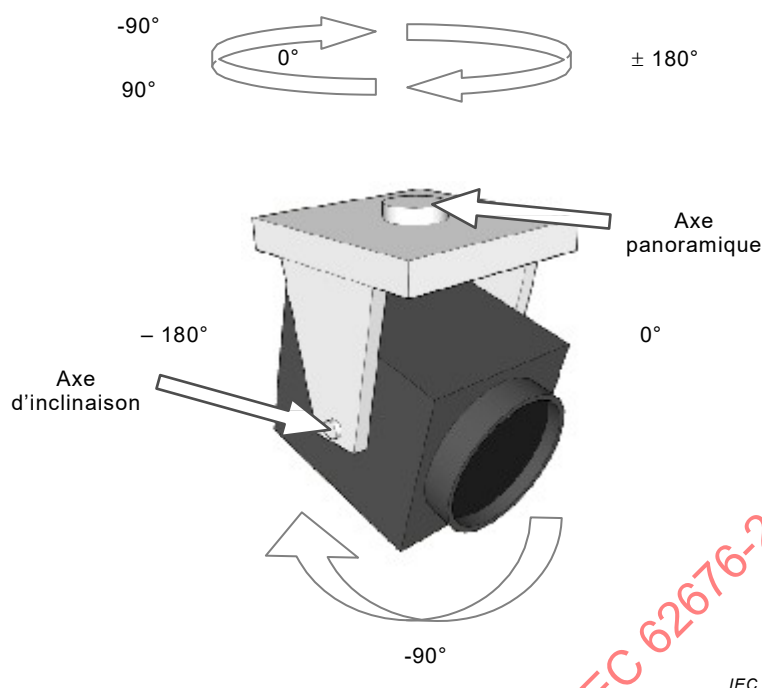


Figure 7 – Espace de position de panoramique/inclinaison sphérique en degrés pour une caméra fixée au plafond

Une caméra qui ne prend pas en charge la bascule électronique (ou qui l'a désactivée) et permet une plage d'inclinaison absolue entre 0° et -180° ne doit pas faire pivoter l'image et modifier les directions lors de la demande d'un mouvement d'inclinaison selon un angle inférieur à -90° .

Si une caméra prend en charge la bascule électronique et un client demande un mouvement d'inclinaison au-delà du nadir, la réaction suivante doit être attendue:

- Il peut y avoir un espace d'hystérèse ($\pm\alpha^\circ$) autour de la position du nadir, de sorte que l'image et le sens de rotation ne se situent pas exactement au niveau du nadir qui peut produire une oscillation gênante.
- Lorsque la caméra fait pivoter l'image et les directions de déplacement (au-delà du nadir $\pm\alpha^\circ$), le système de coordonnées subit les modifications suivantes:
 - Le nouvel angle de panoramique est $P_{\text{nouveau}} = P_{\text{ancien}} + 180^\circ$
 - Le nouvel angle d'inclinaison $T_{\text{nouveau}} = -180^\circ - T_{\text{ancien}}$, $-180 \leq T_{\text{ancien}} \leq -90$
 - En ne tenant pas compte de la zone d'hystérèse, il existe une mise en correspondance individuelle entre les directions d'observation et les coordonnées de panoramique/inclinaison, même si deux positions mécaniques de la caméra correspondent à la même direction d'observation, avec pivotement toutefois de l'image. Par exemple, les deux directions mécaniques $P_1: 35$, $T_1: -20$ et $P_2: -145$, $T_2: -160$, sont représentées par la coordonnée $P: 35$, $T: -20$ étant donné qu'une inclinaison $< -90 - \alpha$ met à jour les coordonnées comme décrit ci-dessus.

L'espace de position de panoramique/inclinaison sphérique est défini comme suit:

```

<tt:AbsolutePanTiltPositionSpace>
  <tt:SpaceURI>
    http://www.onvif.org/ver10/tptz/PanTiltSpaces/SphericalPositionSpaceDegrees
  </tt:SpaceURI>
  <tt:XRange>
    <tt:Min>-INF</tt:Min>
    <tt:Max>INF</tt:Max>
  </tt:XRange>
  <tt:YRange>
    <tt:Min>-180.0</tt:Min>
    <tt:Max>180.0</tt:Max>
  </tt:YRange>
</tt:AbsolutePanTiltPositionSpace>

```

Il convient que le dispositif spécifique référence l'espace de nom proposé, mais les intervalles réels sont spécifiques au dispositif. Les intervalles indiqués dans la définition doivent être interprétés comme les limites d'intervalles maximales et minimales applicables à l'espace.

Les éléments Min et Max de XRange sont définis sur plus et moins l'infini afin d'indiquer que l'intervalle n'est pas limité.

8.8.2.4 Espace de position de zoom générique

L'espace de position de zoom générique doit être fourni par chaque nœud PTZ qui prend en charge un zoom absolu, dans la mesure où il n'est pas associé à un intervalle physique spécifique. Il convient en revanche de définir l'intervalle comme l'intervalle total du zoom normalisé à l'intervalle compris entre 0 (large) et 1 (zoomé). Aucune hypothèse n'est formulée concernant la façon dont l'intervalle de zoom générique est mis en correspondance avec le grossissement, le champ de vision ou une autre dimension de zoom physique. Cela donne lieu à la description d'espace suivante:

```

<tt:AbsoluteZoomPositionSpace>
  <tt:SpaceURI>
    http://www.onvif.org/ver10/tptz/ZoomSpaces/PositionGenericSpace
  </tt:SpaceURI>
  <tt:Xrange>
    <tt:Min>0.0</tt:Min>
    <tt:Max>1.0</tt:Max>
  </tt:Xrange>
</tt:AbsoluteZoomPositionSpace>

```

8.8.2.5 Espace de position de zoom en millimètres

L'espace de position de zoom en millimètres exprime la longueur focale absolue d'une caméra télescopique. La longueur focale maximale et minimale est spécifique à la caméra. L'espace de position de zoom en millimètres est défini comme suit:

```

<tt:AbsoluteZoomPositionSpace>
  <tt:SpaceURI>
    http://www.onvif.org/ver10/tptz/ZoomSpaces/PositionSpaceMillimeter
  </tt:SpaceURI>
  <tt:XRange>
    <tt:Min>0.0</tt:Min>
    <tt:Max>INF</tt:Max>
  </tt:XRange>
</tt:AbsoluteZoomPositionSpace>

```

Il convient que le dispositif spécifique référence l'espace de nom proposé, mais l'intervalle réel est spécifique au dispositif. L'intervalle indiqué dans la définition doit être interprété comme constituant les limites d'intervalles maximales et minimales applicables à l'espace.

Le grossissement de zoom actuel (M) d'une caméra peut être calculé à partir de la longueur focale comme suit:

$$M = \frac{f_c}{f_0},$$

où

f_0 est la longueur focale minimale, et

f_c est la longueur focale actuelle.

L'angle de vision actuel peut être calculé à l'aide de la formule suivante:

$$\alpha = 2 \arctan \frac{d}{2f_c},$$

où

d représente la taille du capteur dans la direction concernée, et

f_c est la longueur focale actuelle.

8.8.2.6 Espace de position de zoom numérique normalisé

8.8.2.6.1 Généralités

L'espace de position de zoom numérique normalisé représente la mise en correspondance entre la largeur de la fenêtre utilisée pour un flux vidéo et la largeur de la source vidéo (capteur). Cet espace exige un profil multimédia qui inclut une VideoEncoderConfiguration et une VideoSourceConfiguration.

Le facteur de zoom numérique normalisé est calculé comme suit: $Z = 1 / \left(\frac{W_w}{S_w} \right)$.

Le niveau de zoom maximal qui peut être représenté par cet espace est limité par la plus petite taille de la fenêtre admise par la caméra.

L'espace de zoom de position numérique normalisé est défini comme suit:

```
<tt:AbsoluteZoomPositionSpace>
  <tt:SpaceURI>
    http://www.onvif.org/ver10/tptz/ZoomSpaces/NormalizedDigitalPositionSpace
  </tt:SpaceURI>
  <tt:XRange>
    <tt:Min>1.0</tt:Min>
    <tt:Max>INF</tt:Max>
  </tt:XRange>
</tt:AbsoluteZoomPositionSpace>
```

Il convient que le dispositif spécifique référence l'espace de nom proposé, mais l'intervalle réel est spécifique au dispositif. L'intervalle indiqué dans la définition doit être interprété comme constituant la limite d'intervalle maximale applicable à l'espace.

Il est important de constater que le fonctionnement du zoom dans cet espace n'altère pas la résolution actuelle du flux vidéo configuré dans la VideoEncoderConfiguration du profil multimédia. Il modifie en revanche la largeur et la hauteur de la fenêtre configurée dans la VideoSourceConfiguration.

8.8.2.6.2 Exemple: Niveau de zoom pour une zone recadrée

Par exemple, une unité PTZ numérique avec une fenêtre VGA recadrée d'origine (640 x 480) montée sur un capteur 3 MP (2048 x 1536) a un niveau de zoom $Z = 1 / \left(\frac{640}{2048} \right) = 3,2$.

8.8.3 Espaces de translation relative

8.8.3.1 Généralités

Un espace de translation de panoramique/inclinaison relative déplace l'unité PTZ d'une certaine translation dans une certaine direction sans connaître la position actuelle de panoramique/inclinaison de la caméra.

Une translation de panoramique/inclinaison relative peut être déduite d'un espace de position de panoramique/inclinaison (numérique/sphérique) absolu correspondant en prenant la différence de deux positions de panoramique/inclinaison absolues. Cependant, il existe également des translations de panoramique/inclinaison relatives pour lesquelles aucun espace de panoramique/inclinaison absolu correspondant ne peut être défini.

8.8.3.2 Espace de translation de panoramique/inclinaison générique

L'espace de translation de panoramique/inclinaison générique doit être fourni par chaque nœud PTZ qui prend en charge le panoramique/inclinaison relatif, dans la mesure où il n'est pas associé à un intervalle physique spécifique. Il convient en revanche de définir l'intervalle comme l'intervalle de translation positive et négative total de l'unité PTZ normalisé à l'intervalle compris entre -1 et 1, dans lequel une translation positive désigne une rotation horaire ou un mouvement vers la droite/le haut conduisant à la description d'espace suivante:

```
<tt:RelativePanTiltTranslationSpace>
  <tt:SpaceURI>
    http://www.onvif.org/ver10/tptz/PanTiltSpaces/TranslationGenericSpace
  </tt:SpaceURI>
  <tt:Xrange>
    <tt:Min>-1.0</tt:Min>
    <tt:Max>1.0</tt:Max>
  </tt:Xrange>
  <tt:Yrange>
    <tt:Min>-1.0</tt:Min>
    <tt:Max>1.0</tt:Max>
  </tt:Yrange>
</tt:RelativePanTiltTranslationSpace>
```

8.8.3.3 Espace de translation de zoom générique

L'espace de translation de zoom générique doit être fourni par chaque nœud PTZ qui prend en charge le zoom relatif, dans la mesure où il n'est pas associé à un intervalle physique spécifique. Il convient en revanche de définir l'intervalle absolu correspondant comme l'intervalle de translation positive et négative total de la fonction Zoom normalisé à l'intervalle compris entre -1 et 1, dans lequel une translation positive correspond à un mouvement dans la direction d'un zoom positif. La translation est signée de manière à indiquer la direction (négative pour large, positive pour zoom positif). Aucune hypothèse n'est formulée concernant la façon dont l'intervalle de zoom générique est mis en correspondance avec le grossissement, le champ de vision ou une autre dimension de zoom physique. Cela donne lieu à la description d'espace suivante:

```
<tt:RelativeZoomTranslationSpace>
  <tt:SpaceURI>
    http://www.onvif.org/ver10/tptz/ZoomSpaces/TranslationGenericSpace
  </tt:SpaceURI>
  <tt:Xrange>
    <tt:Min>-1.0</tt:Min>
```



```

<tt:Max>1.0</tt:Max>
</tt:Xrange>
</tt:RelativeZoomTranslationSpace>

```

8.8.4 Espaces de vitesse continue

8.8.4.1 Généralités

Les espaces de vitesse continue sont utilisés pour déplacer en continu l'unité PTZ dans une certaine direction.

Lorsqu'une caméra prend en charge la bascule électronique, et l'a activée, il convient de prévoir la réaction suivante des espaces de panoramique/inclinaison définis en 8.8.4.3:

Lorsque la translation d'inclinaison dépasse la position du nadir (comprenant l'espace pour une hystérèse $\pm\alpha^\circ$), il convient que la caméra fasse pivoter l'image à la position du nadir $\pm\alpha^\circ$ tel que décrit en 8.8.2.3 (uniquement pour les caméras qui prennent en charge le positionnement de panoramique/inclinaison absolu). En cas de rotation, la caméra continue les mouvements actuels selon les directions indiquées lors de l'émission de la commande. Si la commande est interrompue par une nouvelle demande (après la bascule), cette demande est traitée selon la nouvelle direction (avec bascule) et les nouvelles coordonnées. Une caméra qui ne prend pas en charge la bascule électronique ou qui l'a désactivée ne fait pivoter ni l'image ni les directions lors d'un mouvement d'inclinaison au-delà de la position du nadir.

8.8.4.2 Espace de vitesse de panoramique/inclinaison générique

L'espace de vitesse de panoramique/inclinaison générique doit être fourni par chaque nœud PTZ, dans la mesure où il n'est pas associé à un intervalle physique spécifique. Il convient en revanche de définir l'intervalle comme un intervalle de la vitesse de l'unité PTZ normalisé à l'intervalle compris entre -1 et 1, dans lequel une vitesse positive correspond à une rotation horaire ou un mouvement vers la droite/le haut. Une vitesse signée peut être indépendamment spécifiée pour la composante de panoramique et d'inclinaison conduisant à la description d'espace suivante:

```

<tt:ContinuousPanTiltVelocitySpace>
  <tt:SpaceURI>
    http://www.onvif.org/ver10/tptz/PanTiltSpaces/VelocityGenericSpace
  </tt:SpaceURI>
  <tt:Xrange>
    <tt:Min>-1.0</tt:Min>
    <tt:Max>1.0</tt:Max>
  </tt:Xrange>
  <tt:Yrange>
    <tt:Min>-1.0</tt:Min>
    <tt:Max>1.0</tt:Max>
  </tt:Yrange>
</tt:ContinuousPanTiltVelocitySpace>

```

8.8.4.3 Espace de vitesse de panoramique/inclinaison en degrés par seconde

L'espace de vitesse de panoramique/inclinaison en degrés par seconde décrit une rotation continue autour de l'axe de panoramique et d'inclinaison (spécifié en 8.8.2.3) avec une vitesse de rotation constante. De fait, la vitesse décrit la modification de l'angle de panoramique et d'inclinaison signé, en degrés par seconde. La vitesse de panoramique est exprimée par la coordonnée X du vecteur de direction et la vitesse d'inclinaison par la coordonnée Y , et l'espace est défini comme suit:

```

<tt:ContinuousPanTiltVelocitySpace>
  <tt:SpaceURI>
    http://www.onvif.org/ver10/tptz/PanTiltSpaces/VelocitySpaceDegrees
  </tt:SpaceURI>
  <tt:XRange>
    <tt:Min>-INF</tt:Min>
    <tt:Max>INF</tt:Max>
  </tt:XRange>
  <tt:YRange>
    <tt:Min>-INF</tt:Min>
    <tt:Max>INF</tt:Max>
  </tt:YRange>
</tt:ContinuousPanTiltVelocitySpace>

```

Il convient que le dispositif spécifique référence l'espace de nom proposé, mais les intervalles réels sont spécifiques au dispositif. Les intervalles indiqués dans la définition doivent être interprétés comme les limites d'intervalles maximales et minimales applicables à l'espace.

8.8.4.4 Espace de vitesse de zoom générique

L'espace de vitesse de zoom générique spécifie une vitesse de facteur de zoom sans connaître le modèle physique sous-jacent. Il convient que l'intervalle soit normalisé entre -1 et 1, une vitesse positive correspondant à une direction de zoom positif. Une description d'espace de vitesse de zoom générique se présente comme suit:

```

<tt:ContinuousZoomVelocitySpace>
  <tt:SpaceURI>
    http://www.onvif.org/ver10/tptz/ZoomSpaces/VelocityGenericSpace
  </tt:SpaceURI>
  <tt:Xrange>
    <tt:Min>-1.0</tt:Min>
    <tt:Max>1.0</tt:Max>
  </tt:Xrange>
</tt:ContinuousZoomVelocitySpace>

```

8.8.5 Espaces de vitesse

8.8.5.1 Généralités

Les espaces de vitesse spécifient la vitesse d'un mouvement de panoramique/inclinaison et de zoom lors d'un mouvement vers une position absolue ou une translation relative. Contrairement aux espaces de vitesse, les espaces de vitesse ne contiennent pas d'information de direction. La vitesse d'un mouvement de panoramique/inclinaison combiné est représentée par une seule valeur scalaire non négative.

Si un espace de translation relative et un espace de vitesse continue sont déjà définis, l'espace de vitesse correspondant est déduit comme suit: la demande d'un mouvement continu avec une vitesse V pendant T secondes, équivaut (jusqu'à des inexactitudes d'accélération et de position) à une demande de mouvement relatif avec une position relative R et une vitesse S , où R est égale à V fois T et S est égale à la longueur du vecteur V . Par conséquent, les valeurs de vitesse sont toujours positives.

8.8.5.2 Espace de vitesse de panoramique/inclinaison générique

L'espace de vitesse de panoramique/inclinaison générique doit être fourni par chaque nœud PTZ qui prend en charge la vitesse configurable du mouvement de panoramique/inclinaison, dans la mesure où il n'est pas associé à un intervalle physique spécifique. Il convient en revanche de définir l'intervalle comme l'intervalle total de l'intervalle de vitesse normalisé à l'intervalle compris entre 0 (arrêt) et 1 (vitesse maximale). Cela donne lieu à la description d'espace suivante:

```

<tt:PanTiltSpeedSpace>
  <tt:SpaceURI>
    http://www.onvif.org/ver10/tptz/PanTiltSpaces/GenericSpeedSpace
  </tt:SpaceURI>
  <tt:XRange>
    <tt:Min>0.0</tt:Min>
    <tt:Max>1.0</tt:Max>
  </tt:XRange>
</tt:PanTiltSpeedSpace>

```

8.8.5.3 Espace de vitesse de panoramique/inclinaison en degrés par seconde

L'espace de vitesse de panoramique/inclinaison en degrés par seconde est déduit de l'espace de position de panoramique/inclinaison absolu en degrés pour des mouvements absolus et relatifs. L'espace de vitesse de panoramique/inclinaison en degrés par seconde est défini comme suit:

```

<tt:PanTiltSpeedSpace>
  <tt:SpaceURI>
    http://www.onvif.org/ver10/tptz/PanTiltSpaces/SpeedSpaceDegrees
  </tt:SpaceURI>
  <tt:XRange>
    <tt:Min>0.0</tt:Min>
    <tt:Max>INF</tt:Max>
  </tt:XRange>
</tt:PanTiltSpeedSpace>

```

Il convient que le dispositif spécifique référence l'espace de nom proposé, mais l'intervalle réel est spécifique au dispositif. L'intervalle indiqué dans la définition doit être interprété comme constituant les limites d'intervalles maximales et minimales applicables à l'espace.

8.8.5.4 Espace de vitesse de zoom générique

L'espace de vitesse de zoom générique doit être fourni par chaque nœud PTZ qui prend en charge la vitesse configurable du zoom, dans la mesure où il n'est pas associé à un intervalle physique spécifique. Il convient en revanche de définir l'intervalle comme l'intervalle total de l'intervalle de vitesse normalisé à l'intervalle compris entre 0 (arrêt) et 1 (vitesse maximale). Cela donne lieu à la description d'espace suivante:

```

<tt:ZoomSpeedSpace>
  <tt:SpaceURI>
    http://www.onvif.org/ver10/tptz/ZoomSpaces/ZoomGenericSpeedSpace
  </tt:SpaceURI>
  <tt:XRange>
    <tt:Min>0.0</tt:Min>
    <tt:Max>1.0</tt:Max>
  </tt:XRange>
</tt:ZoomSpeedSpace>

```

8.8.5.5 Espace de vitesse de zoom en millimètres par seconde

L'espace de vitesse de zoom en millimètres par seconde spécifie la modification absolue par seconde de la longueur focale. Si le dispositif prend en charge l'espace de vitesse de zoom, l'espace de vitesse de zoom en millimètres par seconde est défini comme suit:

```

<tt:ZoomSpeedSpace>
  <tt:SpaceURI>
    http://www.onvif.org/ver10/tptz/ZoomSpaces/SpeedSpaceMillimeter
  </tt:SpaceURI>
  <tt:XRange>
    <tt:Min>0.0</tt:Min>
    <tt:Max>INF</tt:Max>
  </tt:XRange>
</tt:ZoomSpeedSpace>

```

Il convient que le dispositif spécifique référence l'espace de nom proposé, mais l'intervalle réel est spécifique au dispositif. L'intervalle indiqué dans la définition doit être interprété comme la limite d'intervalle maximale et minimale applicable à l'espace.

8.9 Opérations de tours de préréglage

8.9.1 Généralités

Une opération de tour de préréglage est une fonctionnalité propre aux dispositifs comportant la fonctionnalité PTZ, qui permet le déplacement de l'unité PTZ sur des préréglages spécifiés de manière séquentielle à un certain intervalle.

Le Paragraphe 8.9 décrit les opérations de gestion des tours de préréglage. Ces opérations doivent être mises en oeuvre lorsqu'un nœud PTZ du service de même nature indique la prise en charge de la fonctionnalité de tour de préréglage avec `MaximumNumberOfPresetTours` > valeur de fonctionnalité nulle. Toutes les opérations exigent un `ProfileToken` qui fait référence un profil multimédia comprenant une `PTZConfiguration`. Toutes les opérations de tours de préréglage doivent être toujours permanentes.

La fonctionnalité `PresetTours` peut s'exécuter dans les sens avant et arrière, et, par hypothèse, tout sens de préréglage omis, est considéré comme un sens avant. Cependant, il est possible de spécifier une exécution aléatoire en définissant l'attribut `RandomPresetOrder`. Si la valeur de `RandomPresetOrder` est "true" et si l'élément `Direction` est également présent, cet élément est ignoré et les tours de préréglage font l'objet d'un rappel aléatoire.

8.9.2 GetPresetTours

Un dispositif qui prend en charge la fonctionnalité de tour de préréglage doit retourner tous les tours de préréglage disponibles au moyen de la commande `GetPresetTours`.

DEMANDE:

- **ProfileToken [tt:ReferenceToken]**
Référence à un profil multimédia existant.

RÉPONSE:

- **PresetTour – facultatif, non limité [tt:PTZPresetTour]**
Liste des tours de préréglage définis pour le profil multimédia demandé.

DÉFAUTS:

- **env:Sender – ter:InvalidArgVal – ter:NoProfile**
Le jeton de profil demandé n'existe pas.
- **env:Sender – ter:InvalidArgVal – ter:NoPTZProfile**
Le jeton de profil demandé ne référence pas une configuration PTZ.

CLASSE D'ACCÈS:

READ_MEDIA

8.9.3 GetPresetTour

Un dispositif qui prend en charge la fonctionnalité PresetTour doit retourner le tour de préséglage demandé au moyen de la commande GetPresetTour.

DEMANDE:

- **ProfileToken [tt:ReferenceToken]**
Référence à un profil multimédia existant.
- **PresetTourToken [tt:ReferenceToken]**
Référence à un tour de préséglage existant.

RÉPONSE:

- **PresetTour [tt:PTZPresetTour]**
Tour de préséglage demandé.

DÉFAUTS:

- **env:Sender – ter:InvalidArgVal – ter:NoProfile**
Le jeton de profil (ProfileToken) demandé n'existe pas.
- **env:Sender – ter:InvalidArgVal – ter:NoPTZProfile**
Le jeton de profil demandé ne référence pas une configuration PTZ.
- **env:Sender – ter:InvalidArgVal – ter:NoToken**
Le jeton de tour préséglage demandé n'existe pas.

CLASSE D'ACCÈS:

READ_MEDIA

8.9.4 GetPresetTourOptions

Un dispositif qui prend en charge la fonctionnalité PresetTour doit proposer des options de mode de configuration des tours de préséglage au moyen de la commande GetPresetTourOptions.

DEMANDE:

- **ProfileToken [tt:ReferenceToken]**
Référence à un profil multimédia existant.
- **PresetTourToken – facultatif [tt:ReferenceToken]**
Référence facultative à un tour de préséglage existant.

RÉPONSE:

- **PresetTour [tt:PTZPresetTourOptions]**
Options de tours de préséglage demandées.

DÉFAUTS:

- **env:Sender – ter:InvalidArgVal – ter:NoProfile**
Le jeton de profil (ProfileToken) demandé n'existe pas.
- **env:Sender – ter:InvalidArgVal – ter:NoPTZProfile**
Le jeton de profil demandé ne référence pas une configuration PTZ.

CLASSE D'ACCÈS:

READ_MEDIA

8.9.5 CreatePresetTour

Un dispositif qui prend en charge la fonctionnalité de tour de préréglage doit permettre de créer un nouveau tour de préréglage au moyen de la commande CreatePresetTour.

DEMANDE:

- **ProfileToken [tt:ReferenceToken]**
Référence à un profil multimédia existant.

RÉPONSE:

- **PresetTourToken [tt:ReferenceToken]**
Référence à un tour de préréglage existant.

DÉFAUTS:

- **env:Sender – ter:InvalidArgVal – ter:NoProfile**
Le jeton de profil demandé n'existe pas.
- **env:Sender – ter:InvalidArgVal – ter:NoPTZProfile**
Le jeton de profil demandé ne référence pas une configuration PTZ.
- **env:Sender – ter:InvalidArgVal – ter:TooManyPresetTours**
Espace insuffisant dans le dispositif pour créer le nouveau tour de préréglage dans le profil

CLASSE D'ACCÈS:

ACTUATE

8.9.6 ModifyPresetTour

Un dispositif qui prend en charge la fonctionnalité de tour de préréglage doit permettre de modifier un tour de préréglage au moyen de la commande ModifyPresetTour.

DEMANDE:

- **ProfileToken [tt:ReferenceToken]**
Référence à un profil multimédia existant.
- **PresetTour [tt:PTZPresetTour]**
Paramètres de tours de préréglage à définir.

RÉPONSE:

Ceci est un message vide.

DÉFAUTS:

- **env:Sender – ter:InvalidArgVal – ter:NoProfile**
Le jeton de profil demandé n'existe pas.
- **env:Sender – ter:InvalidArgVal – ter:NoPTZProfile**
Le jeton de profil demandé ne référence pas une configuration PTZ.
- **env:Sender – ter:InvalidArgVal – ter:InvalidPresetTour**
Le PresetTour demandé comprend un ou des paramètres non valides.
- **env:Sender – ter:InvalidArgVal – ter:TooManyPresets**
Un PresetTour comprend un trop grand nombre de TourSpots.
- **env:Sender – ter:InvalidArgVal – ter:NoToken**
Le jeton de tour préréglage demandé n'existe pas.
- **env:Sender – ter:InvalidArgVal – ter:SpaceNotSupported**
Un espace est référencé, mais n'est pas pris en charge par le nœud PTZ.

CLASSE D'ACCÈS:

ACTUATE**8.9.7 OperatePresetTour**

Un dispositif qui prend en charge la fonctionnalité PresetTour doit permettre le démarrage, l'interruption ou la pause d'un tour de préréglage au moyen de la commande OperatePresetTour.

Un tour de préréglage peut être exécuté avec le paramètre PresetTourOperation de la commande OperatePresetTour.

- Démarrage: indique le démarrage du tour de préréglage ou le redémarrage du tour de préréglage mis en pause.
- Interruption: indique l'interruption du tour de préréglage.
- Pause: indique la pause du tour de préréglage.

À réception d'une autre commande OperatePresetTour de démarrage du tour de préréglage déjà initié, le tour de préréglage doit faire l'objet d'un nouveau démarrage avec le paramètre nouvellement demandé.

DEMANDE:

- **ProfileToken [tt:ReferenceToken]**
Référence à un profil multimédia existant.
- **PresetTourToken [tt:PTZPresetTourToken]**
Référence à un tour de préréglage existant.
- **Operation [tt:PTZPresetTourOperation]**
Information de fonctionnement.

RÉPONSE:

- **Ceci est un message vide.**

DÉFAUTS:

- **env:Sender – ter:InvalidArgVal – ter:NoProfile**
Le jeton de profil demandé n'existe pas.
- **env:Sender – ter:InvalidArgVal – ter:NoPTZProfile**
Le jeton de profil demandé ne référence pas une configuration PTZ.
- **env:Sender – ter:InvalidArgVal – ter:NoToken**
Le jeton de tour pré réglage demandé n'existe pas.
- **env:Sender – ter:InvalidArgVal – ter:InvalidPresetTour**
Le PresetTour demandé comprend un ou des paramètres non valides.
- **env:Receiver – ter:InvalidArgVal – ter:ActivationFailed**
Le tour de pré réglage demandé ne peut pas être activé lorsque l'unité PTZ se déplace ou un autre tour de pré réglage est à présent activé.

CLASSE D'ACCÈS:

ACTUATE

8.9.8 RemovePresetTour

Un dispositif qui prend en charge la fonctionnalité PresetTour doit prendre en charge la suppression des tours de pré réglage au moyen de la commande RemovePresetTour.

DEMANDE:

- **ProfileToken [tt:ReferenceToken]**
Référence à un profil multimédia existant.
- **PresetTourToken [tt:PTZPresetTourToken]**
Référence à un tour de pré réglage existant.

RÉPONSE:

Ceci est un message vide.

DÉFAUTS:

- **env:Sender – ter:InvalidArgVal – ter:NoProfile**
Le jeton de profil (ProfileToken) demandé n'existe pas.
- **env:Sender – ter:InvalidArgVal – ter:NoPTZProfile**
Le jeton de profil demandé ne référence pas une configuration PTZ.
- **env:Sender – ter:InvalidArgVal – ter:NoToken**
Le jeton de tour pré réglage demandé n'existe pas.

CLASSE D'ACCÈS:

ACTUATE

8.9.9 Paramètres de tour de pré réglage

Un tour de pré réglage est constitué des paramètres suivants:

PresetTour: Configuration d'un tour de pré réglage.

- Name: nom du tour de pré réglage [Facultatif]

- Status: indique le statut actuel du tour de pré-réglage. Ce paramètre de l'argument de la commande ModifyPresetTour doit être ignoré.
 - Idle: le tour de pré-réglage n'est pas activé.
 - Touring: le tour de pré-réglage est activé et s'exécute dans le cadre d'un cycle.
 - Paused: le tour de pré-réglage est en position de pause au milieu du tour exécuté par le client. Ce statut exige une opération de démarrage de la part du client afin de procéder à un redémarrage.
- AutoStart: drapeau permettant l'activation permanente du tour de pré-réglage. Lorsque la valeur de ce drapeau de l'un des tours de pré-réglage est True, la valeur des drapeaux de tous les autres tours de pré-réglage faisant référence au même PTZNode doit être maintenue en tant que valeur False. Le démarrage et l'interruption du tour de pré-réglage par OperatePresetTour doivent modifier la valeur de ce drapeau et lui attribuer la valeur False. La pause et la reprise du tour de pré-réglage par OperatePresetTour ne doivent pas modifier ce drapeau.
- TourSpot: pré-réglages avec configuration pour cycle de tours de pré-réglage [Facultatif]
 - PresetDetail: configuration détaillée du pré-réglage pour le repère de tour. Une des trois options ci-dessous doit être choisie.
 - i) PresetToken: spécifie un PresetToken existant comme repère de tour.
 - ii) Home: spécifie la position de départ comme repère de tour. Seule la valeur "True" de ce paramètre booléen doit être valide. "False" doit être traité comme une erreur en raison d'un argument non valide.
 - iii) PTZPosition: spécifie le vecteur du noeud PTZ comme repère de tour
 - Speed: vitesse PTZ de déplacement vers ce repère de tour. [Facultatif]
 - StayTime: durée de séjour sur le repère. [Facultatif]
- PresetTourStartingCondition: Configuration du tour de pré-réglage de démarrage.
 - RecurringTime: paramètre de spécification du nombre de récurrences du tour de pré-réglage. [Facultatif]
 - RecurringDuration: paramètre de spécification de la durée de récurrence du tour de pré-réglage [Facultatif]
 - i) Si les deux conditions RecurringTime et RecurringDuration font l'objet d'une configuration, le tour de pré-réglage doit s'achever avec l'une des conditions satisfaites précédemment. Si les deux conditions sont omises, la récurrence de PresetTour doit se produire sans limites.
 - PresetTourDirection: un dispositif fait référence à ce drapeau pour choisir le sens du tour de pré-réglage. Il s'agit d'un paramètre qui peut être omis, et le sens Forward (Avant) doit être choisi en cas d'omission. [Facultatif]
 - i) Forward: tour de pré-réglage avant.
 - ii) Backward: tour de pré-réglage arrière.
- Token: identifiant unique du tour de pré-réglage.

8.10 Configuration de direction de contrôle de panoramique/inclinaison

La configuration de direction de contrôle de panoramique/inclinaison est un ensemble de paramètres de la configuration PTZ utilisé pour configurer les directions de mouvement de panoramique et d'inclinaison des deux manières suivantes:

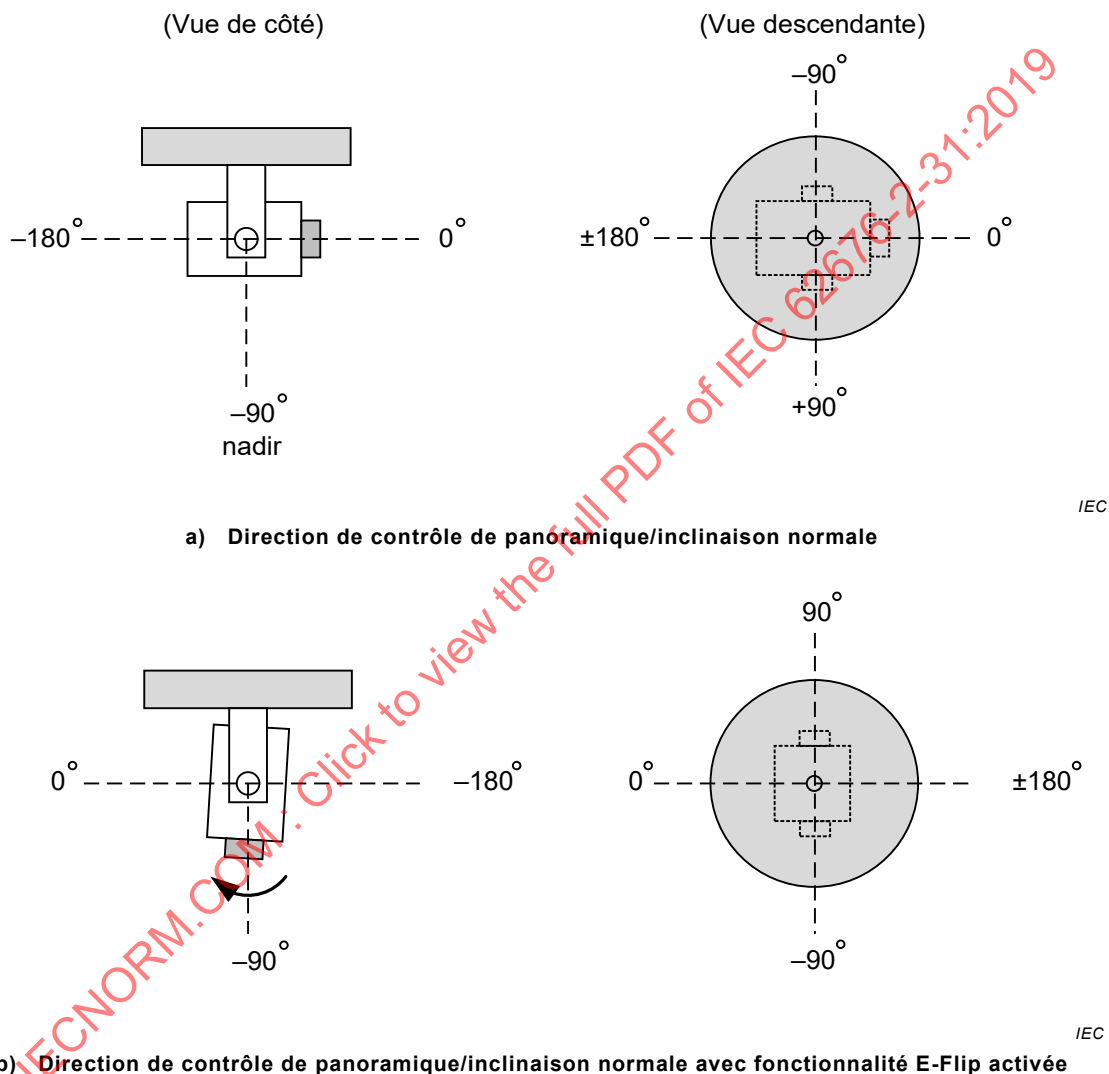
1) E-flip (bascule électronique)

La bascule électronique consiste à basculer les directions de contrôle PT lorsque la lentille dépasse la position du nadir ou le point zénithal lors d'un mouvement d'inclinaison. Cette fonctionnalité est utile lors du contrôle des dômes par une commande à manette, avec laquelle un client peut suivre la trajectoire d'un objet qui dépasse la position du nadir et ne doit pas se préoccuper des contrôles inversés.

2) Reverse (Inverse)

Le paramètre Reverse s'applique à la commutation de la direction de contrôle de panoramique/inclinaison sur la position positive et la position négative. Cette fonctionnalité peut être utilisée pour configurer une direction de contrôle de panoramique/inclinaison appropriée basée sur les sens de montage du dispositif, par exemple, au plafond dans le sens descendant ou à la surface dans le sens ascendant.

La Figure 8 présente un exemple de mode de modification des directions de contrôle panoramique/inclinaison par les fonctionnalités E-Flip et Reverse.



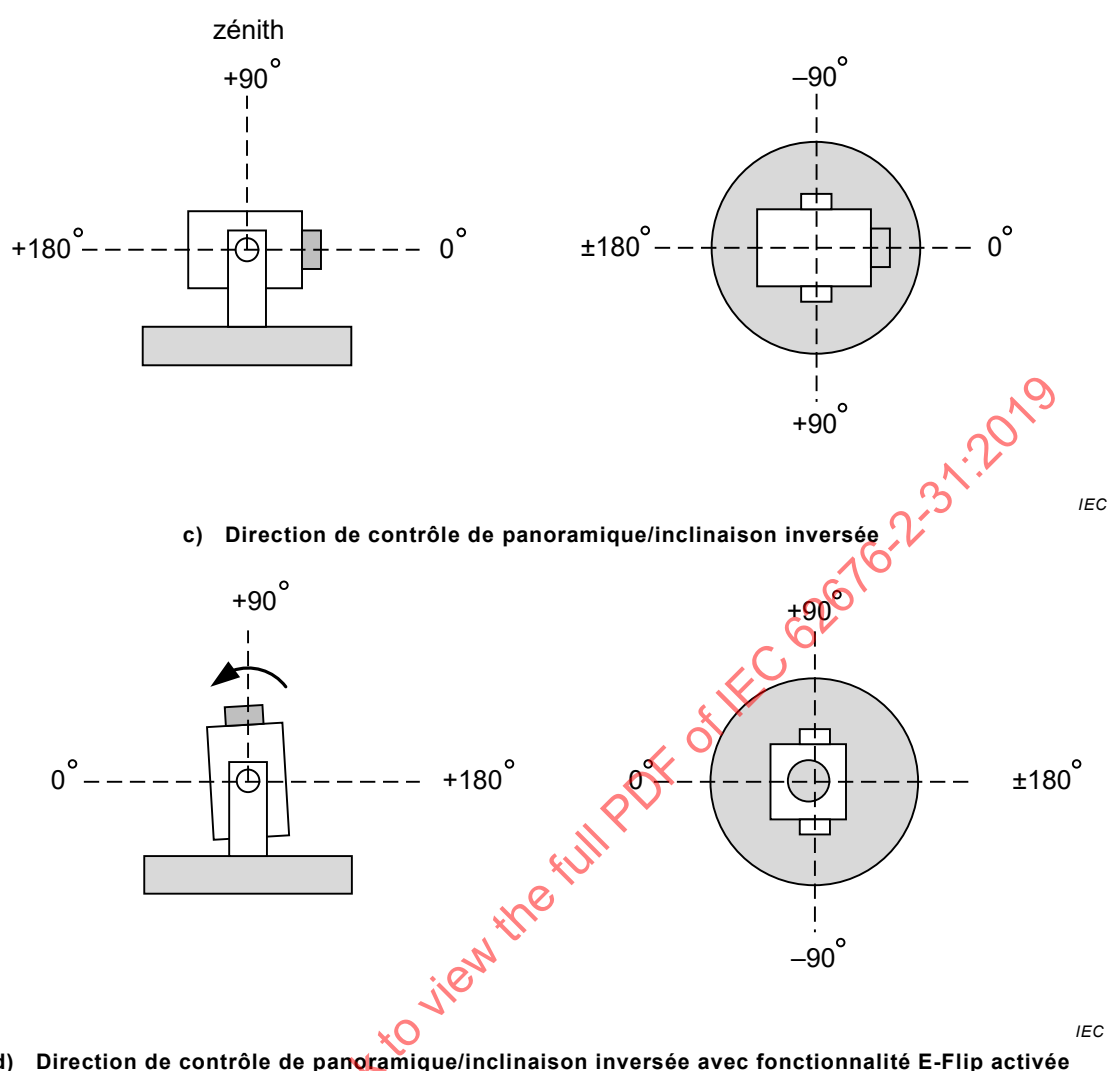


Figure 8 – Exemple de modifications de la direction de contrôle de panoramique/inclinaison par les fonctionnalités E-Flip et Reverse

En cas de modification de la direction de contrôle de panoramique/inclinaison, il est recommandé d'adapter l'orientation de l'image vidéo par le paramètre Rotation de la VideoSourceConfiguration afin d'aligner la fonctionnalité panoramique/inclinaison et la vidéo.

8.11 Fonctionnalités

Les fonctionnalités reflètent les fonctions facultatives et la fonctionnalité d'un service. Les informations sont statiques et ne varient pas pendant le fonctionnement du dispositif. Les fonctionnalités suivantes sont disponibles:

EFlip:	Indique si la fonctionnalité E-flip est prise en charge ou non.
Reverse:	Indique si l'inversement de la direction de contrôle de panoramique/inclinaison est pris en charge ou non.
GetCompatibleConfigurations:	Indique la prise en charge de la commande GetCompatibleConfigurations.
MoveStatus:	Indique que le PTZStatus comprend l'information MoveStatus.
StatusPosition:	Indique que le PTZStatus comprend l'information de position (Position).

DEMANDE:

Ceci est un message vide.

RÉPONSE:

- **Capabilities [tptz:Capabilities]**

Liste des fonctionnalités définies ci-dessus.

DÉFAUTS:

Aucun

CLASSE D'ACCÈS:

PRE_AUTH

8.12 Événements

8.12.1 Généralités

Un dispositif qui prend en charge le service PTZ expédie les événements répertoriés dans le Paragraphe 8.12 par l'intermédiaire du service d'événement.

8.12.2 PréréglagesPTZ

Un dispositif qui prend en charge le service PTZ spécifie le traitement des préréglages PTZ. Étant donné que les opérations de déplacement sont non bloquantes, un client n'est pas informé lorsque le préréglage PTZ a été atteint. Par conséquent, il convient qu'un dispositif qui prend en charge le service PTZ prenne également en charge les événements suivants destinés à informer les abonnés du statut des mouvements prédéfinis.

Topic:

```
tns1:PTZController/PTZPresets/Invoked
tns1:PTZController/PTZPresets/Reached
tns1:PTZController/PTZPresets/Aborted
tns1:PTZController/PTZPresets/Left
```

La séquence d'événements typique prévoit qu'un client exige en premier lieu un certain préréglage. Lorsque le dispositif accepte cette demande, il envoie un événement invoqué. L'événement invoqué doit suivre soit un événement atteint soit un événement avorté. Le premier est utilisé lorsque l'unité PTZ a pu atteindre la position de préréglage invoquée, le second dans tous les autres cas. Un événement atteint doit suivre un événement quitté, dès que l'unité PTZ quitte la position de préréglage.

La structure du message de ces événements est donnée par la description de message suivante.

```
<tt:MessageDescription>
  <tt:Source>
    <tt:SimpleItemDescription Name="PTZConfigurationToken"
      Type="tt:ReferenceToken"/>
  </tt:Source>
  <tt:Data>
    <tt:SimpleItemDescription Name="PresetToken" Type="tt:ReferenceToken"/>
    <tt:SimpleItemDescription Name="PresetName" Type="tt:Name"/>
  </tt:Data>
</tt:MessageDescription>
```

8.12.3 PresetTours

Lorsqu'un nœud PTZ interne à un dispositif qui prend en charge le service PTZ indique la prise en charge de la fonctionnalité Preset Tour avec `MaximumNumberOfPresetTours > valeur de fonctionnalité nulle`, le dispositif doit prendre en charge la rubrique suivante afin d'informer les abonnés de l'existence des événements de tours de préréglage. À chaque modification de la fonctionnalité Preset Tour, le service PTZ doit expédier cet événement.

Topic: `tnsl:PTZController/PTZPresetTours/Configuration`

```
<tt:MessageDescription isProperty=True>
  <tt:Source>
    <tt:SimpleItemDescription Name="ProfileToken" Type="tt:ReferenceToken"/>
    <tt:SimpleItemDescription Name="PresetTourToken" Type="tt:ReferenceToken"/>
  </tt:Source>
  <tt:Data>
    <tt:SimpleItemDescription Name="StateChange" Type="xs:boolean"/>
    <tt:ElementItemDescription Name="PresetTour" Type="tt:PTZPresetTour"/>
  </tt:Data>
</tt:MessageDescription>
```

Noter que la valeur du drapeau `StateChange` dans la partie de données est "true" lorsque la fonctionnalité Preset Tour mentionnée inclut les modifications éventuelles de son état.

9 Service d'analyse

9.1 Généralités

Le présent article couvre les principaux aspects suivants de cette architecture:

- interface de module d'analyse;
- description de scène;
- interface de règles.

Le service d'analyse permet une configuration fine des règles individuelles et des modules d'analyse individuels (voir 9.2.3.5 et 9.3.4.7). Le Paragraphe 9.2 présente la description de scène basée sur XML, qui peut être transmise en continu sous forme de métadonnées à des clients par le protocole RTP tel que défini en 10.2.2.1.2. Des définitions supplémentaires de règles et de modules d'analyse figurent à l'Annexe C, l'Annexe D et l'Annexe E. Pour la définition du schéma, voir l'Article F.5.

9.2 Interface description de scène

9.2.1 Présentation

La présente spécification définit le schéma XML qui doit être utilisé pour coder des descriptions de scène par un dispositif. Le domaine d'application de la description de scène couvre des éléments de scène de base qui peuvent être affichés dans une disposition vidéo à l'utilisateur final, ainsi qu'un cadre pour des extensions spécifiques au fournisseur. L'Annexe A présente des éléments de scène supplémentaires qui peuvent être utilisés pour traiter des règles spécifiques au fournisseur.

9.2.2 Contenu associé aux trames

9.2.2.1 Généralités

L'entrée du moteur d'analyse est composée d'images provenant d'une source vidéo. Les éléments de scène extraits sont associés à l'image depuis laquelle ils ont été extraits. Une scène extraite se distingue de la description générale de la source vidéo traitée par le moteur d'analyse (informations telles que la ligne d'entrée vidéo, la résolution vidéo, le recadrage de trame, le taux de trames), la liaison temporelle de trames dans le flux d'entrée, et le positionnement spatial des éléments dans une trame.

La relation temporelle et spatiale d'éléments de scène par rapport à la source vidéo sélectionnée est décrite en 9.2.2.2 et en 9.2.2.3. L'aspect et le comportement des objets suivis sont décrits en 9.2.3.2. Les interactions entre les objets tels que séparations et fusions sont décrites en 9.2.3.3.

Un dispositif PTZ peut placer des informations relatives au panoramique, à l'inclinaison et au zoom au début d'une trame, ce qui permet à un client d'estimer les coordonnées 3D des éléments de scène. Ensuite, le système de coordonnées d'image peut être adapté avec un nœud de transformation facultatif, décrit en 9.2.2.2. Enfin, plusieurs descriptions d'objet peuvent être placées, et leur liaison peut être spécifiée dans un nœud ObjectTree. Les définitions sont présentées ci-dessous pour des raisons pratiques³:

```
<xs:complexType name="Frame">
  <xs:sequence>
    <xs:element name="PTZStatus" type="tt:PTZStatus"
      minOccurs="0"/>
    <xs:element name="Transformation" type="tt:Transformation"
      minOccurs="0"/>
    <xs:element name="Object" type="tt:Object" minOccurs="0"
      maxOccurs="unbounded"/>
    <xs:element name="ObjectTree" type="tt:ObjectTree" minOccurs="0"/>
    ...
  </xs:sequence>
  <xs:attribute name="UtcTime" type="xs:dateTime" use="required"/>
  ...
</xs:complexType>

<xs:element name="Frame" type="tt:Frame">
```

Le Paragraphe 9.2.2.2 explique comment les trames traitées par l'algorithme d'analyse sont référencées dans le flux d'analyse.

9.2.2.2 Relation temporelle

Étant donné que plusieurs éléments de scène peuvent être extraits de la même image, les éléments de scène figurent sous un nœud de trame qui établit le lien avec une image spécifique provenant de l'entrée vidéo. Le nœud de trame contient un attribut UtcTime obligatoire. Cet horodatage UtcTime doit permettre à un client de mettre en correspondance le nœud de trame avec exactement une trame vidéo. Par exemple, l'horodatage RTP de la trame vidéo codée correspondante doit donner lieu au même horodatage TUC après conversion. La synchronisation entre les flux vidéo et les flux de métadonnées est décrite plus en détail en 10.2.2.2.2.

Exemple:

```
<tt:Frame UtcTime="2008-10-10T12:24:57.321">
  ...
</tt:Frame>
...
<tt:Frame UtcTime="2008-10-10T12:24:57.521">
  ...
</tt:Frame>
```

9.2.2.3 Relation spatiale

La plupart des éléments de scène font référence à une partie d'une image à partir de laquelle des informations ont été extraites. Par exemple, si des objets sont suivis dans le temps, leur position dans chaque trame doit être spécifiée. Ces positions doivent être associées à un système de coordonnées. Le système de coordonnées par défaut est présenté à la Figure 9.

³ Noter que le schéma est inclus ici à titre informatif uniquement. Voir l'Article F.8 pour la définition de schéma normative.

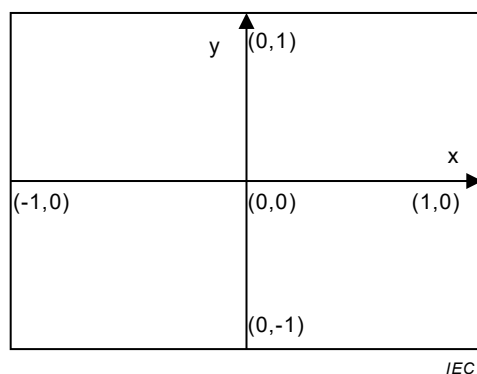


Figure 9 – Système de coordonnées de trame par défaut

La présente spécification permet la modification du système de coordonnées pour des nœuds individuels de l'arbre XML. En conséquence, chaque nœud de trame démarre avec le système de coordonnées par défaut. Chaque nœud enfant hérite du système de coordonnées le plus récent de son parent. Un nœud de transformation modifie le système de coordonnées le plus récent de son parent. Les spécifications de coordonnées sont toujours associées au système de coordonnées le plus récent du nœud parent.

La spécification définit des nœuds de transformation pour la mise à l'échelle et la translation. La description de scène contient des paramètres fictifs dans lesquels ces nœuds de transformation sont placés⁴.

```
<xs:complexType name="Transformation">
  <xs:sequence>
    <xs:element name="Translate" type="Vector" minOccurs="0"/>
    <xs:element name="Scale" type="Vector" minOccurs="0"/>
    ...
  </xs:sequence>
</xs:complexType>
```

Une description mathématique des systèmes de coordonnées et des transformations est présentée ci-après. Un système de coordonnées est composé d'un vecteur translationnel

$t = \begin{pmatrix} t_x \\ t_y \end{pmatrix}$ et d'une mise à l'échelle $s = \begin{pmatrix} s_x \\ s_y \end{pmatrix}$. Un point $p = \begin{pmatrix} p_x \\ p_y \end{pmatrix}$ défini par rapport à ce système de

coordonnées est transformé en un point correspondant $q = \begin{pmatrix} q_x \\ q_y \end{pmatrix}$ du système de coordonnées

par défaut par la formule suivante: $\begin{pmatrix} q_x \\ q_y \end{pmatrix} = \begin{pmatrix} p_x \cdot s_x + t_x \\ p_y \cdot s_y + t_y \end{pmatrix}$. De même, un vecteur v défini par

rapport au système de coordonnées est transformé en vecteur correspondant w du système de coordonnées par défaut au moyen de la formule suivante: $\begin{pmatrix} w_x \\ w_y \end{pmatrix} = \begin{pmatrix} v_x \cdot s_x \\ v_y \cdot s_y \end{pmatrix}$.

Un nœud de transformation contient un vecteur de mise à l'échelle $u = \begin{pmatrix} u_x \\ u_y \end{pmatrix}$ et un vecteur

translationnel $v = \begin{pmatrix} v_x \\ v_y \end{pmatrix}$ facultatifs. Si la mise à l'échelle n'est pas spécifiée, sa valeur par défaut

⁴ Noter que le schéma est inclus ici à titre informatif uniquement. Voir l'Article F.8 pour la définition de schéma normative.

$u = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$ est déterminée par hypothèse. De même, la valeur par défaut pour la translation est $v = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$. Le nœud de transformation modifie le système de coordonnées supérieur de la façon suivante:

$\begin{pmatrix} t'_x \\ t'_y \end{pmatrix} = \begin{pmatrix} v_x \cdot s_x + t_x \\ v_y \cdot s_y + t_y \end{pmatrix}$, $\begin{pmatrix} s'_x \\ s'_y \end{pmatrix} = \begin{pmatrix} u_x \cdot s_x \\ u_y \cdot s_y \end{pmatrix}$, où $\begin{pmatrix} t'_x \\ t'_y \end{pmatrix}$ et $\begin{pmatrix} s'_x \\ s'_y \end{pmatrix}$ remplacent le système de coordonnées supérieur.

Par exemple, les coordonnées de la description de scène sont présentées dans un système de coordonnées de trame, dans lequel les coordonnées du coin inférieur gauche sont (0,0) et celles du coin supérieur droit sont (320,240). Le nœud de trame s'apparente au code suivant, avec lequel la mise à l'échelle est définie au double de l'inverse de la largeur de trame et de la hauteur de trame:

```
<tt:Frame UtcTime="2008-10-10T12:24:57.321">
  <tt:Transformation>
    <tt:Translate x="-1.0" y="-1.0"/>
    <tt:Scale x="0.00625" y="0.00834"/>
  </tt:Transformation>
  ...
</tt:Frame>
```

9.2.3 Éléments de scène

9.2.3.1 Généralités

Le Paragraphe 9.2.3 concerne les éléments de scène générés par des algorithmes de poursuite d'objets et définit la manipulation d'objet et les formes d'objet pour ceux-ci.

Les trames dans lesquelles aucun objet n'a été détecté peuvent être ignorées dans la description de scène afin d'économiser la bande passante, tant que la dernière trame dans la description de scène est également vide. Il est recommandé que le dispositif envoie périodiquement l'événement de description de scène même s'il est vide, afin d'indiquer que le moteur d'analyse est opérationnel. Le dispositif doit envoyer une description de scène si un SynchronizationPoint est demandé pour le flux correspondant.

Lorsque le récepteur d'une description de scène reçoit une trame vide, il convient qu'il considère par hypothèse que toutes les trames suivantes sont également vides tant qu'il n'a pas reçu la trame non vide suivante. Lorsque la dernière trame reçue n'est pas vide, il convient que le récepteur considère par hypothèse qu'une description de la trame traitée suivante va être transmise.

9.2.3.2 Objets

Les objets sont identifiés par leur ObjectID. Les caractéristiques relatives à un objet particulier sont collectées dans un nœud d'objet avec l'ObjectID correspondant en tant qu'attribut. Les associations d'objets telles que les objets renommés (Object Renaming), divisions d'objets (Object Splits), fusions d'objets (Object Merges) et suppressions d'objets (Object Deletions) sont représentées dans un nœud ObjectTree séparé. Un ObjectID est créé de manière implicite lors de la première apparition de l'ObjectID dans un nœud d'objet⁵.

⁵ Noter que le schéma est inclus ici à titre informatif uniquement. Voir l'Article F.8 pour la définition de schéma normative.


```

<xs:complexType name="ObjectId">
  <xs:attribute name="ObjectId" type="xs:integer"/>
</xs:complexType>

<xs:complexType name="Object">
  <xs:complexContent>
    <xs:extension base="ObjectId">
      <xs:sequence>
        <xs:element name="Appearance" type="Appearance" minOccurs="0"/>
        <xs:element name="Behaviour" type="Behaviour" minOccurs="0"/>
        ...
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

Le nœud d'objet comporte deux paramètres fictifs pour les informations d'aspect et de comportement. Le nœud d'aspect commence par un nœud de transformation facultatif qui peut être utilisé pour passer d'un système de coordonnées orienté trame à un système de coordonnées orienté objet. Ensuite, la forme d'un objet peut être spécifiée. Si un objet est détecté dans une trame, il convient que les informations de forme soient présentes dans la description d'aspect. L'algorithme d'analyse peut ajouter des nœuds d'objet pour des objets non visibles actuellement, s'il est capable de déduire les informations concernant cet objet. Dans de tels cas, la description de forme peut être omise.

Des caractéristiques d'objets comme la forme (voir 9.2.3.4), la couleur (voir 9.2.3.5) et la classe d'objet (voir 9.2.3.6) peuvent être ajoutées au nœud d'aspect.

La présente spécification définit deux comportements normalisés pour les objets: Removed (Supprimé) ou Idle (stationnaire). Ces comportements doivent être répertoriés en tant que nœuds enfants du nœud de comportement d'un objet. La présence d'un nœud supprimé ou stationnaire ne supprime pas automatiquement l'ObjectId correspondant, ce qui permet de réutiliser le même ObjectId lorsque l'objet recommence à se déplacer.

Un objet marqué avec le comportement supprimé spécifie l'emplacement duquel l'objet réel a été supprimé. Il convient de ne pas utiliser le marqueur en tant que comportement de l'objet supprimé. La suppression d'un objet peut être détectée même si l'action d'enlèvement de celui-ci n'a pas été détectée.

Les objets précédemment en mouvement peuvent être marqués comme étant Idle (stationnaires) afin d'indiquer que l'objet ne se déplace plus. Tant que ces objets ne sont pas modifiés, ils ne sont plus répertoriés dans la description de scène. Lorsqu'un objet stationnaire apparaît de nouveau dans la description de scène, le drapeau Idle est automatiquement enlevé.

Exemple

```

...
<tt:Frame UtcTime="2008-10-10T12:24:57.321">
  <tt:Transformation>
    <tt:Translate x="-1.0" y="-1.0"/>
    <tt:Scale x="0.003125" y="0.00416667"/>
  </tt:Transformation>
  <tt:Object ObjectId="12">
    <tt:Appearance>
      <tt:Shape>
        <tt:BoundingBox left="20.0" top="30.0" right="100.0" bottom="80.0"/>
        <tt:CenterOfGravity x="60.0" y="50.0"/>
      </tt:Shape>
    </tt:Appearance>
  </tt:Object>
</tt:Frame>
...
<tt:Frame UtcTime="2008-10-10T12:24:57.421">

```

```

<tt:Transformation>
  <tt:Translate x="-1.0" y="-1.0"/>
  <tt:Scale x="0.003125" y="0.00416667"/>
</tt:Transformation>
<tt:Object ObjectId="12">
  <tt:Appearance>
    <tt:Shape>
      <tt:BoundingBox left="20.0" top="30.0" right="100.0" bottom="80.0"/>
      <tt:CenterOfGravity x="60.0" y="50.0"/>
    </tt:Shape>
  </tt:Appearance>
  <tt:Behaviour>
    <tt:Idle/>
  </tt:Behaviour>
</tt:Object>
</tt:Frame>

...

<tt:Frame UtcTime="2008-10-10T12:24:57.521">
  <tt:Transformation>
    <tt:Translate x="-1.0" y="-1.0"/>
    <tt:Scale x="0.003125" y="0.00416667"/>
  </tt:Transformation>
</tt:Frame>

...

<tt:Frame UtcTime="2008-10-10T12:24:57.621">
  <tt:Transformation>
    <tt:Translate x="-1.0" y="-1.0"/>
    <tt:Scale x="0.003125" y="0.00416667"/>
  </tt:Transformation>
  <tt:Object ObjectId="12">
    <tt:Appearance>
      <tt:Shape>
        <tt:BoundingBox left="25.0" top="30.0" right="105.0" bottom="80.0"/>
        <tt:CenterOfGravity x="65.0" y="50.0"/>
      </tt:Shape>
    </tt:Appearance>
  </tt:Object>
</tt:Frame>

...

<tt:Frame UtcTime="2008-10-10T12:24:57.721">
  <tt:Transformation>
    <tt:Translate x="-1.0" y="-1.0"/>
    <tt:Scale x="0.003125" y="0.00416667"/>
  </tt:Transformation>
  <tt:Object ObjectId="19">
    <tt:Appearance>
      <tt:Shape>
        <tt:BoundingBox left="20.0" top="30.0" right="100.0" bottom="80.0"/>
        <tt:CenterOfGravity x="60.0" y="50.0"/>
      </tt:Shape>
    </tt:Appearance>
    <tt:Behaviour>
      <tt:Removed/>
    </tt:Behaviour>
  </tt:Object>
</tt:Frame>

```

9.2.3.3 Arbre d'objets

Lorsque deux objets sont trop proches l'un de l'autre, de sorte que l'analyse ne peut plus les suivre individuellement, il convient d'indiquer une fusion d'objets en ajoutant un nœud de fusion au nœud ObjectTree du nœud de trame. Le nœud de fusion contient un nœud From (De) qui répertorie les ObjectIds de fusion et un nœud To (Vers) contenant l'ObjectId. L'objet fusionné est utilisé dans les trames ultérieures en tant qu'ID de suivi. Si l'algorithme d'analyse détecte

qu'un objet masque les autres, et qu'il est capable de suivre plus avant cet objet, il convient de placer l'objet masquant dans le nœud To.

La séparation d'objets est indiquée par un nœud Split (Séparation). Dans ce cas, le nœud From contient un seul ObjectId représentant l'objet qui est divisé dans la trame actuelle. Les objets se séparant de cet objet divisé sont répertoriés dans le nœud To. L'ObjectId du nœud From peut réapparaître dans le nœud To, si cet objet masquait les autres et que l'algorithme d'analyse était capable de suivre cet objet pendant le masquage.

Un objet ne doit pas être impliqué dans une opération de fusion pour faire partie d'une opération de séparation. Par exemple, si un objet se déplace conjointement avec une personne et que la personne laisse l'objet quelque part, l'objet peut être détecté la première fois par l'analyse lorsque la personne s'éloigne de l'objet. Dans ces cas, le premier aspect de l'objet peut être combiné avec une opération de séparation.

Lorsqu'un objet fusionné réapparaît sous la forme d'un nœud d'objet dans une trame ultérieure sans indication de séparation, cet objet est alors implicitement séparé. Toutefois, l'algorithme d'analyse ne peut pas déterminer la provenance de l'objet séparé.

Un algorithme d'analyse peut suivre et mémoriser un nombre limité d'objets. Afin d'indiquer qu'un certain objet a été supprimé de la mémoire de l'algorithme et qu'il ne réapparaîtra donc jamais, la description de scène peut contenir un nœud Delete (Supprimer) dans le nœud ObjectTree.

Si l'algorithme d'analyse ne peut pas déterminer l'identité d'un objet lors d'une opération de séparation, il convient qu'il utilise un nouvel ObjectId. Lorsque l'algorithme a rassemblé des preuves suffisantes quant à l'identité de cet objet, il peut modifier l'ObjectId par l'intermédiaire de l'opération Rename (Renommer). L'opération Rename peut également être utilisée lorsqu'un objet entre de nouveau dans la scène et que sa véritable identité est découverte au bout d'un certain temps.

Un ObjectId supprimé ne doit pas être réutilisé dans la description de scène tant que son conteneur n'est pas de nouveau utilisé.

Exemple:

```
<tt:Frame UtcTime="2008-10-10T12:24:57.321">
  <tt:Object ObjectId="12">
    ...
  </tt:Object>
  <tt:Object ObjectId="17">
    ...
  </tt:Object>
</tt:Frame>
```

```
<tt:Frame UtcTime="2008-10-10T12:24:57.421">
  <tt:Object ObjectId="12">
    ...
  </tt:Object>
  <tt:ObjectTree>
    <tt:Merge>
      <tt:From ObjectId="12"/>
      <tt:From ObjectId="17"/>
      <tt:To ObjectId="12"/>
    </tt:Merge>
  </tt:ObjectTree>
</tt:Frame>
```

```
<tt:Frame UtcTime="2008-10-10T12:24:57.521">
  <tt:Object ObjectId="12">
```

```

...
</tt:Object>
</tt:Frame>

<tt:Frame UtcTime="2008-10-10T12:24:57.621">
  <tt:Object ObjectId="12">
    ...
  </tt:Object>
  <tt:Object ObjectId="17">
    ...
  </tt:Object>
  <tt:ObjectTree>
    <tt:Split>
      <tt:From ObjectId="12"/>
      <tt:To ObjectId="17"/>
      <tt:To ObjectId="12"/>
    </tt:Split>
  </tt:ObjectTree>
</tt:Frame>

```

9.2.3.4 Descripteur de forme

Des informations de forme doivent être placées au-dessous du nœud de forme facultatif dans un nœud d'aspect d'objet. S'il est présent, le nœud de forme contient des informations sur la détection de l'objet en question dans la trame spécifiée. Un nœud de forme doit contenir au moins deux nœuds représentant le BoundingBox et le CenterOfGravity de l'objet détecté.

Le BoundingBox épais est affiné avec des nœuds enfants supplémentaires, représentant chacun une primitive de forme. En présence de plusieurs primitives de forme, leur union définit la forme de l'objet. Dans la présente spécification, un descripteur de polygone générique est fourni.

Les polygones qui décrivent la forme d'un objet doivent être de simples polygones définis par une liste de points.

Deux points consécutifs (le dernier point étant relié au premier) dans la liste définissent un segment de droite. L'ordre des points doit être choisi de sorte que la zone délimitée de l'objet puisse être présente sur le côté gauche de tous les segments de droite. La ligne brisée définie par la liste de points ne doit pas présenter d'auto-intersection.

Exemple:

```

<tt:Frame UtcTime="2008-10-10T12:24:57.321">
  <tt:Transformation>
    <tt:Translate x="-1.0" y="-1".0/>
    <tt:Scale x="0.003125" y="0.00416667"/>
  </tt:Transformation>
  <tt:Object ObjectId="12">
    <tt:Appearance>
      <tt:Shape>
        <tt:BoundingBox left="20.0" top="30.0" right="100.0" bottom="80.0"/>
        <tt:CenterOfGravity x="60.0" y="50.0"/>
        <tt:Polygon>
          <tt:Point x="20.0" y="30.0"/>
          <tt:Point x="100.0" y="30.0"/>
          <tt:Point x="100.0" y="80.0"/>
          <tt:Point x="20.0" y="80.0"/>
        </tt:Polygon>
      </tt:Shape>
    </tt:Appearance>
  </tt:Object>
</tt:Frame>

```

9.2.3.5 Descripteur de couleur

Un descripteur de couleur est défini comme étant un élément facultatif du nœud d'aspect d'un nœud d'objet. Le descripteur de couleur est défini par une liste de groupes de couleurs, chacun étant constitué d'une valeur de couleur, d'une pondération facultative et d'une matrice de covariance facultative. Le descripteur de couleur ne précise pas comment les groupes de couleurs sont créés. Ils peuvent représenter des tranches d'un histogramme de couleurs ou le résultat d'un algorithme de groupage.

Les couleurs sont représentées par des vecteurs tridimensionnels. De plus, l'espace de couleurs de chaque vecteur de couleur peut être spécifié par un attribut d'espace de couleurs. Si cet attribut est absent, l'espace de couleurs YCbCr est utilisé par défaut. Celui-ci fait référence à la gamme "sRGB" avec la transformation de RVB en YCbCr conformément à l'ISO/IEC 10918-1, également appelé JPEG. L'URI d'espace de couleurs pour l'espace de couleurs YcbCr est www.onvif.org/ver10/colourspace/YcbCr.

Un exemple de métadonnées contenant des informations de couleur de l'objet détecté est donné ci-dessous.

```
<tt:Frame UtcTime="2010-09-10T12:24:57.721">

  <tt:Object ObjectId="34">
    <tt:Appearance>
      <tt:Shape>
        <tt:BoundingBox left="20.0" top="30.0" right="100.0" bottom="80.0"/>
        <tt:CenterOfGravity x="60.0" y="50.0"/>
      </tt:Shape>
      <tt:Color>
        <tt:ColorCluster>
          <tt:Color X="58" Y="105" Z="212"/>
          <tt:Covariance XX="7.2" YY="6" ZZ="3"/>
          <tt:Weight>90</tt:Weight>
        </tt:ColorCluster>
        <tt:ColorCluster>
          <tt:Color X="165" Y="44" Z="139"/>
          <tt:Covariance XX="4" YY="4" ZZ="4"/>
          <tt:Weight>5</tt:Weight>
        </tt:ColorCluster>
      </tt:Color>
    </tt:Appearance>
  </tt:Object>

</tt:Frame>
```

Le descripteur de couleur (Colour Descriptor) contient les couleurs représentatives de l'objet/zone détectés. Les couleurs représentatives sont calculées à chaque fois à partir de l'image de l'objet/zone détectés. Chaque valeur de couleur représentative est un vecteur de l'espace de couleurs spécifié. La pondération (Weight) de couleur représentative désigne la fraction de pixels attribués à la couleur représentative. La covariance de couleur décrit la variation des valeurs de couleur autour de la valeur de couleur représentative dans l'espace de couleurs. Ainsi, la couleur représentative désigne une zone de l'espace de couleurs. Le Tableau 2 répertorie les valeurs acceptables de l'attribut Colourspace

Tableau 2 – Valeurs relatives à l'espace de nom Colourspace

URI d'espace de nom	Description
http://www.onvif.org/ver10/colourspace/YcbCr	YcbCr colourspace
http://www.onvif.org/ver10/colourspace/CIEUV	CIE LUV
http://www.onvif.org/ver10/colourspace/CIELAB	CIE 1976 (L*a*b*)
http://www.onvif.org/ver10/colourspace/HSV	HSV colourspace

9.2.3.6 Descripteur de classe d'objets

Un descripteur de classe est défini comme étant un élément facultatif du nœud d'aspect d'un nœud d'objet. Le descripteur de classe (Class Descriptor) est défini par une liste de classes d'objets conjointement avec une probabilité que l'objet correspondant appartienne à cette classe. La somme des probabilités ne doit pas dépasser 1.

L'exemple suivant présente un échantillon de métadonnées d'objet contenant des informations de classe d'objets concernant l'objet détecté:

```
<tt:Frame UtcTime="2010-11-10T12:24:57.721">

<tt:Object ObjectId="22">
  <tt:Appearance>
    <tt:Shape>
      <tt:BoundingBox left="20.0" top="30.0" right="100.0" bottom="80.0"/>
      <tt:CenterOfGravity x="60.0" y="50.0"/>
    </tt:Shape>
    <tt:Class>
      <tt:ClassCandidate>
        <tt:Type>Vehicle</tt:Type>
        <tt:Likelihood>0.93</tt:Likelihood >
      </tt:ClassCandidate>
    </tt:Class>
  </tt:Appearance>
</tt:Object>

</tt:Frame>
```

Le mécanisme d'extensibilité du schéma d'origine introduit les nouvelles étiquettes de classes d'objets par la révision du type de données ClassDescriptorExtension. Un exemple d'introduction d'un nouveau type de classe et d'utilisation de métadonnées d'objets est présenté ci-dessous.

```
<tt:Frame UtcTime="2011-08-11T12:44:57.721">
  <tt:Object ObjectId="122">
    <tt:Appearance>
      <tt:Shape>
        <tt:BoundingBox left="20.0" top="30.0" right="80.0" bottom="40.0"/>
        <tt:CenterOfGravity x="50.0" y="45.0"/>
      </tt:Shape>
      <tt:Class>
        <tt:ClassCandidate>
          <tt:Type>Other</tt:Type>
          <tt:Likelihood>0.81</tt:Likelihood >
        </tt:ClassCandidate>
        <tt:Extension>
          <tt:OtherTypes>
            <tt:Type>LicensePlate</tt:Type>
            <tt:Likelihood>0.81</tt:Likelihood >
          </tt:OtherTypes>
        </tt:Extension>
      </tt:Class>
    </tt:Appearance>
  </tt:Object>
</tt:Frame>
```

Les anciennes mises en oeuvre peuvent ignorer l'élément Extension. Les nouvelles mises en oeuvre peuvent analyser et utiliser les données de l'élément Extension. Cette approche conserve la définition du type de données ClassType tout en élargissant la signification lorsque la valeur est "Other". La spécification doit répertorier les valeurs acceptables applicables à l'élément OtherType/Type, et les mises en oeuvre peuvent alors valider les valeurs acceptables.

9.2.3.7 Descripteur de mouvement de cellules

La description de scène d'un mouvement de cellules contient les cellules dont le mouvement est détecté. Les colonnes et les lignes de la grille de cellules sont fournies afin de décoder l'élément base64Binary.

L'élément "Transformation" est utilisé pour la relation spatiale, voir 9.2.2.3. Les attributs des cellules sont définis dans le Tableau 3. L'angle gauche supérieur constitue le point de départ de la grille de cellules. Dimension *X* de gauche à droite et dimension *Y* du haut vers le bas, voir Figure 9.

Cet exemple contient un fragment de flux de métadonnées utilisant le détecteur de mouvement de cellules (Cell Motion Detector).

```
<?xml version="1.0" encoding="UTF-8"?>
<tt:MetaDataStream
  xmlns:tt="http://www.onvif.org/ver10/schema">

  <tt:VideoAnalytics>
    <tt:Frame UtcTime="2010-10-20T12:24:57.321">
      <tt:Transformation>
        <tt:Translate x="-0.66666" y="-0.6"/>
        <tt:Scale x="0.166666" y="-0.2"/>
      </tt:Transformation>
      <tt:Extension>
        <tt:MotionInCells Columns="8" Rows="6" Cells="AAD8AA==" />
      </tt:Extension>
    </tt:Frame>
    <tt:Frame UtcTime="2010-10-20T12:24:57.621">
      ...
    </tt:Frame>
  </tt:VideoAnalytics>

  <tt:Event>
    <wsnt:NotificationMessage>
      <wsnt:Topic Dialect="...Concrete">
        tns1:RuleEngine/CellMotionDetector/Motion
      </wsnt:Topic>
      <wsnt:Message>
        <tt:Message UtcTime="2010-10-20T12:24:57.628">
          <tt:Source>
            <tt:SimpleItem Name="VideoSourceConfigurationToken" Value="1"/>
            <tt:SimpleItem Name="VideoAnalyticsConfigurationToken" Value="1"/>
            <tt:SimpleItem Name="Rule" Value="MotionInDefinedCells"/>
          </tt:Source>
          <tt>Data>
            <tt:SimpleItem Name="IsMotion" Value="true"/>
          </tt>Data>
        </tt:Message>
      </wsnt:Message>
    </wsnt:NotificationMessage>
  </tt:Event>

</tt:MetaDataStream>
```

Tableau 3 – Description des attributs du type MotionInCells

Attribut	Description
Columns (Colonnes)	Nombre de colonnes de la grille de cellules (dimension X)
Rows (Lignes)	Nombre de lignes de la grille de cellules (dimension Y).
Cells (Cellules)	Une valeur "1" désigne une cellule dont le mouvement est détecté et une valeur "0" désigne une cellule vide. La première cellule se situe dans l'angle gauche supérieur. Ensuite, l'ordre des cellules s'effectue tout d'abord de gauche à droite, puis de haut en bas. Si le nombre de cellules n'est pas un multiple de 8, le dernier octet est rempli de zéros. Le codage des informations est un codage à longueur de ligne conforme au codage Packbit défini dans l'ISO 12639 (TIFF, Révision 6.0).

9.3 Interface de règles

9.3.1 Généralités

Une structure XML est introduite en 9.3.2 pour communiquer la configuration des règles. Le Paragraphe 9.3.3 spécifie un langage de description de la configuration d'un type de règle spécifique. Un dispositif qui met en œuvre un moteur de règles peut prendre en charge les règles décrites à l'Annexe A. Le Paragraphe 9.3.4 introduit des opérations de gestion de ces règles. Si le dispositif prend en charge un moteur de règles (Rule Engine), il doit mettre en œuvre l'interface de règles complète.

9.3.2 Représentation des règles

La configuration d'une règle exige l'utilisation de deux attributs, l'un spécifiant le nom et l'autre le type de la règle. Les différents paramètres de configuration sont répertoriés sous l'élément de paramètres de l'élément de règle. Chaque paramètre est soit un SimpleItem, soit un ElementItem. L'attribut de nom de chaque élément doit être unique dans la liste des paramètres. SimpleItems comporte un attribut Value (Valeur) supplémentaire contenant la valeur du paramètre. La valeur des ElementItems est donnée par l'élément enfant d'ElementItem. Il est recommandé de représenter autant de paramètres que possible par des SimpleItems.

L'exemple suivant présente une configuration d'analyse complète contenant deux règles:

```
<tt:RuleEngineConfiguration>
  <tt:Rule Name="MyLineDetector" Type="tt:LineDetector">
    <tt:Parameters>
      <tt:SimpleItem Name="Direction" Value="Any"/>
      <tt:ElementItem Name="Segments">
        <tt:Polyline>
          <tt:Point x="10.0" y="50.0"/>
          <tt:Point x="100.0" y="50.0"/>
        </tt:Polyline>
      </tt:ElementItem>
    </tt:Parameters>
  </tt:Rule>
  <tt:Rule Name="MyFieldDetector" Type="tt:FieldDetector">
    <tt:Parameters>
      <tt:ElementItem Name="Field">
        <tt:Polygon>
          <tt:Point x="10.0" y="50.0"/>
          <tt:Point x="100.0" y="50.0"/>
          <tt:Point x="100.0" y="150.0"/>
        </tt:Polygon>
      </tt:ElementItem>
    </tt:Parameters>
  </tt:Rule>
</tt:RuleEngineConfiguration>
```

9.3.3 Langage de description de règles

La description d'une règle contient les informations de type de tous les paramètres appartenant à un certain type de règle, ainsi que la description de la sortie générée par une telle règle. La sortie du moteur de règles est composée d'événements, qui peuvent être utilisés dans un moteur d'événements (Event Engine) ou auxquels un client peut s'abonner.

Les définitions sont présentées ci-dessous pour des raisons pratiques⁶:

```
<xs:complexType name="ConfigDescription">
  <xs:sequence>
    <xs:element name="Parameters"
      type="tt:ItemListDescription"/>
    <xs:element name="Messages" minOccurs="0" maxOccurs="unbounded">
      <xs:complexType>
        <xs:complexContent>
          <xs:extension base="tt:MessageDescription">
            <xs:sequence>
              <xs:element name="ParentTopic" type="xs:string"/>
            </xs:sequence>
          </xs:extension>
        </xs:complexContent>
      </xs:complexType>
    </xs:element>
    ...
  </xs:sequence>
  <xs:attribute name="Name" type="xs:QName" use="required"/>
</xs:complexType>

<xs:complexType name="ItemListDescription">
  <xs:sequence>
    <xs:element name="SimpleItemDescription" minOccurs="0"
      maxOccurs="unbounded">
      <xs:complexType>
        <xs:attribute name="Name" type="xs:string" use="required"/>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
```

⁶ Noter que le schéma est inclus ici à titre informatif uniquement. Voir l'Article F.8 pour la définition de schéma normative.

```

    <xs:attribute name="Type" type="xs:Qname" use="required"/>
  </xs:complexType>
</xs:element>
<xs:element name="ElementItemDescription" minOccurs="0"
  maxOccurs="unbounded">
  <xs:complexType>
    <xs:attribute name="Name" type="xs:string" use="required"/>
    <xs:attribute name="Type" type="xs:Qname" use="required"/>
  </xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>

```

Les paramètres d'un certain type de règle sont répertoriés sous l'élément Parameters. Tous les paramètres sont des SimpleItems ou des ElementItems, et peuvent être décrits par une SimpleItemDescription ou une ElementItemDescription. Les deux ItemDescriptions contiennent un attribut de nom pour identifier le paramètre et un attribut de type pour référencer un type de schéma XML spécifique. Dans le cas de la SimpleItemDescription, l'attribut de type doit référencer une définition de schéma SimpleType. Dans le cas de la ElementItemDescription, l'attribut de type doit référencer une déclaration globale d'élément d'un schéma XML.

La sortie générée par ce type de règle est décrite dans plusieurs éléments Messages. Chaque élément Messages contient une description de la charge utile de message conformément au langage de description de message (Message Description Language) décrit en détail dans la spécification principale de l'ONVIF. De plus, l'élément Messages doit contenir un élément ParentTopic indiquant la rubrique (Topic) à laquelle un client doit s'abonner pour recevoir cette sortie spécifique. La rubrique doit être spécifiée sous la forme d'une expression de rubrique concrète (Concrete Topic Expression).

9.3.4 Opérations sur les règles

9.3.4.1 Généralités

Si le dispositif prend en charge un moteur de règles tel que défini par l'ONVIF, il doit alors mettre en œuvre les opérations suivantes de gestion des règles. Les opérations Create/Delete/Modify (Créer/Supprimer/Modifier) sont atomiques, ce qui signifie que toutes les modifications peuvent être traitées ou que l'opération complète doit échouer.

9.3.4.2 GetSupportedRules

Le dispositif doit indiquer les règles qu'il prend en charge en mettant en œuvre l'opération suivante. Il retourne une liste de descriptions de règle conformément au langage de description de règle décrit en 9.3.3. De plus, il contient une liste d'URL qui indiquent l'emplacement des fichiers de schéma. Ces fichiers de schéma décrivent les types et éléments utilisés dans les descriptions de règles. Les descriptions de règles qui référencent les types ou les éléments importés des fichiers de schémas définis par l'ONVIF peuvent ne pas répertorier de manière explicite ces fichiers.

DEMANDE:

- **ConfigurationToken [tt:ReferenceToken]**

Jeton de la configuration d'analyse pour laquelle il convient de répertorier les règles prises en charge.

RÉPONSE:

- **SupportedRules [tt:SupportedRules]**
- **Règles prises en charge pour la configuration d'analyse**

DÉFAUTS:

- **env:Sender – ter:InvalidArgVal – ter:NoConfig**
La configuration d'analyse demandée n'existe pas.

CLASSE D'ACCÈS:

READ_MEDIA

9.3.4.3 Get Rules

L'opération suivante extrait les règles actuellement installées:

DEMANDE:

- **ConfigurationToken [tt:ReferenceToken]**
Jeton de la configuration d'analyse pour laquelle il convient de répertorier les règles.

RÉPONSE:

- **Rule – facultatif, non limité [tt:Config]**
Liste de règles installées pour la configuration spécifiée.

DÉFAUTS:

- **env:Sender – ter:InvalidArgVal – ter:NoConfig**
La configuration d'analyse demandée n'existe pas.

CLASSE D'ACCÈS:

READ_MEDIA

9.3.4.4 CreateRules

L'opération suivante ajoute des règles à une AnalyticsConfiguration. Si toutes les règles ne peuvent pas être créées comme demandé, le dispositif répond par un message de défaut.

DEMANDE:

- **ConfigurationToken [tt:ReferenceToken]**
Jeton de la configuration d'analyse pour laquelle il convient d'installer les règles.
- **Rule – non limité [tt:Config]**
Liste de règles à installer pour la configuration spécifiée.

RÉPONSE:

Ceci est un message vide.

DÉFAUTS:

- **env:Sender – ter:InvalidArgVal – ter:NoConfig**
La configuration d'analyse demandée n'existe pas.
- **env:Sender – ter:InvalidArgVal – ter:InvalidRule**
La configuration de règles proposée n'est pas valide sur le dispositif.
- **env:Sender – ter:InvalidArgVal – ter:RuleAlreadyExistent**
La configuration comporte déjà une règle avec le même nom.
- **env:Receiver – ter:Action – ter:TooManyRules**
Espace insuffisant dans le dispositif pour ajouter les règles à la configuration.
- **env:Receiver – ter:Action – ter:ConfigurationConflict**
Le dispositif ne peut pas créer les règles sans créer une configuration conflictuelle.

CLASSE D'ACCÈS:

ACTUATE

9.3.4.5 ModifyRules

L'opération suivante modifie plusieurs règles. Si toutes les règles ne peuvent pas être modifiées comme demandé, le dispositif répond par un message de défaut.

DEMANDE:

- **ConfigurationToken [tt:ReferenceToken]**
Jeton de la configuration d'analyse pour laquelle il convient de modifier les règles.
- **Rule – non limité [tt:Config]**
Liste de règles à modifier pour la configuration spécifiée.

RÉPONSE:

- **Ceci est un message vide.**

DÉFAUTS:

- **env:Sender – ter:InvalidArgVal – ter:NoConfig**
La configuration d'analyse demandée n'existe pas.
- **env:Sender – ter:InvalidArgVal – ter:InvalidRule**
La configuration de règles proposée n'est pas valide sur le dispositif.
- **env:Sender – ter:InvalidArgVal – ter:RuleNotExist**
Le ou les noms de règles n'existent pas.
- **env:Receiver – ter:Action – ter:ConfigurationConflict**
Le dispositif ne peut pas modifier les règles sans créer une configuration conflictuelle.

CLASSE D'ACCÈS:

ACTUATE

9.3.4.6 DeleteRules

L'opération suivante supprime plusieurs règles. Si toutes les règles ne peuvent pas être supprimées comme demandé, le dispositif répond par un message de défaut.

DEMANDE:

- **ConfigurationToken [tt:ReferenceToken]**
Jeton de la configuration d'analyse pour laquelle il convient de supprimer les règles.
- **RuleName – non limité [xs:string]**
Liste de règles à supprimer pour la configuration spécifiée.

RÉPONSE:

Ceci est un message vide.

DÉFAUTS:

- **env:Sender – ter:InvalidArgVal – ter:NoConfig**
La configuration d'analyse demandée n'existe pas.
- **env:Sender – ter:InvalidArgVal – ter:RuleNotExistent**
Le ou les noms de règles n'existent pas.
- **env:Sender – ter:Action – ter:FixedRules**
Au moins une des règles demandées est fixée et ne peut pas être supprimée.
- **env:Receiver – ter:Action – ter:ConfigurationConflict**
Le dispositif ne peut pas supprimer les règles sans créer une configuration conflictuelle.

CLASSE D'ACCÈS:

ACTUATE**9.3.4.7 GetRuleOptions**

L'opération suivante retourne les options applicables aux règles prises en charge qui spécifient un attribut Option.

DEMANDE:

- **RuleType – facultatif [xs:QName]**
Type de règle facultatif pour lequel les options doivent être extraites. S'il est omis, tous les types d'options de règles comportant toutes les options de règles prises en charge par le dispositif doivent être retournés (le cas échéant).
- **ConfigurationToken [tt:ReferenceToken]**
Jeton de la configuration d'analyse pour laquelle il convient de répertorier les règles prises en charge.

RÉPONSE:

- **RuleOptions – facultatif, non limité [tan:ConfigOptions]**
Règles prises en charge pour la configuration d'analyse

DÉFAUTS:

- **env:Sender – ter:InvalidArgVal – ter:NoConfig**
La configuration d'analyse demandée n'existe pas.
- **env:Sender – ter:InvalidArgVal – ter:RuleNotExistent**
Le type d'option de règle n'existe pas.

CLASSE D'ACCÈS:

READ_MEDIA

9.4 Interface de modules d'analyse

9.4.1 Généralités

Le Paragraphe 9.4.2 définit une structure XML qui communique la configuration de modules d'analyse. Le Paragraphe 9.4.3 définit le langage de description de la configuration d'un module d'analyse spécifique. Le Paragraphe 9.4.4 définit les opérations exigées par l'interface de modules d'analyse. Si le dispositif prend en charge un moteur d'analyse tel que défini par l'ONVIF, il doit mettre en œuvre l'interface complète de modules d'analyse.

9.4.2 Configuration de module d'analyse

La configuration de module d'analyse est identique à la configuration de règle décrite en 9.3.2. L'exemple suivant présente une configuration possible d'un élément ObjectTracker spécifique au fournisseur. Ce dispositif de poursuite permet de configurer la taille d'objet minimale et la taille d'objet maximale concernant la géométrie de trame traitée.

```
<tt:AnalyticsEngineConfig>
  <tt:AnalyticsModule Name="MyObjectTracker" Type="nn:ObjectTracker">
    <tt:Parameters>
      <tt:SimpleItem Name="MinObjectWidth" Value="0.01"/>
      <tt:SimpleItem Name="MinObjectHeight" Value="0.01"/>
      <tt:SimpleItem Name="MaxObjectWidth" Value="0.5"/>
      <tt:SimpleItem Name="MaxObjectHeight" Value="0.5"/>
    </tt:Parameters>
  </tt:AnalyticsModule>
</tt:AnalyticsEngineConfig>
```

9.4.3 Langage de description de module d'analyse

Le module d'analyse réutilise le langage de description de règle décrit en 9.3.3. L'élément AnalyticsModuleDescription suivant remplace l'élément RuleDescription:

```
<xs:element name="AnalyticsModuleDescription"
  type="tt:ConfigDescription"/>
```

De même que les règles, les modules d'analyse génèrent des événements et doivent être répertoriés dans la description de module d'analyse. La description suivante correspond à l'exemple de 9.4.2. L'exemple de module génère un événement SceneTooCrowded lorsque la scène devient trop complexe pour le module.

```
<tt:AnalyticsModuleDescription Name="nn:ObjectTracker">
  <tt:Parameters>
    <tt:SimpleItemDescription Name="MinObjectWidth" Type="xs:float"/>
    <tt:SimpleItemDescription Name="MinObjectHeight" Type="xs:float"/>
    <tt:SimpleItemDescription Name="MaxObjectWidth" Type="xs:float"/>
    <tt:SimpleItemDescription Name="MaxObjectHeight" Type="xs:float"/>
  </tt:Parameters>
  <tt:Messages>
    <tt:Source>
      <tt:SimpleItemDescription Name="VideoSourceConfiguration"
        Type="tt:ReferenceToken"/>
      <tt:SimpleItemDescription Name="AnalyticsConfiguration"
        Type="tt:ReferenceToken"/>
      <tt:SimpleItemDescription Name="AnalyticsModule" Type="xs:string"/>
    </tt:Source>
    <tt:ParentTopic>
      tns1:VideoAnalytics/nn:ObjectTracker/SceneTooCrowded
    </tt:ParentTopic>
  </tt:Messages>
</tt:RuleDescription>
```

9.4.4 Opérations sur les modules d'analyse

9.4.4.1 Généralités

Si le dispositif prend en charge un moteur d'analyse comme défini par l'ONVIF, il doit prendre en charge les opérations suivantes pour gérer les modules d'analyse. Les opérations Create/Delete/Modify doivent être atomiques. Toutes les modifications peuvent être traitées ou l'opération complète doit échouer.

9.4.4.2 GetSupportedAnalyticsModules

Le dispositif indique les modules d'analyse qu'il prend en charge en mettant en œuvre l'opération GetSupportedAnalyticsModule. Il retourne une liste de modules d'analyse conformément au langage de description de module d'analyse décrit en 9.3.3. De plus, il contient une liste d'URL qui indiquent l'emplacement des fichiers de schéma. Ces fichiers de schéma décrivent les types et éléments utilisés dans les descriptions de module d'analyse. Les descriptions de modules d'analyse qui référencent les types ou les éléments importés des fichiers de schémas définis par l'ONVIF peuvent ne pas répertorier de manière explicite ces fichiers.

DEMANDE:

- **ConfigurationToken [tt:ReferenceToken]**

Jeton de la configuration d'analyse pour laquelle il convient de répertorier les modules pris en charge.

RÉPONSE:

- **SupportedAnalyticsModules[tt:SupportedModules]**

Modules d'analyse pris en charge pour la configuration d'analyse

DÉFAUTS:

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

La configuration d'analyse demandée n'existe pas.

CLASSE D'ACCÈS:

READ_MEDIA

9.4.4.3 GetAnalyticsModules

L'opération suivante extrait les modules d'analyse actuellement installés.

DEMANDE:

- **ConfigurationToken [tt:ReferenceToken]**

Jeton de la configuration d'analyse pour laquelle il convient de répertorier les modules.

RÉPONSE:

- **AnalyticsModule – facultatif, non limité [tt:Config]**

Liste de modules installés pour la configuration spécifiée.

DÉFAUTS:

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

La configuration d'analyse demandée n'existe pas.

CLASSE D'ACCÈS:

READ_MEDIA

9.4.4.4 CreateAnalyticsModules

L'opération suivante ajoute des modules d'analyse à une AnalyticsConfiguration. Si les modules ne peuvent pas tous être créés conformément à la demande, le dispositif répond par un message de défaut.

DEMANDE:

- **ConfigurationToken [tt:ReferenceToken]**
Jeton de la configuration d'analyse pour laquelle il convient d'installer les modules.
- **AnalyticsModule – non limité [tt:Config]**
Liste de modules à installer pour la configuration spécifiée.

RÉPONSE:

Ceci est un message vide.

DÉFAUTS:

- **env:Sender – ter:InvalidArgVal – ter:NoConfig**
La configuration d'analyse demandée n'existe pas.
- **env:Sender – ter:InvalidArgVal – ter:InvalidModule**
La configuration de module proposée n'est pas valide sur le dispositif
- **env:Sender – ter:InvalidArgVal – ter:NameAlreadyExistent**
Le même nom de module d'analyse existe déjà dans la configuration.
- **env:Receiver – ter:Action – ter:TooManyModules**
Ressources insuffisantes dans le dispositif pour ajouter les modules d'analyse à la configuration.
- **env:Receiver – ter:Action – ter:ConfigurationConflict**
Le dispositif ne peut pas créer les modules d'analyse sans générer une configuration conflictuelle.

CLASSE D'ACCÈS:

ACTUATE

9.4.4.5 ModifyAnalyticsModules

L'opération suivante modifie plusieurs modules d'analyse. Si tous les modules d'analyse ne peuvent pas être modifiés conformément à la demande, le dispositif répond par un message de défaut.

DEMANDE:

- **ConfigurationToken [tt:ReferenceToken]**
Jeton de la configuration d'analyse pour laquelle il convient de modifier les modules.
- **AnalyticsModule – non limité [tt:Config]**
Liste de modules à modifier pour la configuration spécifiée.

RÉPONSE:

Ceci est un message vide.

DÉFAUTS:

- **env:Sender – ter:InvalidArgVal – ter:NoConfig**
La configuration d'analyse demandée n'existe pas.
- **env:Sender – ter:InvalidArgVal – ter:InvalidModule**
La configuration de module proposée n'est pas valide sur le dispositif
- **env:Sender – ter:InvalidArgVal – ter:NameNotExistent**
Le ou les noms de modules n'existent pas.
- **env:Receiver – ter:Action – ter:ConfigurationConflict**
Le dispositif ne peut pas modifier les modules sans créer une configuration conflictuelle.

CLASSE D'ACCÈS:

ACTUATE**9.4.4.6 DeleteAnalyticsModules**

L'opération suivante supprime plusieurs modules d'analyse. Si tous les modules d'analyse ne peuvent pas être supprimés conformément à la demande, le dispositif répond par un message de défaut.

DEMANDE:

- **ConfigurationToken [tt:ReferenceToken]**
Jeton de la configuration d'analyse pour laquelle il convient de supprimer les modules.
- **AnalyticsModuleName – non limité [xs:string]**
Liste de modules à supprimer pour la configuration spécifiée.

RÉPONSE:

Ceci est un message vide.

DÉFAUTS:

- **env:Sender – ter:InvalidArgVal – ter:NoConfig**
La configuration d'analyse demandée n'existe pas.
- **env:Sender – ter:InvalidArgVal – ter:InvalidModule**
La configuration de module proposée n'est pas valide sur le dispositif
- **env:Sender – ter:InvalidArgVal – ter:NameNotExistent**
Le ou les noms de modules n'existent pas.
- **env:Sender – ter:Action – ter:FixedModules**
Au moins un des modules demandés est fixé et ne peut pas être supprimé.
- **env:Receiver – ter:Action – ter:ConfigurationConflict**
Le dispositif ne peut pas supprimer les modules sans créer une configuration conflictuelle.

9.5 GetAnalyticsModuleOptions

L'opération suivante retourne les options applicables aux modules d'analyse pris en charge qui spécifient un attribut Option.

DEMANDE:

- **AnalyticsModuleType – facultatif [xs:QName]**
Type de module facultatif pour lequel les options doivent être extraites. S'il est omis, tous les types d'options de modules comportant toutes les options de modules prises en charge par le dispositif doivent être retournés (le cas échéant).
- **ConfigurationToken [tt:ReferenceToken]**
Jeton de la configuration d'analyse pour laquelle il convient de répertorier les modules pris en charge.

RÉPONSE:

- **Options – facultatif, non limité [tan:AnalyticsModuleConfigOptions]**
Liste d'options pour le module d'analyse spécifié.

DÉFAUTS:

- **env:Sender – ter:InvalidArgVal – ter:NoConfig**
La configuration d'analyse demandée n'existe pas.
- **env:Sender – ter:InvalidArgVal – ter:TypeNotExist**
Le type d'option de module n'existe pas.

CLASSE D'ACCÈS:

READ_MEDIA

9.6 Fonctionnalités

Les fonctionnalités reflètent les fonctions facultatives et la fonctionnalité d'un service. Les informations sont statiques et ne varient pas pendant le fonctionnement du dispositif. Les fonctionnalités suivantes sont disponibles:

RuleSupport:	Indique que le dispositif prend en charge l'interface de règles et la syntaxe de règles telles que spécifiées en 9.3.4.
AnalyticsModuleSupport:	Indique que le dispositif prend en charge l'interface de module d'analyse de scène telle que spécifiée en 9.4.
CellBasedSceneDescriptionSupported:	Indique que le dispositif génère la description de scène basée sur les cellules.
RuleOptionsSupported:	Indique que le dispositif prend en charge l'opération GetRuleOptions sur l'interface de règles.
AnalyticsModuleOptionsSupported:	Indique que le dispositif prend en charge l'opération GetAnalyticsModuleOptions sur l'interface de modules d'analyse.

DEMANDE:

Ce message est vide.

RÉPONSE:

- **Capabilities [tan:Capabilities]**

Ensemble d'indicateurs pour les groupes de fonctions tels que décrits ci-dessus.

DÉFAUTS:

Aucun

CLASSE D'ACCÈS:**PRE_AUTH****9.7 Événements – Détection audio (Audio Detected)**

Le moteur d'analyse peut envoyer l'événement suivant pour les données audio analysées.

```
<tt:MessageDescription IsProperty="false">
  <tt:Source>
    <tt:SimpleItemDescription Name="AudioSourceConfigurationToken"
      Type="tt:ReferenceToken"/>
    <tt:SimpleItemDescription Name="AudioAnalyticsConfigurationToken"
      Type="tt:ReferenceToken"/>
    <tt:SimpleItemDescription Name="Rule"
      Type="xs:string"/>
  </tt:Source>
  <tt>Data>
    <tt:SimpleItemDescription Name="isSoundDetected" Type="xs:boolean"/>
    <tt:SimpleItemDescription Name="UTCTime" Type="xs:dateTime"/>
    <tt:ElementItemDescription Name="AudioClassifications"
      Type="AudioClassDescriptor"/>
  </tt>Data>
  <tt:ParentTopic>tns1:AudioAnalytics/Audio/DetectedSound
  </tt:ParentTopic>
</tt:MessageDescription>
```

Le champ Rule (Règle) est facultatif et contient le nom d'instance Rule qui a produit l'événement de détection audio.

10 Transmission en continu en temps réel**10.1 Généralités**

Le présent article décrit la transmission en continu en temps réel de vidéo, d'audio et de métadonnées. Il n'existe pas de service spécifique associé à la transmission en continu en temps réel. Les configurations en temps réel au moyen des commandes de service Web sont définies dans le service multimédia et le ReceiverService.

10.2 Protocole de flux multimédia**10.2.1 Format de transport****10.2.1.1 Généralités**

Le protocole RTP (Real-Time Transport) est un protocole de transfert multimédia (voir 10.2.1.6). Les paragraphes 10.2.1.2 à 10.2.1.6 décrivent le transfert de données RTP.

10.2.1.2 Transfert de données RTP par le protocole UDP

Le protocole UDP présente la plus faible charge supplémentaire et est capable de transférer des données en temps réel d'une manière efficace. Un dispositif doit prendre en charge le protocole RTP/UDP, et il convient qu'il prenne en charge la multidiffusion RTP/UDP.

10.2.1.3 RTP/TCP

Ce mode facultatif est déconseillé en raison des ambiguïtés d'interprétation des protocoles RFC respectifs. Le protocole RTP/TCP est défini dans les normes RFC 4571 et RFC 4572.

10.2.1.4 RTP/RTSP/TCP

Il convient que le dispositif prenne en charge le transfert multimédia en utilisant le protocole RTP/RTSP pour traverser un pare-feu à l'aide d'un tunnel RTSP. Ce protocole doit être conforme à 10.12 de la norme RFC 2326.

10.2.1.5 RTP/RTSP/HTTP/TCP

Le flux de données doit être envoyé par le protocole HTTP pour traverser un pare-feu. Un dispositif doit prendre en charge le transfert multimédia en utilisant le protocole RTP/RTSP/HTTP/TCP. De plus, si un dispositif prend en charge TLS, le flux de données doit être envoyé ou reçu par le protocole HTTPS pour traverser un pare-feu, et un dispositif doit prendre en charge le transfert multimédia à l'aide du protocole RTP/RTSP/HTTPS/TCP.

Ce protocole doit être conforme à 10.12 de la norme RFC 2326 (RTSP: Embedded [Interleaved] Binary Data).

Cette méthode de tunnellation doit également être conforme au QuickTime disponible auprès d'Apple Inc⁷. Les parties obligatoires des protocoles RTSP et RTP d'Apple Tunneling QuickTime par le protocole HTTP doivent être mises en œuvre par un dispositif.

10.2.1.6 RTP/RTSP/TCP/WebSocket

Le dispositif qui indique la prise en charge de RTSPOverWebSocket tel qu'expliqué en 6.12 doit prendre en charge la transmission en continu multimédia à l'aide du protocole WebSocket conformément à la présente spécification. Un dispositif doit utiliser le même port du protocole HTTP pour le protocole WebSocket dans le cas d'une connexion normale. Si un dispositif prend en charge le protocole TLS, il doit alors prendre en charge la tunnellation des connexions WebSocket par TLS (voir RFC 2818) et le numéro du port doit être identique au numéro du port HTTPS. Pour la tunnellation du protocole RTSP par WebSocket, un dispositif doit prendre en charge les données binaires entrelacées RTP/RTSP/TCP telles que définies en 10.12 de la norme RFC 2326.

La mise en œuvre du protocole WebSocket doit être conforme à la norme RFC 6455.

Le mécanisme d'établissement d'une connexion WebSocket entre un client et un serveur est expliqué de manière détaillée à l'Article 7.

⁷ QuickTime est l'appellation commerciale d'un produit distribué par Apple Inc. Cette information est donnée à l'intention des utilisateurs du présent document et ne signifie nullement que l'IEC approuve ou recommande l'emploi exclusif du produit ainsi désigné. Des produits équivalents peuvent être utilisés s'il est démontré qu'ils conduisent aux mêmes résultats.

10.2.2 Transport multimédia

10.2.2.1 RTP

10.2.2.1.1 Généralités

Le protocole RTP permet le transfert en temps réel des flux multimédias entre deux points terminaux. Le protocole RTP prend en charge le reclassement, la stabilisation et la synchronisation multimédia.

Tous les flux multimédias transférés par le protocole RTP doivent être conformes aux normes RFC 3550, RFC 3551, RFC 3984, RFC 7798, RFC 3016 ou RFC 3640, et JPEG par le protocole RTP (voir 10.2.3). La Figure 10 présente l'en-tête de protocole RTP.

0										1										2										3									
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1								
V		P	X	CC				M	PT					numéro de séquence																									
horodatage																																							
identifiant de source de synchronisation (SSRC)																																							

Figure 10 – En-tête RTP

Un en-tête RTP doit être renseigné avec les valeurs du Tableau 4.

Tableau 4 – Valeur d'en-tête RTP

Champ d'en-tête	Valeur	Description
Version (V): 2 bits	2	
Padding (P) (Remplissage): 1 bit	0/1	Si la charge utile comprend un octet de remplissage, il convient de lui attribuer la valeur "1"
Extension (X): 1 bit	0/1	Si l'extension d'en-tête est utilisée, une valeur doit être attribuée au bit Extension. Par exemple, 10.2.3 JPEG sur RTP utilise cette extension.
CSRC count (CC) (Comptage CSRC): 4 bits	0	
Marker (M) (Marqueur): 1 bit	0/1	L'utilisation doit être conforme aux normes RFC correspondantes (RFC 3984 pour H.264 Video, par exemple), au présent document, "JPEG sur RTP", par exemple (voir 10.2.3) ou à la transmission RTP en continu de métadonnées (voir 10.2.2.1.2).
Payload type (PT) (Type de charge utile): 7 bits	Voir Article 6 de la norme RFC 3551.	
Sequence Number (Numéro de séquence): 16 bits		Il convient que la valeur initiale du "numéro de séquence" soit aléatoire (imprévisible) afin de rendre plus difficiles les attaques à texte clair connu sur le chiffrement. Ce numéro est incrémenté d'une unité pour chaque paquet de données RTP envoyé.
Timestamp (horodatage): 32 bits		Il convient que la valeur initiale d' "horodatage" soit aléatoire (imprévisible) afin de rendre plus difficiles les attaques à texte clair connu sur le chiffrement. Voir 10.2.2.2.2 pour des informations détaillées complémentaires sur la synchronisation multimédia. L'utilisation de l'horodatage dépend du codec.
SSRC 32 bits		Source de synchronisation pour le flux de données. La présente spécification n'impose aucune restriction sur l'utilisation de ce champ.

10.2.2.1.2 RTP pour flux de métadonnées

Les flux de métadonnées sont également transportés par RTP. L'utilisation du type de charge utile, du marqueur et de l'horodatage pour l'en-tête RTP du flux de métadonnées est définie de la manière suivante:

- Un type de charge utile dynamique (96-127) doit être utilisé pour le type de charge utile attribué dans le processus d'une configuration de session RTSP.
- La valeur "1" doit être attribuée au bit marqueur RTP lorsque le document XML est fermé.

- Il est recommandé d'utiliser un horodatage RTP représentant le temps de création du paquet RTP avec une fréquence d'horloge RTP de 90 000 Hz. Seuls des horodatages TUC doivent être utilisés dans le flux de métadonnées. La synchronisation de flux de données vidéo et audio est assurée à l'aide du protocole RTCP.

La charge utile de métadonnées est un document XML avec le nœud racine `tt:MetaDataStream`. La taille du document XML n'est pas limitée. Si une compression GZIP est utilisée, la charge utile commence par un en-tête GZIP conformément à la norme RFC 1952, suivi des données compressées. Un bit marqueur indique la fin des données compressées. Lorsqu'un point de synchronisation (voir "Points de synchronisation" de la spécification de service multimédia de l'ONVIF) est demandé pour le flux, le document XML précédent doit être fermé et un nouveau document initié. Il est recommandé d'initier de nouveaux documents XML au bout de 1 s, au plus tard. L'horodatage RTP du flux de métadonnées n'a pas de signification spécifique. Le flux de métadonnées multiplexe les métadonnées provenant de différentes sources. La présente spécification définit des paramètres fictifs pour la description de scène de l'analyse vidéo, le statut PTZ du contrôleur PTZ et les notifications de la configuration d'événement. Un dispositif peut choisir laquelle de ces parties il convient de multiplexer dans les métadonnées pendant la configuration multimédia (voir "Configuration de métadonnées" de la spécification de service multimédia de l'ONVIF). Chaque partie peut apparaître plusieurs fois dans un ordre aléatoire dans le document. Une connexion de métadonnées peut être bidirectionnelle en utilisant le mécanisme de voie de retour (voir 10.4).

Un flux de métadonnées contient les éléments suivants:

- VideoAnalyticsStream;
- PTZStream;
- EventStream.

Les paramètres fictifs des différentes sources de métadonnées présentent la structure XML suivante:

```
<xs:complexType name="VideoAnalyticsStream">
  <xs:choice minOccurs="0" maxOccurs="unbounded">
    <xs:element name="Frame" type="tt:Frame" />
    ...
  </xs:choice>
</xs:complexType>

<xs:complexType name="PTZStream">
  <xs:choice minOccurs="0" maxOccurs="unbounded">
    <xs:element name="PTZStatus" />
    ...
  </xs:choice>
</xs:complexType>

<xs:complexType name="EventStream">
  <xs:choice minOccurs="0" maxOccurs="unbounded">
    <xs:element ref="wsnt:NotificationMessage" />
    ...
  </xs:choice>
</xs:complexType>
```

Voici un exemple de document XML de métadonnées:

```
<?xml version="1.0" encoding="UTF-8"?>
<tt:MetaDataStream xmlns:tt="http://www.onvif.org/ver10/schema">
  <tt:VideoAnalytics>
    <tt:Frame UtcTime="2008-10-10T12:24:57.321">
      ...
    </tt:Frame>
    <tt:Frame UtcTime="2008-10-10T12:24:57.621">
      ...
    </tt:Frame>
  </tt:VideoAnalytics>

```

```

</tt:VideoAnalytics>
</tt:MetadataStream>

```

```

<?xml version="1.0" encoding="UTF-8"?>
<tt:MetadataStream xmlns:tt="http://www.onvif.org/ver10/schema">
  <tt:Event>
    <wsnt:NotificationMessage>
      <wsnt:Message>
        <tt:Message UtcTime= "2008-10-10T12:24:57.628">
          ...
        </tt:Message>
      </wsnt:Message>
    </wsnt:NotificationMessage>
  </tt:Event>
</tt:MetadataStream>

```

10.2.2.2 RTCP

10.2.2.2.1 Généralités

Le protocole de contrôle RTCP produit un retour sur la qualité du service fourni par le protocole RTP et la synchronisation de différents flux multimédia. Le protocole RTCP doit être conforme à la norme RFC 3550. La Figure 11 présente le flux d'informations applicable à la fois aux rapports expéditeur (SR – *sender reports*) et aux rapports récepteur (RR – *receiver reports*).

Pour une demande de retour, il convient de prendre en charge les normes RFC 4585 et RFC 5104.

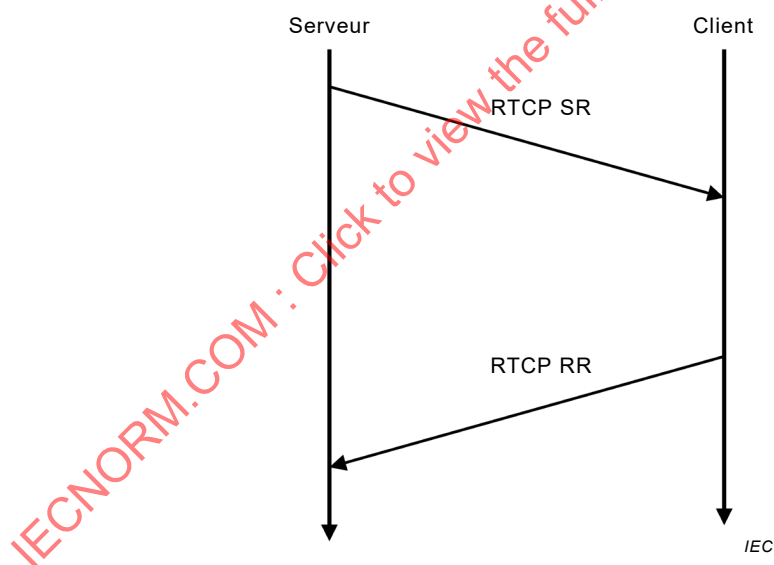


Figure 11 – Séquence RTCP

10.2.2.2.2 Synchronisation multimédia

Un client peut recevoir des flux audio et vidéo simultanément depuis plusieurs dispositifs. Dans ce cas, chaque flux utilise une horloge différente (de l'acquisition de données à la réception de paquets). Les rapports RTCP SR (Rapports expéditeur ou Sender Reports) permettent de synchroniser différents flux multimédias. Les RTCP SR doivent être conformes à la norme RFC 3550.

Le paquet RTCP SR contient des champs pour l'horodatage RTP et pour un horodatage en temps absolu (date et heure absolues, protocole NTP [NTP – *Network Time Protocol*] 64 bits). Voir Figure 12.

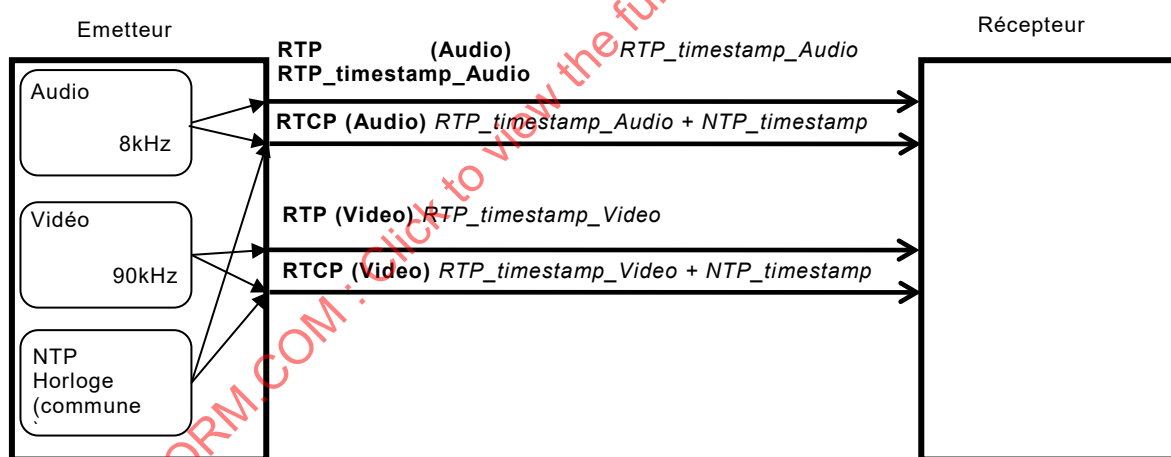
Un dispositif doit prendre en charge RTCP SR pour la synchronisation multimédia. Il convient que le client utilise le protocole RTCP pour la synchronisation multimédia.

0								1								2								3							
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
V		P		RC				PT=SR=200								Longueur															
SSRC de l'expéditeur																															
Horodatage NTP, mot le plus significatif																															
Horodatage NTP, mot le moins significatif																															
Horodatage RTP																															
Comptage de paquets de l'expéditeur																															
:																															
:																															

Figure 12 – Rapport expéditeur RTCP

Il convient que l'horloge absolue soit commune dans le dispositif et il convient de déterminer correctement chaque valeur d'horodatage. Le client peut synchroniser différents flux multimédias avec la temporisation appropriée basée sur les horodatages d'horloge RTP et d'horloge absolue (voir Figure 13).

En présence de plusieurs dispositifs, il convient que l'horodatage NTP soit commun à chacun d'eux et que le système exige le serveur NTP⁸.



IEC

Figure 13 – Synchronisation multimédia

10.2.3 Points de synchronisation

Les points de synchronisation permettent aux clients de décoder et d'utiliser correctement les données après le point de synchronisation. Un point de synchronisation peut être demandé par un client en cas d'erreur de décodeur (du fait de la perte de paquets, par exemple) pour forcer le dispositif à ajouter une trame I dès que possible ou pour demander le statut de PTZ ou d'événement actuel.

⁸ Le client peut obtenir des informations sur la "disponibilité de serveur NTP" à partir des dispositifs au moyen de la commande GetNTP comme défini dans l'IEC 60839-11-31.

Les méthodes basées sur Webservice exigent de prendre en charge la demande de point de synchronisation telle que définie dans la section "Point de synchronisation" de la spécification de service multimédia de l'ONVIF.

En outre, il est recommandé de prendre en charge les messages PLI décrits dans la norme RFC 4585 afin de permettre aux récepteurs tels que définis dans la spécification de service de réception de l'ONVIF de demander un point de synchronisation.

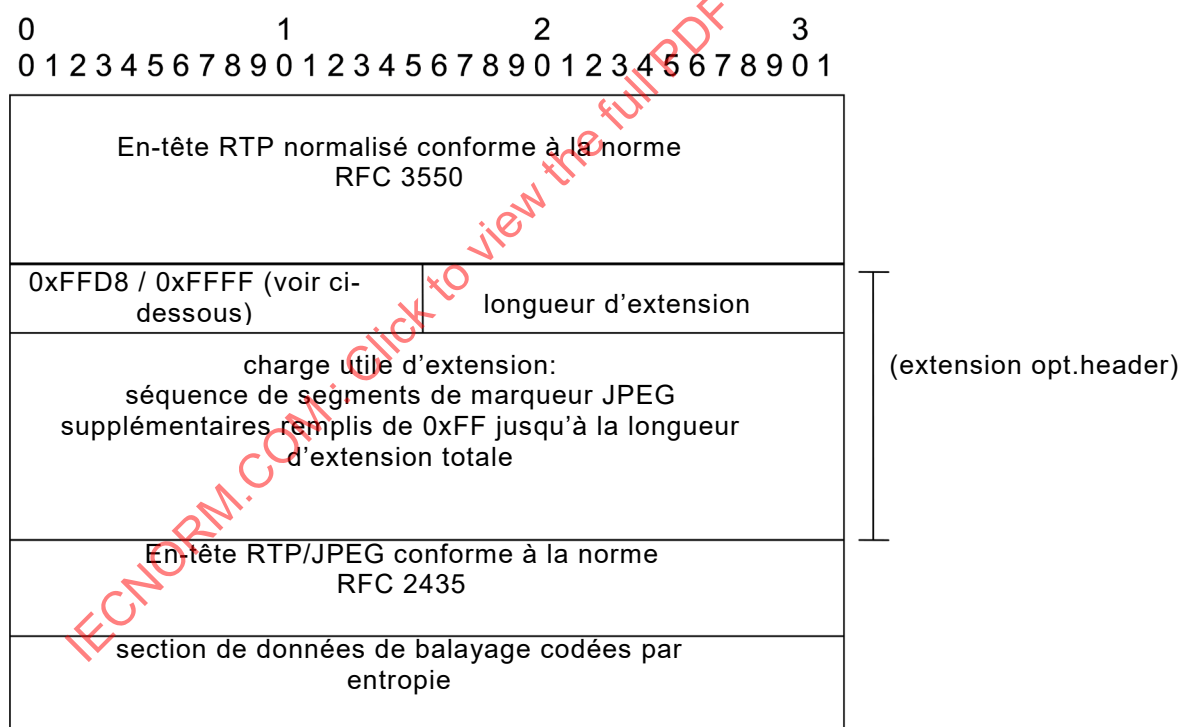
Pour les formats vidéo H.264 et H.265, l'en-tête SPS/PPS doit faire l'objet d'un envoi intrabande si ces messages ont été modifiés lors de la transmission.

10.2.4 JPEG par protocole RTP

10.2.4.1 Structure globale de paquet

La syntaxe de transmission des flux JPEG est conforme à la norme RFC 2435. Elle permet d'incorporer des données supplémentaires, au-delà des limitations de la norme RFC 2435, en utilisant une extension d'en-tête RTP facultative telle que spécifiée ci-dessous, avec certains des paquets RTP. Toutefois, cette option modifie la sémantique exacte des trames contenant ces paquets.

Le contenu type du format global d'un paquet RTP JPEG est présenté à la Figure 14.



IEC

Figure 14 – Structure de paquet RTP/JPEG

Afin de distinguer une extension d'en-tête RTP facultative des autres extensions d'en-tête possibles, les 16 premiers bits (les deux premiers octets de l'en-tête d'extension à quatre octets) d'un protocole RTP doivent avoir la valeur 0xFFD8 (marqueur SOI JPEG) pour le paquet initial et 0xFFFF pour les autres paquets RTP dans une trame.

Comme l'exige la norme RFC 3550, la présence de l'extension d'en-tête facultative doit être indiquée par le bit X de l'en-tête RTP. Le champ de longueur d'extension dans l'extension d'en-tête compte le nombre d'éléments à 32 bits suivants en tant que charges utiles d'extension. Par

exemple, un champ de longueur zéro suivant l'en-tête d'extension de 32 bits représente une extension d'en-tête vide.

La section de données de balayage codées par entropie peut ne pas être présente dans tous les paquets RTP. Toutefois, un en-tête RTP/JPEG complet doit être présent dans le paquet initial de chaque trame et de tous les paquets contenant une section de données de balayage codées par entropie, sinon il peut être absent.

Il convient d'utiliser le champ de décalage de fragment dans l'en-tête RTP/JPEG, conformément à la norme RFC 2435, comme si aucune extension d'en-tête n'était présente. De plus, si un paquet ne contient pas de segment de données de balayage codées par entropie, mais qu'il contient une extension d'en-tête, le champ de décalage de fragment ne doit pas être nul si des paquets contenant une section de données de balayage codées par entropie pour la même trame ont été transmis. Si le paquet initial d'une trame ne contient pas d'extension d'en-tête, selon le présent document, son champ de décalage de fragment doit être nul, sinon il convient qu'il le soit. Tous les paquets comprenant un en-tête RTP/JPEG avec un décalage de fragment nul et une valeur Q comprise entre 128 et 255 doivent contenir un entête de table de quantification selon 3.1.8 de la norme RFC 2435, et les autres paquets ne doivent pas contenir cet en-tête.

10.2.4.2 Spécification de décodage logique

Pour la spécification de décodage, l'ordre des paquets originaux dans le flux RTP a été par hypothèse restauré conformément à la numérotation de séquence RTP.

Si le paquet initial d'une trame ne contient pas d'extension d'en-tête RTP comme spécifié ci-dessus, les décodeurs doivent générer l'en-tête de balayage complet et effectuer le décodage comme spécifié par la norme RFC 2435. Les sections de données de balayage et les charges utiles d'une extension d'en-tête conforme à la présente spécification, jusqu'au paquet RTP suivant inclus avec son bit marqueur défini, doivent être concaténées au fur et à mesure de leur apparition dans le flux, en ignorant leurs valeurs de décalage de fragment.

À défaut (au moins une extension d'en-tête vide telle que spécifiée ci-dessus est présente dans le paquet initial d'une trame), les règles suivantes s'appliquent pour chaque trame de ce type:

- Si le paquet initial d'une trame ne contient pas de segment de données de balayage codées par entropie, mais contient une extension d'en-tête telle que spécifiée ci-dessus, les décodeurs doivent alors concaténer sa charge utile d'extension d'en-tête avec une ou des charges utiles d'extension d'en-tête (éventuellement vides ou inexistantes) conformes à la présente spécification des paquets suivants jusqu'au premier paquet inclus avec le bit marqueur RTP défini ou contenant un segment de données de balayage codées par entropie.
- La charge utile d'extension d'en-tête (séquence) RTP initiale concaténée doit être logiquement préfixée avec un marqueur SOI JPEG (0xFFD8).
- Si la valeur Q de l'en-tête de balayage RTP/JPEG dans le paquet initial d'une trame n'est pas nulle, les tables de quantification doivent être préinitialisées conformément aux règles de la norme RFC 2435. Si Q est égale à zéro, les tables de quantification doivent être copiées à partir de la trame précédente, permettant aux marqueurs DQT présents dans cette charge utile d'extension d'en-tête (séquence) initiale de les remplacer.
- S'il s'agit de la trame initiale d'une séquence, les tables de Huffman doivent être préinitialisées conformément à la norme RFC 2435. Les tables de Huffman de toutes les trames suivantes doivent être copiées à partir de la trame précédente, afin de pouvoir remplacer les trames par des marqueurs DHT dans la charge utile d'extension d'en-tête (séquence) initiale.
- Si la charge utile d'extension d'en-tête (séquence) RTP initiale ne fournit aucun marqueur DRI, mais que l'en-tête RTP/JPEG du paquet initial d'une trame contient un marqueur de redémarrage RTP/JPEG, un marqueur DRI correspondant aux règles de la norme RFC 2435 doit être ajouté à la charge utile d'extension d'en-tête (séquence) initiale. À défaut, si

l'extension d'en-tête (séquence) RTP initiale fournit un marqueur DRI, le marqueur doit être prioritaire sur tout autre marqueur de redémarrage RTP/JPEG conformément à la norme RFC 2435 pour la même trame. Toutefois, pour la compatibilité avec les décodeurs conformes à la norme RFC 2435 uniquement, il convient normalement que les codeurs utilisent un marqueur de redémarrage RTP/JPEG avec des valeurs cohérentes, si des intervalles de redémarrage doivent être utilisés.

- Les marqueurs DRI ne doivent pas être déduits des trames précédentes.
- Si la charge utile d'extension d'en-tête (séquence) RTP initiale ne fournit aucun marqueur SOF, qui sinon est prioritaire, un marqueur SOF doit lui être ajouté avec les valeurs suivantes:
 - si les champs de largeur et de hauteur de l'en-tête RTP/JPEG sont nuls, le marqueur SOF de la trame précédente doit être utilisé;
 - à défaut, il doit être déduit conformément aux règles de la norme RFC 2435.

Toutefois, tant que la taille d'image (arrondie par excès) relève de la plage spécifiée dans la norme RFC 2435, il convient que les codeurs spécifient une taille d'image dans l'en-tête RTP/JPEG cohérente avec les valeurs d'un en-tête SOF supplémentaire.

- Si la charge utile d'extension d'en-tête (séquence) initiale ne fournit aucun marqueur SOS, un marqueur correspondant doit être déduit conformément à la norme RFC 2435 et ajouté à celle-ci, sinon le marqueur SOS de l'extension est prioritaire.

Un marqueur SOS ne doit pas être déduit de trames précédentes.

Si le marqueur SOS est présent et qu'il n'est pas suivi par des données de balayage codées par entropie dans l'extension, il doit être le marqueur final dans la charge utile (séquence) d'extension initiale d'une trame. Le remplissage nécessaire avec des octets 0xFF ne doit pas suivre ce marqueur, mais peut le précéder.

- Les données de balayage codées par entropie restantes et les charges utiles d'extensions d'en-tête doivent être logiquement ajoutées dans le même ordre que celui de leur apparition dans le flux RTP jusqu'à la fin de la trame, comme indiqué par le bit marqueur RTP. Un marqueur EOI final doit également être ajouté s'il n'est pas encore présent dans la séquence logique de cette trame.

Pour chaque trame, la séquence obtenue jusqu'au premier marqueur EOI inclus (éventuellement ajouté) doit être un flux JPEG valide (éventuellement abrégé), générant une image complète issue du processus de décodage de cette trame. La définition des données qui suivent ce premier marqueur EOI pour chaque trame ne relève pas du domaine d'application de la présente spécification.

Il convient que les mises en oeuvre fournissent les en-têtes JPEG complets pour chaque trame. Ce principe s'applique particulièrement aux informations sur la largeur et la hauteur, ainsi qu'aux tables de quantification et de Huffman. En l'absence de ce type d'information importante pour chaque trame, la restitution et la transmission en continu en multidiffusion peuvent toutes deux pâtir d'une information d'en-tête JPEG incomplète.

10.2.4.3 Espaces colorimétriques et facteurs d'échantillonnage pris en charge

Il convient qu'un émetteur utilise uniquement l'échelle de gris et l'espace colorimétrique YCbCr. Un client doit prendre en charge à la fois l'échelle de gris et YCbCr.

Les facteurs d'échantillonnage pour YCbCr doivent correspondre aux valeurs prises en charge par la norme RFC 2435. Par exemple, un facteur d'échantillonnage de 4:2:0 (préférentiel) ou 4:2:2.

10.2.4.4 Traitement du format d'affichage des pixels

Le format d'affichage des pixels des fichiers JPEG peut être spécifié dans le marqueur JFIF. Si le format d'affichage des pixels est différent du facteur normalisé 1:1 et 1:2 conformément à la norme RFC 2435, il convient que ce marqueur soit transmis dans la charge utile d'extension d'en-tête (séquence) initiale de chaque trame pour spécifier le format d'affichage des pixels (pour des données entrelacées à base de champ).

10.2.4.5 Traitement de la vidéo entrelacée

La vidéo entrelacée est codée par deux champs indépendants et indiquée comme il est spécifié dans la norme RFC 2435 dans l'en-tête RTP/JPEG.

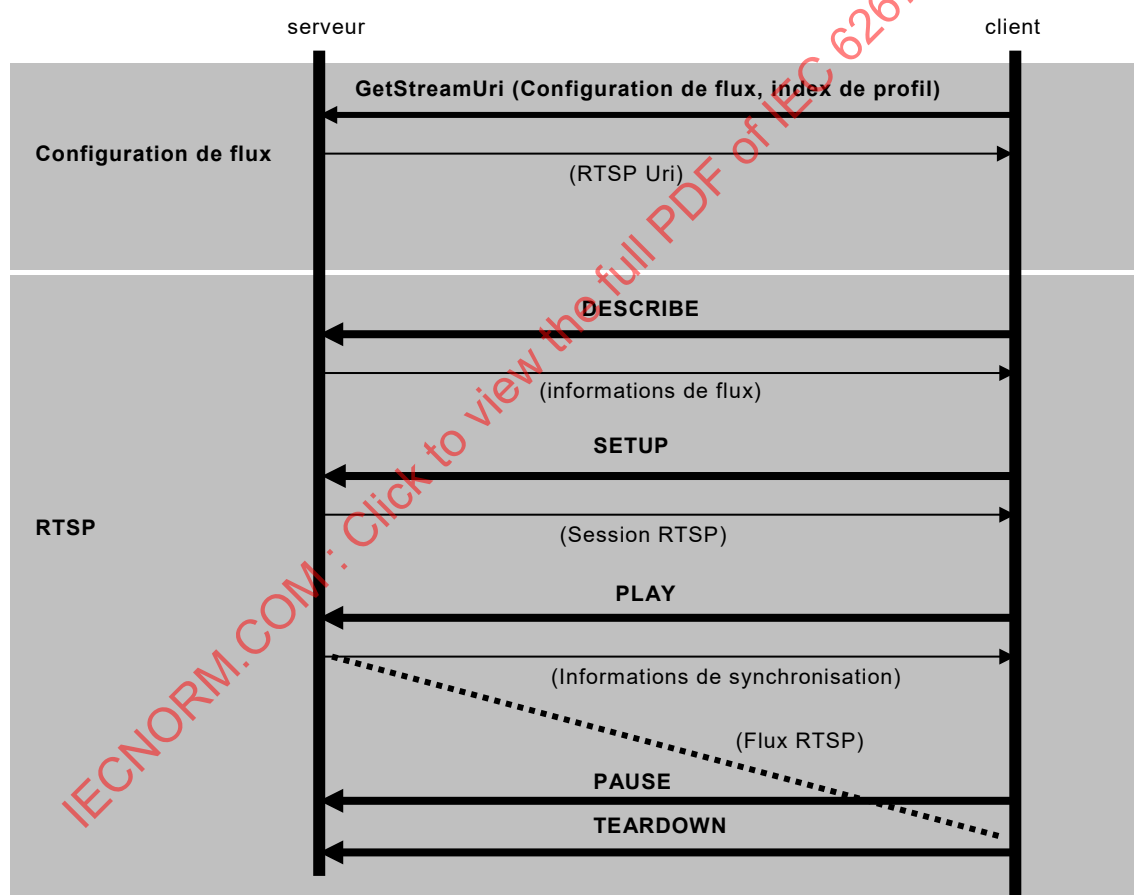
Les deux champs doivent utiliser les mêmes espace colorimétrique, facteurs d'échantillonnage et format d'affichage des pixels.

Il convient de ne pas utiliser le codage entrelacé si la trame a été initialement balayée progressivement.

10.3 Protocole de contrôle multimédia

10.3.1 Contrôle de flux RTSP

Le flux multimédia est contrôlé à l'aide du protocole défini dans l'URI comme présenté à la Figure 15. L'URI est retourné en réponse à la commande `GetStreamUri` définie dans la spécification de service multimédia de l'ONVIF.



IEC

Figure 15 – Contrôle de flux

Tous les dispositifs et clients doivent prendre en charge RTSP (RFC 2326) pour l'initiation de session et le contrôle de la restitution. RTSP doit utiliser TCP comme protocole de transport, le port TCP par défaut du trafic RTSP étant le port 554. Le protocole SDP (*Session Description Protocol*) doit être utilisé pour fournir des informations relatives au flux multimédia et SDP doit satisfaire à la norme RFC 4566. Voir le Tableau 5 pour les exigences concernant les méthodes individuelles.

Tableau 5 – Méthodes RTSP

Méthode	Sens	SPEC	Description
OPTIONS	R->T T->R	M X	Exigé pour obtenir une fonctionnalité de méthode facultative et autoriser différentes versions dans le futur.
DESCRIBE	R->T	M	Exigé pour extraire des paramètres multimédias dans le profil désigné.
ANNOUNCE	R->T T->R	X	
SETUP	R->T	M	Exigé pour définir des paramètres de session multimédia.
PLAY	R->T	M	Exigé pour démarrer un flux multimédia.
PAUSE	R->T	En direct: O Lecture: M	Exigé pour interrompre temporairement la restitution multimédia. Le traitement de flux multiples dans un réseau à bande passante étroite, par suspension du flux RTP, permet de contrôler correctement le trafic en réduisant les données redondantes et d'éviter les congestions du trafic réseau.
TEARDOWN	R->T	M	Exigé pour libérer une session multimédia.
GET_PARAMETER	R->T T->R	O	
SET_PARAMETER	R->T T->R	O O	Méthode facultative permettant de maintenir une session RTSP active (sens R->T uniquement).
REDIRECT	T->R	X	
RECORD	R->T	X	
Légende			
SPEC: La méthode est non prise en charge (X), obligatoire (M) ou facultative (O).			

Les dispositifs doivent prendre en charge le contrôle de flux cumulés, avec lequel les commandes play sont transmises à l'URI de contrôle de la session complète. Les dispositifs peuvent prendre en charge le contrôle de flux non cumulés, avec lequel les commandes play sont transmises séparément aux URI de contrôle de flux dans les sections multimédias du fichier SDP. Les fonctionnalités Media Streaming (Transmission en continu multimédia) permettent d'indiquer la prise en charge du contrôle de flux non cumulés.

10.3.2 Méthode de maintien "keep-alive " pour session RTSP

Un client RTSP maintient la session RTSP active ("keep-alive") et empêche l'expiration du délai d'attente de session (voir 12.37 de la norme RFC 2326). La présente spécification recommande les méthodes suivantes pour maintenir la session RTSP active pour la transmission en continu par diffusion individuelle ou multidiffusion.

- 1) Le client peut éventuellement définir le paramètre Timeout (Expiration du délai d'attente) (en secondes) en utilisant la commande `Set<configurationEntity>EncoderConfiguration` définie dans la spécification de service multimédia de l'ONVIF. Sinon, la valeur par défaut "60" est utilisée.
- 2) Dans toutes les réponses RTSP SETUP, il convient qu'un émetteur inclue la valeur de Timeout selon 12.37 de la norme RFC 2326, et qu'il utilise la valeur de Timeout pour le maintien "keep-alive".
- 3) Pour maintenir la session RTSP active, un client doit appeler le serveur RTSP à l'aide d'une méthode RTSP ou envoyer des rapports de récepteur RTCP. `SET_PARAMETER` est la méthode RTSP recommandée à utiliser comme présenté à la Figure 16.

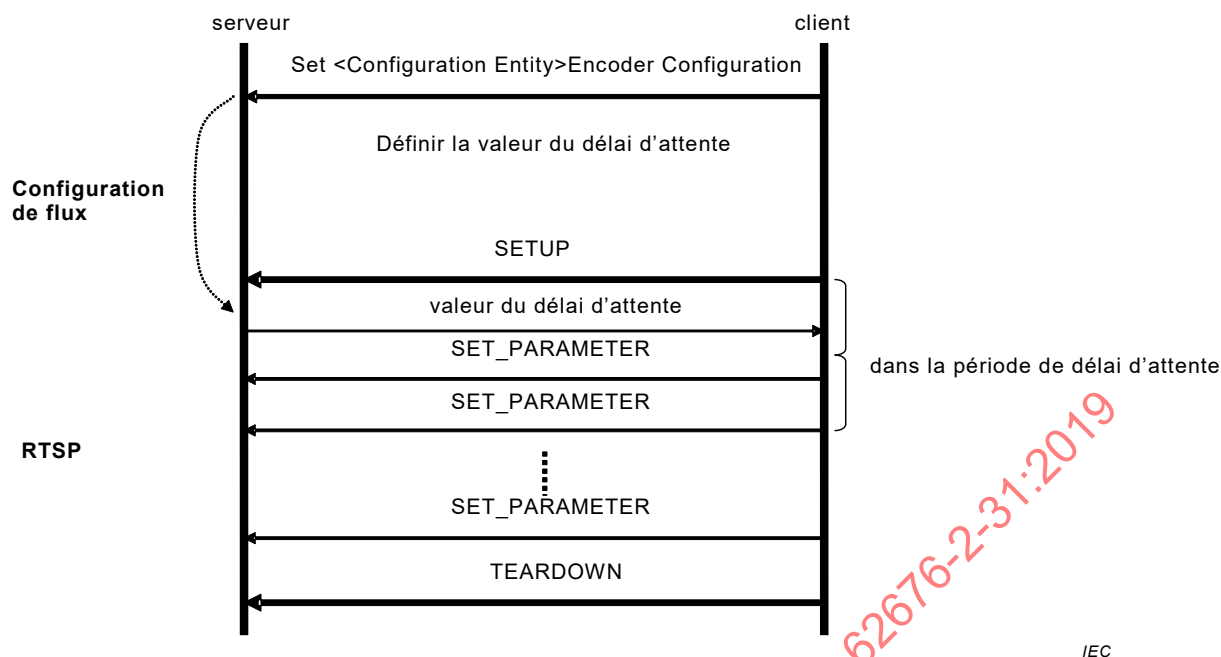


Figure 16 – Méthode Keep alive

10.3.3 Synchronisation audio et vidéo RTSP

Pour que les clients puissent immédiatement commencer la synchronisation des flux audio et vidéo, puis calculer les horodatages TUC absolus des paquets entrants à des fins d'enregistrement, il convient qu'un émetteur inclue les champs d'en-tête suivants dans la réponse RTSP PLAY:

- Range (12.29 de la norme RFC 2326). Cela doit inclure une heure de début en unités d'horloge (3.7 de la norme RFC 2326), mais pas en unités SMPTE ou NPT.
- RTP-Info (12.33 de la norme RFC 2326). Cela doit inclure une valeur rtptime correspondant à l'heure de début spécifiée dans l'en-tête Range.

Exemple:

```

client->>server: PLAY rtsp://example.com/onvif_camera/video RTSP/1.0
Cseq: 4
Range: npt=now-
Session: 12345678

server->>client: RTSP/1.0 200 OK
Cseq: 4
Session: 12345678
Range: clock=20100217T143720.257Z-
RTP-Info: url=rtsp://example.com/onvif_camera/video;
seq=1234;rtptime=3450012

```

10.3.4 Session RTSP pour un flux de métadonnées

Dans le cas d'un flux de métadonnées, la description SDP "application" doit être utilisée dans la réponse DESCRIBE pour le type de média et un de ces noms de codage doit être utilisé.

- "vnd.onvif.metadata" pour des données non compressées;
- "vnd.onvif.metadata.gzip" pour des données GZIP compressées;
- "vnd.onvif.metadata.exi.onvif" pour le format EXI utilisant des paramètres de compression par défaut ONVIF;

- "vnd.onvif.metadata.exi.ext" pour le format EXI utilisant des paramètres de compression qui font l'objet d'un envoi intrabande.

Exemple d'échange de message RTSP DESCRIBE entre un serveur RTSP (serveur) et un client RTSP (client):

```
client->server: DESCRIBE rtsp://example.com/onvif_camera RTSP/1.0
                  Cseq: 1

server->client: RTSP/1.0 200 OK
                  Cseq: 1
                  Content-Type: application/sdp
                  Content-Length: XXX
                  v=0
                  o=- 2890844256 2890842807 IN IP4 172.16.2.93
                  s=RTSP Session
                  m=audio 0 RTP/AVP 0
                  a=control:rtsp://example.com/onvif_camera /audio
                  m=video 0 RTP/AVP 26
                  a=control:rtsp://example.com/onvif_camera /video
                  m=application 0 RTP/AVP 107
                  a=control:rtsp://example.com/onvif_camera/metadata
                  a=recvonly
                  a=rtpmap
                  a=rtpmap:107 vnd.onvif.metadata/90000
```

10.3.5 Transmission en continu en multidiffusion

Un dispositif doit inclure une adresse de multidiffusion valide dans le champ "c=" d'une réponse DESCRIBE conformément à la norme RFC 4566.

Remarque: l'exception d'attribution d'adresse de multidiffusion dynamique facultative décrite en C.1.7 de la norme RFC 2326 autorisant des adresses 0.0.0.0 ne s'applique pas.

Le chapitre 10.7 (TEARDOWN) de la norme RFC 2326 indique qu'un dispositif doit interrompre le flux pour l'URI donné lors d'une opération de démontage. Cette disposition doit être clarifiée dans le cas d'une multidiffusion: le dispositif doit, pour un flux de multidiffusion, interrompre l'envoi de paquets pour une configuration de multidiffusion lorsqu'aucune session RTSP supplémentaire n'utilise la même configuration de multidiffusion, et que son drapeau AutoStart n'a pas été défini.

10.3.6 Exemple de message RTSP

Cet exemple présente le transfert de message entre un client RTSP (client) et un serveur RTSP (serveur). Le client demande un flux audio et vidéo depuis le dispositif. L'URI de flux "rtsp://example.com/onvif_camera" peut être extrait au moyen de la commande GetStreamUri. Se reporter à "Stream URI" de la spécification de service multimédia de l'ONVIF.

```
Client->server:      DESCRIBE rtsp://example.com/onvif_camera RTSP/1.0
                      Cseq: 1

server->client:      RTSP/1.0 200 OK
                      Cseq: 1
                      Content-Type: application/sdp
                      Content-Length: XXX
                      v=0
                      o=- 2890844256 2890842807 IN IP4 172.16.2.93
                      s=RTSP Session
                      m=audio 0 RTP/AVP 0
                      a=control:rtsp://example.com/onvif_camera/audio
                      m=video 0 RTP/AVP 26
                      a=control:rtsp://example.com/onvif_camera/video

client->server:      SETUP rtsp://example.com/onvif_camera/audio RTSP/1.0
                      Cseq: 2
```



```

Transport: RTP/AVP;unicast;client_port=8002-8003

server->client:      RTSP/1.0 200 OK
                    Cseq: 2
                    Transport: RTP/AVP;unicast;client_port=8002-8003;
                        server_port=9004-9005
                    Session: 12345678; timeout=60

client->server:      SETUP rtsp://example.com/onvif_camera/video RTSP/1.0
                    Cseq: 3
                    Transport: RTP/AVP;unicast;client_port=8004-8005
                    Session: 12345678

server->client:      RTSP/1.0 200 OK
                    Cseq: 3
                    Transport: RTP/AVP;unicast;client_port=8004-8005;
                        server_port=9006-9007
                    Session: 12345678; timeout=60

client->server:      PLAY rtsp://example.com/onvif_camera RTSP/1.0
                    Cseq: 4
                    Range: npt=now-
                    Session: 12345678

server->client:      RTSP/1.0 200 OK
                    Cseq: 4
                    Session: 12345678
RTP-Info: url=rtsp://example.com/onvif_camera/video; seq=1234;rtptime=3450012,
url=rtsp://example.com/onvif_camera/audio;
                    seq=22434;rtptime=1234566

client->server:      TEARDOWN rtsp://example.com/onvif_camera RTSP/1.0
                    Cseq: 5
                    Session: 12345678

server->client:      RTSP/1.0 200 OK
                    Cseq: 5
                    Session: 12345678

```

10.3.7 RTSP par le protocole HTTP

Le protocole RTSP par le protocole HTTP/HTTPS doit être pris en charge pour traverser un pare-feu. Voir 10.2.1.5 RTP/RTSP/HTTP/TCP.

10.4 Connexion de voie de retour

10.4.1 Généralités

Le Paragraphe 10.4 explique comment une connexion bidirectionnelle peut être établie entre un client et un serveur. Le traitement de la connexion de voie de retour est assuré à l'aide du protocole RTSP (voir norme RFC 2326). Un mécanisme est donc présenté, indiquant qu'un client souhaite établir une connexion de voie de retour. Le protocole RTSP offre des balises de caractéristiques permettant de gérer ce type d'ajouts de fonctionnalités.

Un dispositif qui prend en charge les connexions bidirectionnelles (les connexions audio ou de métadonnées, par exemple) doit prendre en charge les extensions RTSP présentées.

10.4.2 Balise RTSP Require

La norme RTSP [RFC 2326] peut être étendue à l'aide d'objets d'en-tête supplémentaires. Une balise Require est de ce fait introduite pour gérer les ajouts de fonctionnalités particuliers (voir 1.5 [Extending RTSP] et 12.32 [Require] de la norme RFC 2326).

La balise Require permet de déterminer la prise en charge de cette caractéristique. Cet en-tête doit être inclus dans toute demande qui exige du serveur qu'il comprenne cette caractéristique pour exécuter correctement la demande.

Un dispositif qui prend en charge la voie de retour et indique la prise en charge de la sortie audio par la fonctionnalité AudioOutputs doit comprendre la balise de voie de retour:

- www.onvif.org/ver20/backchannel

Un client RTSP qui souhaite établir une connexion RTSP avec une voie de retour de données doit inclure l'en-tête Require dans ses demandes.

10.4.3 Configuration de connexion pour une connexion bidirectionnelle

Un client doit inclure la balise de caractéristique dans sa demande DESCRIBE pour indiquer qu'une connexion de données bidirectionnelle doit être établie.

Un serveur qui comprend cette balise Require doit inclure un flux multimédia supplémentaire dans son fichier SDP tel que configuré dans son profil multimédia.

Un serveur RTSP qui ne comprend pas la balise de caractéristiques de voie de retour ou qui ne prend pas en charge les connexions de données bidirectionnelles doit répondre par un code d'erreur "551 Option not supported" (Option non prise en charge) conformément à la norme RTSP. Le client peut ensuite tenter d'établir une connexion RTSP sans voie de retour.

Un fichier SDP est utilisé pour décrire la session. Le serveur doit inclure les attributs a=sendonly ou a=recvonly dans chaque section multimédia du fichier SDP pour indiquer le sens d'envoi des données multimédia.

Le serveur doit répertorier tous les codecs de décodage pris en charge en tant que section multimédia propre, et le client choisit celui à utiliser.

10.4.4 Exemple de description d'un serveur sans prise en charge de voie de retour:

```
Client - Server:    DESCRIBE rtsp://192.168.0.1 RTSP/1.0
                   Cseq: 1
                   User-Agent: ONVIF Rtsp client
                   Accept: application/sdp
                   Require: www.onvif.org/ver20/backchannel

Server - Client:    RTSP/1.0 551 Option not supported
                   Cseq: 1
                   Unsupported: www.onvif.org/ver20/backchannel
```

10.4.5 Exemple de description d'un serveur avec prise en charge de voie de retour ONVIF:

```
Client - Server:    DESCRIBE rtsp://192.168.0.1 RTSP/1.0
                   Cseq: 1
                   User-Agent: ONVIF Rtsp client
                   Accept: application/sdp
                   Require: www.onvif.org/ver20/backchannel

Server - Client:    RTSP/1.0 200 OK
                   Cseq: 1
                   Content-Type: application/sdp
                   Content-Length: xxx

                   v=0
                   o= 2890842807 IN IP4 192.168.0.1
                   s=RTSP Session with audiobackchannel
                   m=video 0 RTP/AVP 26
                   a=control:rtsp://192.168.0.1/video
```

```

a=recvonly
m=audio 0 RTP/AVP 0
a=control:rtsp://192.168.0.1/Audio
a=recvonly
m=audio 0 RTP/AVP 0
a=control:rtsp://192.168.0.1/audioback
a=rtpmap:0 PCMU/8000
a=sendonly

```

Ce fichier SDP décrit de manière exhaustive la session RTSP. Le serveur donne au client ses URL de contrôle pour configurer les flux.

À l'étape suivante, le client peut configurer les sessions:

```

Client - Server:    SETUP rtsp://192.168.0.1/video RTSP/1.0
                   Cseq: 2
                   Transport: RTP/AVP;unicast;client_port=4588-4589

Server - Client:    RTSP/1.0 200 OK
                   Cseq: 2
                   Session: 123124;timeout=60
                   Transport:RTP/AVP;unicast;client_port=4588-4589; server_port=6256-6257

Client - Server:    SETUP rtsp://192.168.0.1/audio RTSP/1.0
                   Cseq: 3
                   Session: 123124
                   Transport: RTP/AVP;unicast;client_port=4578-4579

Server - Client:    RTSP/1.0 200 OK
                   Cseq: 3
                   Session: 123124;timeout=60
                   Transport:RTP/AVP;unicast;client_port=4578-4579; server_port=6276-6277

Client - Server:    SETUP rtsp://192.168.0.1/audioback RTSP/1.0
                   Cseq: 4
                   Session: 123124
                   Transport: RTP/AVP;unicast;client_port=6296-6297
                   Require: www.onvif.org/ver20/backchannel

Server - Client:    RTSP/1.0 200 OK
                   Cseq: 4
                   Session: 123124;timeout=60
                   Transport:RTP/AVP;unicast;client_port=6296-6297; server_port=2346-2347

```

La troisième demande de configuration établit la connexion de voie de retour audio.

À l'étape suivante, le client démarre la session en envoyant une demande PLAY.

```

Client - Server:    PLAY rtsp://192.168.0.1 RTSP/1.0
                   Cseq: 5
                   Session: 123124
                   Require: www.onvif.org/ver20/backchannel

Server - Client:    RTSP/1.0 200 OK
                   Cseq: 5
                   Session: 123124;timeout=60

```

Après avoir reçu la réponse OK à la demande PLAY, le client peut commencer à envoyer les données audio au serveur. Il ne doit pas commencer l'envoi des données au serveur avant d'avoir reçu la réponse.

L'en-tête Require indique qu'une interprétation particulière de la commande PLAY est nécessaire. La commande couvre à la fois le début du flux audio et vidéo entre le NVT et le client, et le début de la connexion audio du client au serveur.

Pour mettre fin à la session, le client envoie une demande TEARDOWN.

```
Client - NVT:   TEARDOWN rtsp://192.168.0.1 RTSP/1.0
                Cseq: 6
                Session: 123124
                Require: www.onvif.org/ver20/backchannel

NVT - Client:   RTSP/1.0 200 OK
                Cseq: 6
                Session: 123124
```

10.4.6 Transmission en continu en multidiffusion

Si le client envisage d'envoyer ses données par multidiffusion, il doit utiliser le paramètre de transport de la demande SETUP pour indiquer au serveur l'adresse et le port de multidiffusion.

Exemple:

```
Client - Server:   SETUP rtsp://192.168.0.1/audioback RTSP/1.0
                  Cseq: 4
                  Session: 123124
                  Transport:RTP/AVP,multicast;destination=224.2.1.1;port=60000-60001;ttl=128
                  Require: www.onvif.org/ver20/backchannel

Server - Client:   RTSP/1.0 200 OK
                  Cseq: 4
                  Session: 123124;timeout=60
                  Transport:RTP/AVP,multicast;destination=224.2.1.1;
                        port=60000-60001;ttl=128;mode="PLAY"
```

10.5 Gestion des erreurs

Les erreurs de protocole RTSP et HTTP sont classées dans différentes catégories (par exemple, les codes de statut 1xx, 2xx, 3xx, 4xx et 5xx). Le dispositif et le client doivent prendre en charge et gérer ces codes de statut. Pour les définitions de code de statut RTSP, se reporter à 11.0 de la norme RFC 2326. Pour les définitions de code de statut HTTP, se reporter à 10.0 (HTTP/1.1) de la norme RFC 2616.

Annexe A (normative)

Échange XML effectif (EXI)

Le codage EXI permet une représentation plus compacte des métadonnées XML. Cette fonctionnalité est indiquée si le CompressionType retourné au moyen de la commande GetMetadataConfigurationOptions contient la mention "EXI". La configuration EXI définie par l'ONVIF (voir Tableau A.1 et Tableau A.2) doit être prise en charge par tous les dispositifs qui indiquent une prise en charge de la compression EXI. Un codage EXI basé sur un schéma doit être utilisé. Le schéma exigé peut être obtenu à partir de tout dispositif qui utilise la commande GetWsdUrl.

L'en-tête EXI doit être transmis uniquement si un réglage différent de la configuration définie par l'ONVIF est utilisé. À l'exception du réglage des deux éléments "Presence Bit" ("Bit de présence") et "EXI Options" ("Options EXI"), les réglages d'en-tête EXI définis par l'ONVIF (voir Tableau A.1) doivent être toujours utilisés.

Tableau A.1 – Réglages d'en-tête EXI définis par l'ONVIF

Élément d'en-tête EXI	Valeur
EXI Cookie (Cookie EXI)	obligatoire
Distinguishing Bits (Bits de distinction)	obligatoire
EXI Format Version (Version de format EXI)	0 0000
Presence Bit for EXI Options (Bit de présence pour les options EXI)	0
Exi Options (Options EXI)	Voir Tableau A.2.
Padding Bits (Bits de remplissage)	Lorsqu'ils existent, leur valeur doit être "0".

Tableau A.2 – Réglages de configuration EXI définis par l'ONVIF

Option EXI	Valeur
alignment	par défaut (bit-packed)
compression	par défaut (false)
strict	par défaut (false)
fragment	par défaut (false)
preserve	par défaut (all false)
selfContained	par défaut (false)
schemaID	Insérer ici la référence au schéma obtenu à partir du dispositif.
datatypeRepresentationMap	aucune
blockSize	par défaut (1 000 000)
valueMaxLength	par défaut (unbounded)
valuePartitionCapacity	par défaut (unbounded)
métadonnées définies par l'utilisateur	aucune

Annexe B (normative)

Description de la lentille

Les lentilles à grand angle produisent une image à déformation géométrique. La description de la lentille décrit le point d'impact du faisceau de lumière incidente sur le capteur d'image, comme représenté à la Figure B.1. Cette disposition permet aux clients de compenser la déformation. La présente spécification décrit uniquement la mise en correspondance des faisceaux de lumière incidente et de l'emplacement du capteur. La méthode réelle de compensation de la déformation peut être choisie librement par le client et ne relève pas de la présente spécification.

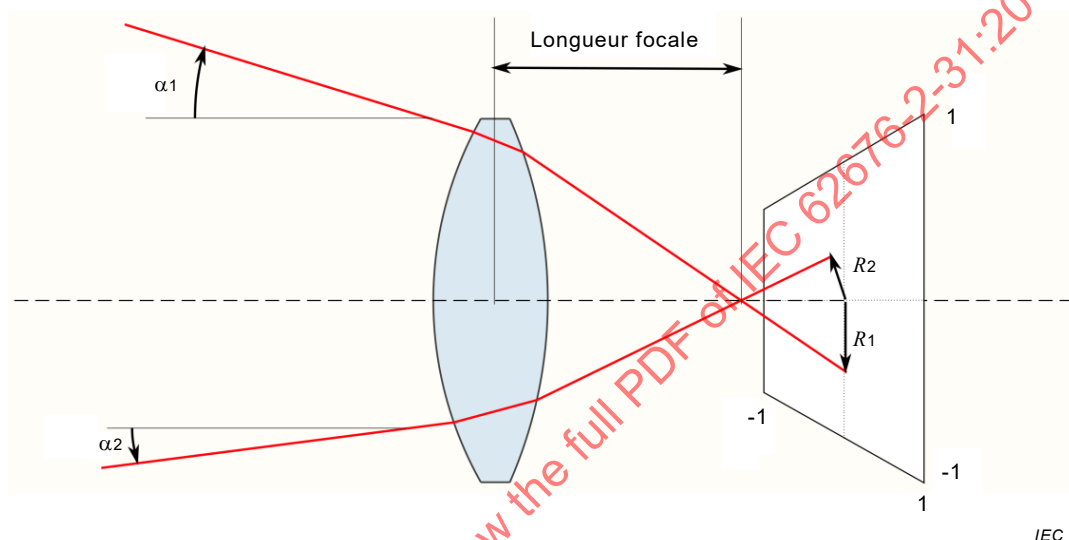


Figure B.1 – Mise en correspondance optique de l'angle (α) par le rayon (R) avec les coordonnées x/y normalisées

La fonction de projection de la lentille est définie par un ensemble de points. L'approximation B-spline présentée à la Figure B.2 permet de réaliser une mise en correspondance lisse.

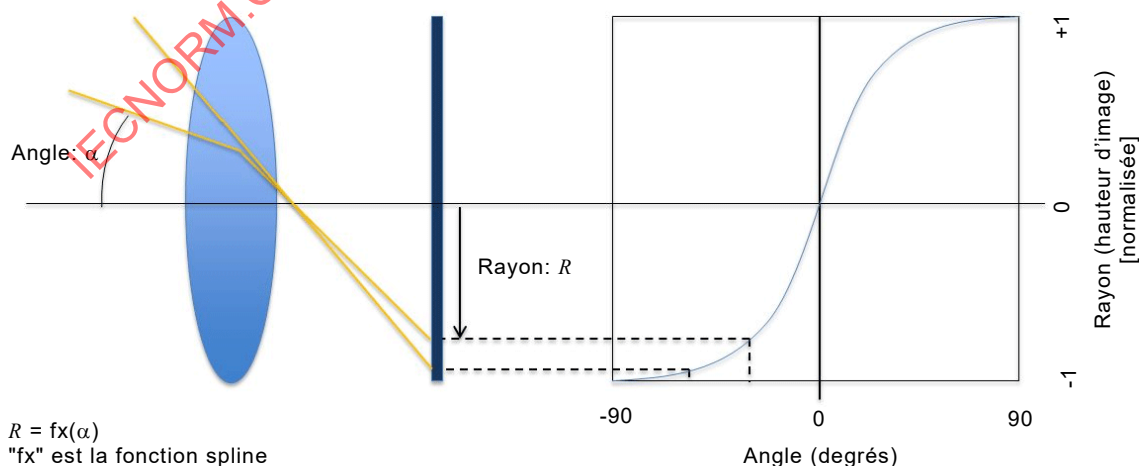
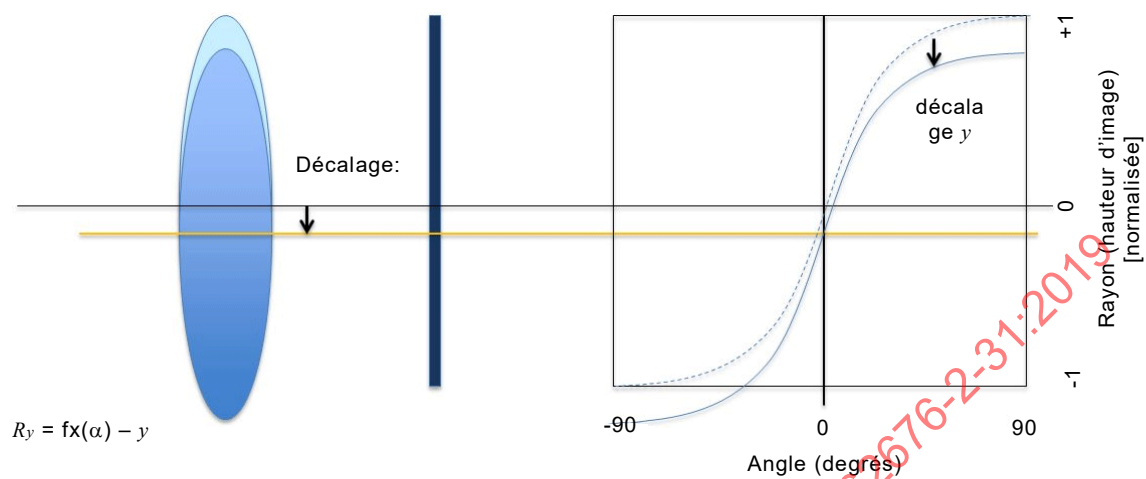


Figure B.2 – Mise en correspondance lisse avec approximations B-spline

La fonctionnalité dite LensOffset permet de décrire les décalages de l'axe entre l'imageur et la lentille. La Figure B.3 indique que la compensation d'un décalage de l'axe vertical entraîne un décalage de la courbe spline.



IEC

Figure B.3 – Compensation du décalage de l'axe vertical

Annexe C (informative)

Spécification de règles

C.1 Généralités

Les règles suivantes s'appliquent aux caméras statiques. Dans le cas d'un dispositif PTZ, il convient que les règles basées sur les images contiennent un ElementItem supplémentaire. L'élément ElementItem identifie la position du dispositif pour laquelle la règle a été définie. L'élément ElementItemDescription correspondant se présente comme suit:

```
<tt:ElementItemDescription Name="PTZStatus" Type="tt:PTZStatusType">
```

C.2 LineDetector

L'élément LineDetector est défini par une ligne brisée simple sans intersection. Si un objet coupe la ligne brisée dans la direction spécifiée, le moteur de règles envoie un événement Crossed (Coupé) contenant le nom de l'élément LineDetector et une référence à l'objet qui a coupé la ligne. Les directions Left (gauche), Right (droite) et Any (toute direction) peuvent être sélectionnées, les directions gauche et droite faisant référence à la direction de traversée de la ligne du premier point au deuxième point et constituant les directions interdites.

L'élément LineDetector s'apparente au code suivant utilisant le langage de description de règle, détaillé dans la section précédente:

```
<tt:RuleDescription Name="tt:LineDetector">
  <tt:Parameters>
    <tt:SimpleItemDescription Name="Direction" Type="tt:Direction"/>
    <tt:ElementItemDescription Name="Segments" Type="tt:Polyline"/>
  </tt:Parameters>
  <tt:Messages>
    <tt:Source>
      <tt:SimpleItemDescription Name="VideoSourceConfigurationToken"
        Type="tt:ReferenceToken"/>
      <tt:SimpleItemDescription Name="VideoAnalyticsConfigurationToken"
        Type="tt:ReferenceToken"/>
      <tt:SimpleItemDescription Name="Rule" Type="xs:string"/>
    </tt:Source>
    <tt>Data>
      <tt:SimpleItemDescription Name="ObjectId" Type="xs:integer"/>
    </tt>Data>
    <tt:ParentTopic>tns1:RuleEngine/LineDetector/Crossed</tt:ParentTopic>
  </tt:Messages>
</tt:RuleDescription>
```

Le code ci-dessus définit deux paramètres, Segments et Direction, et produit un événement lié à la rubrique tns1:RuleEngine/LineDetector/Crossed.

C.3 FieldDetector

Un élément FieldDetector est défini par un polygone simple sans intersection. L'élément FieldDetector détermine si chaque objet de la scène est à l'intérieur ou à l'extérieur du polygone. Ces informations sont placées dans une propriété.

L'élément FieldDetector s'apparente au code suivant, utilisant le langage de description de règle détaillé dans la section précédente:

```
<tt:RuleDescription Name="tt:FieldDetector">
  <tt:Parameters>
```



```

    <tt:ElementItemDescription Name="Field" Type="tt:Polygon"/>
  </tt:Parameters>
  <tt:Messages IsProperty="true">
    <tt:Source>
      <tt:SimpleItemDescription Name="VideoSourceConfigurationToken"
                               Type="tt:ReferenceToken"/>
      <tt:SimpleItemDescription Name="VideoAnalyticsConfigurationToken"
                               Type="tt:ReferenceToken"/>
      <tt:SimpleItemDescription Name="Rule" Type="xs:string"/>
    </tt:Source>
    <tt:Key>
      <tt:SimpleItemDescription Name="ObjectId" Type="xs:integer"/>
    </tt:Key>
    <tt>Data>
      <tt:SimpleItemDescription Name="IsInside" Type="xs:boolean"/>
    </tt>Data >
    <tt:ParentTopic>
      tns1:RuleEngine/FieldDetector/ObjectsInside
    </tt:ParentTopic>
  </tt:Messages>
</tt:RuleDescription>

```

À partir de la propriété Inside (A l'intérieur), un client peut déduire les paramètres Entering (Entrée) et Leaving (Sortie) du détecteur. Un client peut simuler des événements Entering et Leaving en ajoutant un filtre MessageContent à l'abonnement, qui laisse passer uniquement les messages ObjectsInside, la valeur true (vrai) ou false (faux) étant respectivement affectée à l'élément IsInside.

C.4 LoiteringDetector

Un élément LoiteringDetector est défini par un polygone simple sans intersection comme une zone d'intérêt et un intervalle d'errance seuil. L'élément LoiteringDetector détermine si le temps de séjour de chaque objet constitutif de la scène à l'intérieur du polygone est supérieur à la valeur seuil donnée. Il signale le moment déclencheur de l'errance de l'objet.

L'élément LoiteringDetector est défini par le code suivant en utilisant le langage de description de règles:

```

<tt:RuleDescription Name="tt:LoiteringDetector">
  <tt:Parameters>
    <tt:ElementItemDescription Name="Field"
                              Type="tt:PolygonConfiguration"/>
    <tt:SimpleItemDescription Name="TimeThreshold"
                              Type="xs:nonNegativeInteger"/>
    <tt:SimpleItemDescription Name="isPeriodicNotify" Type="xs:boolean"/>
    <tt:SimpleItemDescription Name="TimeInterval" Type="xs:duration"/>
  </tt:Parameters>
  <tt:Messages IsProperty="true">
    <tt:Source>
      <tt:SimpleItemDescription Name="VideoSourceConfigurationToken"
                               Type="tt:ReferenceToken"/>
      <tt:SimpleItemDescription Name="VideoAnalyticsConfigurationToken"
                               Type="tt:ReferenceToken"/>
      <tt:SimpleItemDescription Name="Rule" Type="xs:string"/>
    </tt:Source>
    <tt:Key>
      <tt:SimpleItemDescription Name="ObjectId" Type="xs:integer"/>
    </tt:Key>
    <tt>Data>
      <tt:SimpleItemDescription Name="Since" Type="xs:dateTime"/>
      <tt:SimpleItemDescription Name="TimeStamp" Type="xs:dateTime"/>
    </tt>Data >
    <tt:ParentTopic>
      tns1:RuleEngine/LoiteringDetector/ObjectIsLoitering
    </tt:ParentTopic>
  </tt:Messages>

```

```
</tt:RuleDescription>
```

La définition de règle ci-dessus précise qu'une instance de règle génère un événement associé à la rubrique `tns1:RuleEngine/LoiteringDetector/ObjectIsLoitering`. Les paramètres de configuration sont indiqués dans le Tableau C.1.

Tableau C.1 – Paramètres de configuration des règles de l'élément LoiteringDetector

Nom de paramètre	Description
Field (Champ)	Zone du champ de vision de la caméra permettant de détecter l'errance d'un objet
TimeThreshold	Seuil de durée maximal permettant de déterminer l'errance d'un objet
isPeriodicNotify	Production maintenue ou non d'événements périodiques relatifs à un objet errant
TimeInterval	Intervalle de temps périodique destiné à maintenir l'envoi d'événements d'errance relatifs à un objet errant

L'événement contient les champs indiqués dans le Tableau C.2.

Tableau C.2 – Description des champs d'un événement d'errance

Nom de paramètre	Description
ObjectId	Identifiant de l'objet
Since (Durée)	Durée d'errance de cet objet
TimeStamp	Horodatage du dernier échantillon d'objet qui déclenche la production d'un événement de règle

C.5 Détecteur de mouvement déclaratif

Le détecteur de mouvement déclaratif détecte le mouvement d'un objet par rapport aux attributs de mouvement spécifiés. La règle est active dans une zone (zone d'intérêt) et un intervalle de temps donnés. La condition est définie par rapport à l'élément de trame/objet du descripteur de scène.

```
<tt:RuleDescription Name="tt:DeclarativeMotionDetector">
  <tt:Parameters>
    <tt:ElementItemDescription Name="Field" Type="tt:PolygonConfiguration"/>
    <tt:ElementItemDescription Name="MotionExpression"
      Type="tt:MotionExpressionConfiguration"/>
  </tt:Parameters>
  <tt:Messages IsProperty="true">
    <tt:Source>
      <tt:SimpleItemDescription Name="VideoSourceConfigurationToken"
        Type="tt:ReferenceToken"/>
      <tt:SimpleItemDescription Name="VideoAnalyticsConfigurationToken"
        Type="tt:ReferenceToken"/>
      <tt:SimpleItemDescription Name="Rule" Type="xs:string"/>
    </tt:Source>
    <tt:Key>
      <tt:SimpleItemDescription Name="ObjectId" Type="xs:integer"/>
    </tt:Key>
    <tt>Data>
      <tt:SimpleItemDescription Name="TimeStamp" Type="xs:dateTime"/>
    </tt>Data>
    <tt:ParentTopic>
      tns1:RuleEngine/DeclarativeMotionDetector/MotionMatched
    </tt:ParentTopic>
  </tt:Messages>
</tt:RuleDescription>
```

La description de règle ci-dessus précise qu'une instance de règle génère un événement associé à la rubrique tns1:RuleEngine/DeclarativeMotionDetector/MotionMatched. Les paramètres de configuration sont indiqués dans le Tableau C.3.

Tableau C.3 – Paramètres de configuration des règles du détecteur de mouvement déclaratif

Nom de paramètre	Description
Field	Définit une zone du champ de vision de la caméra. La requête de mise en correspondance des mouvements est appliquée dans cette zone pour l'objet détecté.
MotionExpression	Expression de mouvement pour les éléments de trame/objet dans le schéma de descripteur de scène (Scene Descriptor)

L'événement contient les champs indiqués dans le Tableau C.4.

Tableau C.4 – Description des champs d'un événement de mouvement déclaratif

Nom de paramètre	Description
ObjectId	Identifiant de l'objet
TimeStamp	Horodatage du mouvement détecté

C.6 Règle de comptage

La règle de comptage compte le nombre d'objets en mouvement qui traversent l'ensemble de segments de lignes (barrière) et spécifie l'attribut de direction facultatif. Les paramètres de configuration incluent également l'intervalle de temps de consignation des événements, ainsi que l'intervalle de temps de réinitialisation du compteur.

```
<tt:RuleDescription Name="tt:CountAggregation">
  <tt:Parameters>
    <tt:ElementItemDescription Name="LineSegments"
      Type="tt:PolylineArrayConfiguration"/>
    <tt:SimpleItemDescription Name="ReportTimeInterval" Type="xs:duration"/>
    <tt:SimpleItemDescription Name="ResetTimeInterval" Type="xs:duration"/>
    <tt:SimpleItemDescription Name="Direction" Type="tt:Direction"/>
  </tt:Parameters>
  <tt:Messages IsProperty="true">
    <tt:Source>
      <tt:SimpleItemDescription Name="VideoSourceConfigurationToken"
        Type="tt:ReferenceToken"/>
      <tt:SimpleItemDescription Name="VideoAnalyticsConfigurationToken"
        Type="tt:ReferenceToken"/>
      <tt:SimpleItemDescription Name="Rule" Type="xs:string"/>
    </tt:Source>
    <tt:Key>
      <tt:SimpleItemDescription Name="ObjectId" Type="xs:integer"/>
    </tt:Key>
    <tt>Data>
      <tt:SimpleItemDescription Name="Count" Type="xs:nonNegativeInteger"/>
    </tt>Data>
    <tt:ParentTopic>
      tns1:RuleEngine/CountAggregation/Counter
    </tt:ParentTopic>
  </tt:Messages>
</tt:RuleDescription>
```

La description de règle ci-dessus précise qu'une instance de règle génère un événement de comptage de la rubrique tns1:RuleEngine/CountAggregation/Counter. Les paramètres de configuration sont indiqués dans le Tableau C.5.

Tableau C.5 – Paramètres de configuration de la règle de comptage

Nom de paramètre	Description
LineSegments	Réseau de lignes qui permet de détecter sa traversée par l'objet à des fins de comptage
ReportTimeInterval	Intervalle de temps destiné à la consignation des informations de comptage
ResetTimeInterval	Temps de réinitialisation de comptage périodique
Direction (Sens)	Sens de comptage

L'événement contient les champs indiqués dans le Tableau C.6.

Tableau C.6 – Description des champs d'un événement de comptage

Nom de paramètre	Description
ObjectId	Identifiant du dernier objet compté
Count (comptage)	Valeur du compteur

C.7 Règle de requête

La règle de requête permet à un client de définir une requête portant sur les métadonnées transmises en continu de flux multimédia. Le client s'abonne à la rubrique associée à la règle afin de recevoir les notifications du dispositif.

```
<tt:RuleDescription Name="tt:QueryRule">
  <tt:Parameters>
    <tt:ElementItemDescription Name="Query" Type="tt:MetadataFilter"/>
    <tt:SimpleItemDescription Name="VideoSource" Type="tt:ReferenceTokenList"/>
  </tt:Parameters>
  <tt:Messages>
    <tt:Source>
      <tt:SimpleItemDescription Name="VideoSourceConfigurationToken"
        Type="tt:ReferenceToken"/>
      <tt:SimpleItemDescription Name="VideoAnalyticsConfigurationToken"
        Type="tt:ReferenceToken"/>
      <tt:SimpleItemDescription Name="Rule" Type="xs:string"/>
    </tt:Source>
    <tt>Data>
      <tt:ElementItemDescription Name="Result" Type="xs:any"/>
    </tt>Data>
    <tt:ParentTopic>
      tns1:RuleEngine/QueryRule
    </tt:ParentTopic>
  </tt:Messages>
</tt:RuleDescription>
```

La rubrique est tns1:RuleEngine/QueryRule et la valeur de "Rule" ("Règle") dans l'événement de notification indique quelle instance de Query Rule a produit l'événement. Il convient que le client s'abonne à la rubrique avec le nom de règle correct afin de recevoir les événements de la règle de requête spécifique. Les paramètres de configuration sont indiqués dans le Tableau C.7.

Tableau C.7 – Paramètres de configuration de la règle de requête

Nom de paramètre	Description
Query (requête)	Contient la définition de requête
VideoSource	Sélectionne la ou les sources vidéo qu'exécutera la requête

Annexe D (informative)

Détection de mouvement de cellules

D.1 Détecteur de mouvement de cellules

Les règles du détecteur de mouvement de cellules traitent de la sortie du moteur d'analyse de mouvement de cellules. La règle de type "tt:CellMotionDetector" est composée de quatre paramètres: "MinCount", "AlarmOnDelay", "AlarmOffDelay" et "ActiveCells", répertoriés dans le Tableau D.1.

Le paramètre "MinCount" décrit le nombre minimal de cellules adjacentes qu'un objet en mouvement doit couvrir afin de produire un événement d'alarme.

Les paramètres AlarmOnDelay et AlarmOffDelay correspondent à des délais exprimés en millisecondes. Il convient que ces délais évitent les modifications d'état d'alarme dans un intervalle de temps court. Une alarme doit être active, pendant au moins la durée AlarmOnDelay afin de déclencher une alarme. Le paramètre AlarmOffDelay retarde la transition entre une alarme activée et une alarme inactive. Une alarme doit être inactive, pendant au moins la durée AlarmOffDelay afin d'être inactive une nouvelle fois.

Chacune des cellules définies par le type "tt:CellLayout" dans l'élément AnalyticsModule peut être activée ou désactivée de manière individuelle. Pour exclure de toute surveillance des zones spécifiques en raison de mouvements continus (par exemple arbre soumis aux effets du vent), les cellules correspondantes peuvent être désactivées. Un masque binaire est utilisé pour représenter les cellules activées et désactivées. Par exemple, lorsque 1 000 cellules sont disponibles, 1 000 bits doivent alors être définis. L'algorithme "Packbits" défini dans l'ISO 12369 (TIFF, Version 6.0), qui est une sorte de codeur à longueur de ligne, est utilisé pour un codage effectif de ce masque binaire. La valeur de sortie résultante est ensuite archivée sous la forme base64Binary.

La définition de règle est la suivante:

```
<tt:RuleDescription Name="tt:CellMotionDetector">
  <tt:Parameters>
    <tt:SimpleItemDescription Name="MinCount" Type="xs:integer"/>
    <tt:SimpleItemDescription Name="AlarmOnDelay" Type="xs:integer"/>
    <tt:SimpleItemDescription Name="AlarmOffDelay" Type="xs:integer"/>
    <tt:SimpleItemDescription Name="ActiveCells" Type="xs:base64Binary"/>
  </tt:Parameters>
  <tt:Messages IsProperty="true">
    <tt:Source>
      <tt:SimpleItemDescription
        Name="VideoSourceConfigurationToken"
        Type="tt:ReferenceToken"/>
      <tt:SimpleItemDescription
        Name="VideoAnalyticsConfigurationToken"
        Type="tt:ReferenceToken"/>
      <tt:SimpleItemDescription Name="Rule" Type="xs:string"/>
    </tt:Source>
    <tt>Data>
      <tt:SimpleItemDescription Name="IsMotion" Type="xs:boolean"/>
    </tt>Data>
    <tt:ParentTopic>
      tns1:RuleEngine/CellMotionDetector/Motion
    </tt:ParentTopic>
  </tt:Messages>
</tt:RuleDescription>
```

**Tableau D.1 – Paramètres de configuration des règles
du détecteur de mouvement de cellules**

Nom de paramètre	Description
MinCount	Comptage minimal des cellules activées adjacentes afin de déclencher le mouvement. Ce paramètre permet de supprimer les fausses alarmes.
AlarmOnDelay	Temps minimal en millisecondes nécessaire pour modifier l'état de l'alarme (inactif à actif). Ce délai est destiné à empêcher le déclenchement d'événements d'alarme très courts.
AlarmOffDelay	Temps minimal en millisecondes nécessaire pour modifier l'état de l'alarme (actif à inactif). Ce délai est destiné à empêcher de trop nombreuses modifications d'alarme.
ActiveCells	Une valeur "1" désigne une cellule active et une valeur "0" désigne une cellule inactive. La première cellule se situe dans l'angle gauche supérieur. Ensuite, l'ordre des cellules s'effectue tout d'abord de gauche à droite, puis de haut en bas (voir 9.2.3.7). Si le nombre de cellules n'est pas un multiple de 8, le dernier octet est rempli de zéros. Le codage des informations est un codage à longueur de ligne conforme au codage Packbit défini dans l'ISO 12369 (TIFF, Révision 6.0).

L'événement contient le champ indiqué dans le Tableau D.2.

**Tableau D.2 – Description des champs d'un événement
de détection de mouvement de cellules**

Nom de paramètre	Description
IsMotion	Vrai en cas de détection d'un mouvement

Le fragment de code suivant s'applique à la configuration des règles du détecteur de mouvement de cellules.

```
<tt:VideoAnalyticsConfiguration>
  <tt:RuleEngineConfiguration>
    <tt:Rule Name="MyMotionDetector" Type="tt:CellMotionDetector">
      <tt:Parameters>
        <tt:SimpleItem Name="MinCount" Value="4"/>
        <tt:SimpleItem Name="AlarmOnDelay" Value="1000"/>
        <tt:SimpleItem Name="AlarmOffDelay" Value="1000"/>
        <tt:SimpleItem Name="ActiveCells" Value="/v/+8A==" />
      </tt:Parameters>
    </tt:Rule>
  </tt:RuleEngineConfiguration>
</tt:VideoAnalyticsConfiguration>
```

Les cellules activées sont grisées et les cellules désactivées sont blanches. Les cellules actives avec le codage hex sont les suivantes: "ff ff ff f0 f0 f0". Codage effectué avec l'algorithme Packbit: "fe ff fe f0". Enfin, ces données en paquets sont soumises au codage base64Binary: "/v/+8A==" .

D.2 Moteur d'analyse de mouvement de cellules

D.2.1 Généralités

Un détecteur de mouvement de cellules détermine le mouvement dans une image. Pour ce faire, l'image est sous-divisée en champs de détection carrés ou rectangulaires, appelés cellules. Une cellule est basée sur un nombre défini de pixels, par exemple, une cellule 5 x 5 est une cellule constituée de 25 pixels. Le détecteur tente simplement de déterminer le mouvement dans chaque cellule.

Un détecteur de mouvement de cellules est composé de deux unités, une unité AnalyticsModule et une unité Rule. L'unité AnalyticsModule est composée de deux paramètres: "CellLayout" and "Sensitivity" répertoriés dans le Tableau D.3. Le dernier paramètre peut adapter le détecteur de mouvement à l'environnement auquel est soumise la caméra. Pour des valeurs de sensibilité plus faible, des variations plus élevées des valeurs de luminance doivent être identifiées comme un mouvement. De fait, une sensibilité faible peut être utilisée pour supprimer le mouvement déterminé dû au bruit.

La subdivision des champs de détection dépend du moteur d'analyse et ne peut pas être modifiée par l'utilisateur. Le type "tt:CellLayout", qui décrit la structure des cellules, est composé des trois éléments suivants: "Columns", "Rows", et "Transformation". Les deux premiers paramètres décrivent le nombre de cellules dans les directions X et Y. La première cellule se situe dans l'angle gauche supérieur et la dernière cellule se situe dans l'angle droit inférieur. Le paramètre "Transformation" met en correspondance le système de coordonnées des trames (voir 9.2.2.3) et la grille de cellules. Le point d'origine du nouveau système de coordonnées est l'angle gauche supérieur de la cellule gauche supérieure. La dimension d'unité dans la direction X correspond à la largeur de cellule et la dimension d'unité dans la direction Y correspond à la hauteur de cellule.

D.2.2 Configuration de module

La configuration du module de moteur d'analyse de mouvement de cellules est décrite par le langage de description de module d'analyse:

```
<tt:AnalyticsModuleDescription Name="tt:CellMotionEngine">
  <tt:Parameters>
    <tt:SimpleItemDescription Name="Sensitivity"
                              Type="xs:integer"/>
    <tt:ElementItemDescription Name="Layout" Type="tt:CellLayout"/>
  </tt:Parameters>
</tt:AnalyticsModuleDescription>
```

Tableau D.3 – Paramètres de configuration de module

Nom de paramètre	Description
Sensitivity (Sensibilité)	Plus la valeur de sensibilité définie est faible, plus les variations des valeurs de luminance doivent être élevées afin de pouvoir être identifiées comme un mouvement. De fait, une sensibilité faible peut être utilisée pour supprimer le mouvement déterminé dû au bruit. Plage comprise entre 0 et 100
Layout (Présentation)	La présentation est généralement définie par le matériel du détecteur de mouvement. Elle est fixe et ne peut pas être modifiée. La structure de présentation définit la taille du réseau de cellules, ainsi que leur mise en correspondance avec l'image vidéo.

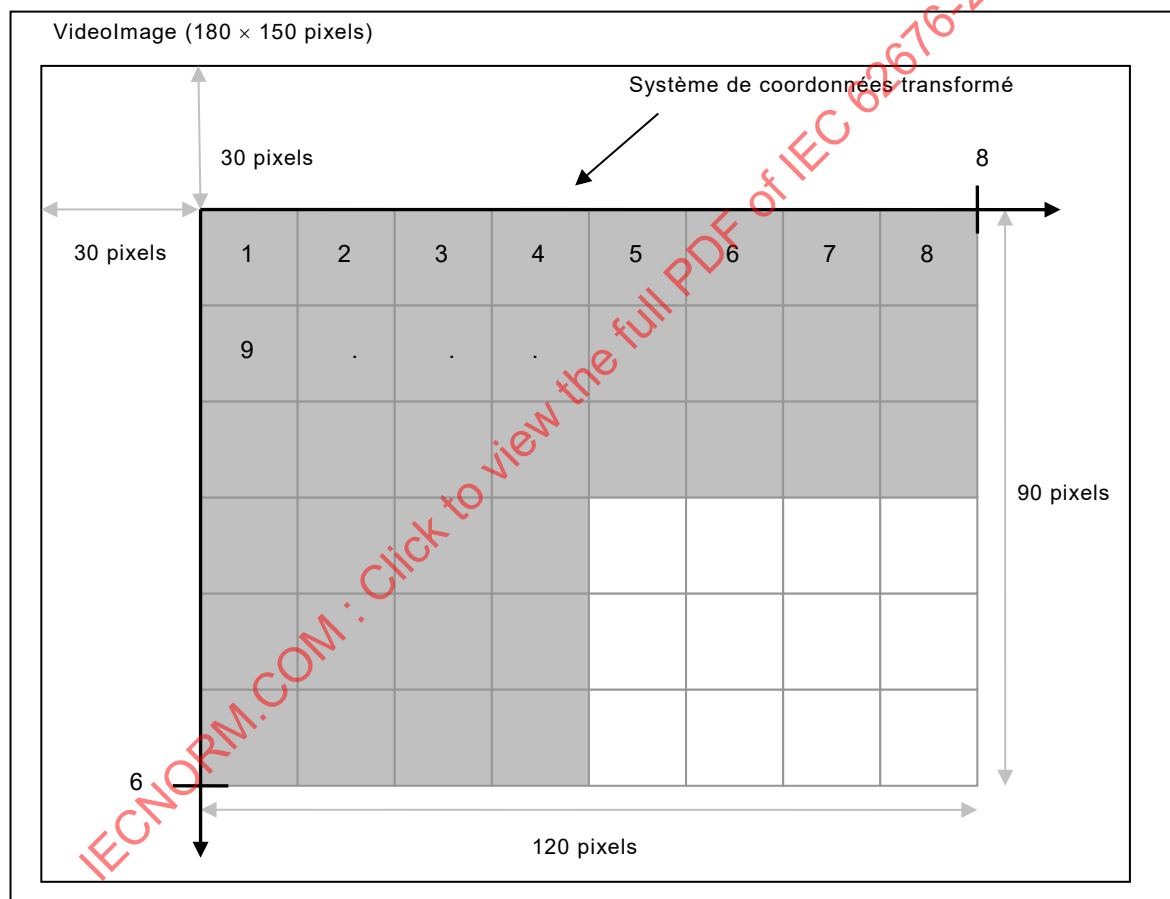
La définition utilise la structure de données tt:CellLayout (voir aussi le Tableau D.4) définie sous la forme:

```
<xs:complexType name="CellLayout">
  <xs:sequence>
    <xs:element name="Transformation" type="tt:Transformation"/>
    <xs:any namespace="##any" processContents="lax"
            minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="Columns" type="xs:integer" use="required"/>
  <xs:attribute name="Rows" type="xs:integer" use="required"/>
  <xs:anyAttribute processContents="lax"/>
</xs:complexType>
```

Tableau D.4 – Description des champs CellLayout

Nom de paramètre	Description
Columns (Colonnes)	Nombre de colonnes de la grille de cellules (dimension x)
Rows (Lignes)	Nombre de lignes de la grille de cellules (dimension y).
Transformation	Mise en correspondance de la grille de cellules et de la trame vidéo. La grille de cellules commence à l'angle gauche supérieur avec la dimension x (de gauche à droite) et la dimension y (de haut en bas).

Un exemple de configuration d'un CellMotionEngine est présenté à la Figure D.1, avec présentation également d'un détecteur de mouvement de cellules 8 x 6. Par hypothèse, la taille de l'image vidéo est de 180 pixels x 150 pixels. Les cellules ont une dimension de 15 pixels x 15 pixels et la taille de la surface active du détecteur de mouvement de cellules est de 120 pixels x 90 pixels.



IEC

Figure D.1 – CellLayout d'un CellMotionEngine 8 x 6


```
<tt:VideoAnalyticsConfiguration>
  <tt:AnalyticsEngineConfiguration>
    <tt:AnalyticsModule Name="MyCellMotion" Type="tt:CellMotionEngine">
      <tt:Parameters>
        <tt:SimpleItem Name="Sensitivity" Value="90"/>
        <tt:ElementItem Name="Layout">
          <tt:CellLayout Columns="8" Rows="6">
            <tt:Transformation>
              <tt:Translate x="-0.66666" y="0.6" />
              <tt:Scale x="0.1666666" y="-0.2" />
            </tt:Transformation>
          </tt:CellLayout>
        </tt:ElementItem>
      </tt:Parameters>
    </tt:AnalyticsModule>
  </tt:AnalyticsEngineConfiguration>
```

IECNORM.COM : Click to view the full PDF of IEC 62676-2-31:2019

Annexe E (normative)

Détection de mouvement

Le détecteur de zone de mouvement détecte tout mouvement par rapport à la zone de mouvement spécifiée. La règle est configurée pour une zone (zone d'intérêt de la source d'image) qui peut être activée ou désactivée. La zone peut être définie par une structure polygonale, mais ne peut l'être par les deux structures. Bien que cette règle soit définie dans le contexte du service d'analyse, elle est destinée à permettre la configuration de la détection de mouvement d'un matériel par opposition à la détection de mouvement à partir des données de description de scène. Pour une description des paramètres, voir le Tableau E.1.

```
<tt:RuleDescription Name="tt:MotionRegionDetector">
  <tt:Parameters>
    <tt:ElementItemDescription Name="MotionRegion"
Type="axt:MotionRegionConfig"/>
  </tt:Parameters>
  <tt:MessageDescription IsProperty="true">
    <tt:Source>
      <tt:SimpleItemDescription Name="VideoSource" Type="tt:ReferenceToken"/>
      <tt:SimpleItemDescription Name="RuleName" Type="xs:string"/>
    </tt:Source>
    <tt:Key/>
    <tt>Data>
      <tt:SimpleItemDescription Name="State" Type="xs:boolean"/>
    </tt>Data>
  </tt:MessageDescription>
</tt:RuleDescription>
```

La description de règle ci-dessus indique qu'une instance de règle génère un événement associé à la rubrique `tns1:RuleEngine/MotionRegionDetector/Motion` avec le champ d'état défini dans le Tableau E.3. L'appel de l'opération `GetRuleOptions` avec le type `tt:MotionRegionDetector` retourne une commande `tt:MotionRegionConfigOptions` avec les paramètres définis dans le Tableau E.2.

**Tableau E.1 – Paramètres de configuration des règles
du détecteur de zone de mouvement**

Nom de paramètre	Description
Armed (Active)	Indique si la Motion Region (Région de mouvement) est active (détection de mouvement) ou inactive (mouvement non détecté).
Rectangle	Fournit les points d'un rectangle spécifié dans une zone définie par l'élément Bounds (Limites) de la VideoSourceConfiguration.
Polygon (Polygone)	Fournit les points d'un polygone spécifié dans une zone définie par l'élément Bounds de la VideoSourceConfiguration. Si le dispositif ne prend pas en charge les polygones, les points du polygone doivent représenter un rectangle.
Sensitivity (Sensibilité)	Indique le niveau de sensibilité du détecteur de mouvement pour cette zone. La valeur de sensibilité est normalisée, 0 représentant la sensibilité inférieure avec laquelle un mouvement important est exigé pour déclencher une alarme et 1 représentant la sensibilité supérieure avec laquelle un mouvement peu important est exigé pour déclencher une alarme.

Tableau E.2 – Options de configuration des règles du détecteur de zone de mouvement

Nom de paramètre	Description
MaxRegions	Nombre total de règles du détecteur de zone de mouvement qui peuvent être créées dans le dispositif.
DisarmSupport	Vraie si le dispositif prend en charge la désactivation d'une règle du détecteur de zone de mouvement.
PolygonSupport	Vraie si le dispositif prend en charge la définition d'une zone en utilisant un polygone et non un rectangle. Un élément Polygon permet toujours de traverser les points du rectangle si le dispositif ne prend pas en charge les zones de polygone. Dans ce cas, les points fournis dans l'élément Polygon doivent représenter un rectangle.
PolygonLimits	Pour les dispositifs qui prennent en charge les polygones avec des limites portant sur le nombre de points, fournit le nombre minimal et maximal de points qui peuvent être définis dans l'élément Polygon.
RuleNotification	Indique que le dispositif fournit le nom de la règle ayant déclenché le mouvement.
SingleSensitivitySupport	Indique que le dispositif ne peut prendre en charge qu'un niveau de sensibilité pour toutes les zones de détection de mouvement définies. La modification de la sensibilité propre à une zone s'applique à toutes les zones.

Tableau E.3 – Description des champs d'un événement de détecteur de zone de mouvement

Nom de paramètre	Description
State (Etat)	Vrai (mouvement détecté) ou Faux (mouvement non détecté)

IECNORM.COM : Click to view the full PDF of IEC 62676-2-31:2019

Annexe F (normative)

Fichiers de schémas

F.1 Dispositif ES

```
<?xml version="1.0" encoding="utf-8"?><?xml-stylesheet type="text/xsl"
href="../../ver20/util/onvif-wsdl-viewer.xsl"?>
<wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap12/"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:tds="http://www.onvif.org/ver10/device/wsdl"
  xmlns:tmd="http://www.onvif.org/ver10/deviceIO/wsdl"
  targetNamespace="http://www.onvif.org/ver10/deviceIO/wsdl">
  <wsdl:import namespace="http://www.onvif.org/ver10/device/wsdl"
    location="../../ver10/device/wsdl/devicemgmt.wsdl"/>
  <wsdl:types>
    <xs:schema xmlns:tt="http://www.onvif.org/ver10/schema"
      targetNamespace="http://www.onvif.org/ver10/deviceIO/wsdl"
      elementFormDefault="qualified" version="16.12">
      <xs:import namespace="http://www.onvif.org/ver10/schema"
        schemaLocation="../../ver10/schema/onvif.xsd"/>
      <xs:element name="GetServiceCapabilities">
        <xs:complexType>
          <xs:sequence/>
        </xs:complexType>
      </xs:element>
      <xs:element name="GetServiceCapabilitiesResponse">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Capabilities" type="tmd:Capabilities"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="Capabilities">
        <xs:sequence>
          <xs:any namespace="##any" processContents="lax" minOccurs="0"
            maxOccurs="unbounded"/>
          <xs:sequence>
            <xs:attribute name="VideoSources" type="xs:int" default="0"/>
            <xs:attribute name="VideoOutputs" type="xs:int" default="0"/>
            <xs:attribute name="AudioSources" type="xs:int" default="0"/>
            <xs:attribute name="AudioOutputs" type="xs:int" default="0"/>
            <xs:attribute name="RelayOutputs" type="xs:int" default="0"/>
            <xs:attribute name="SerialPorts" type="xs:int" default="0"/>
            <xs:attribute name="DigitalInputs" type="xs:int" default="0"/>
            <xs:attribute name="DigitalInputOptions" type="xs:boolean"
              default="false"/>
          </xs:sequence>
          <xs:anyAttribute processContents="lax"/>
        </xs:complexType>
      </xs:element>
      <xs:element name="GetRelayOutputOptions">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="RelayOutputToken" type="tt:ReferenceToken"
              minOccurs="0"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="GetRelayOutputOptionsResponse">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="RelayOutputOptions" type="tmd:RelayOutputOptions"
              minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="RelayOutputOptions">
        <xs:sequence>
          <xs:element name="Mode" type="tt:RelayMode" maxOccurs="unbounded"/>
          <xs:element name="DelayTimes" type="tmd:DelayTimes" minOccurs="0"/>
          <xs:element name="Discrete" type="xs:boolean" minOccurs="0"/>
        </xs:sequence>
      </xs:complexType>
    </xs:schema>
  </wsdl:types>
</wsdl:definitions>
```

```

        <xs:element name="Extension" type="tmd:RelayOutputOptionsExtension"
minOccurs="0"/>

        </xs:sequence>
        <xs:attribute name="token" type="tt:ReferenceToken" use="required"/>
        <xs:anyAttribute processContents="lax"/>
    </xs:complexType>
    <xs:complexType name="RelayOutputOptionsExtension">
        <xs:sequence>
            <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax"/>
        </xs:sequence>
    </xs:complexType>
    <xs:simpleType name="DelayTimes">
        <xs:list itemType="xs:float"/>
    </xs:simpleType>
    <xs:complexType name="Get">
        <xs:sequence/>
    </xs:complexType>
    <xs:complexType name="GetResponse">
        <xs:sequence>
            <xs:element name="Token" type="tt:ReferenceToken" minOccurs="0"
maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
    <xs:element name="GetVideoSources" type="tmd:Get"/>
    <xs:element name="GetVideoSourcesResponse" type="tmd:GetResponse"/>
    <xs:element name="GetAudioSources" type="tmd:Get"/>
    <xs:element name="GetAudioSourcesResponse" type="tmd:GetResponse"/>
    <xs:element name="GetAudioOutputs" type="tmd:Get"/>
    <xs:element name="GetAudioOutputsResponse" type="tmd:GetResponse"/>
    <xs:element name="GetVideoOutputs">
        <xs:complexType>
            <xs:sequence/>
        </xs:complexType>
    </xs:element>
    <xs:element name="GetVideoOutputsResponse">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="VideoOutputs" type="tt:VideoOutput" minOccurs="0"
maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="GetAudioSourceConfiguration">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="AudioSourceToken" type="tt:ReferenceToken"/>
                <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="GetAudioSourceConfigurationResponse">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="AudioSourceConfiguration"
type="tt:AudioSourceConfiguration"/>
                <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="GetAudioOutputConfiguration">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="AudioOutputToken" type="tt:ReferenceToken"/>
                <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="GetAudioOutputConfigurationResponse">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="AudioOutputConfiguration"
type="tt:AudioOutputConfiguration"/>
                <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"

```

```

processContents="lax"/>
    </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="GetVideoSourceConfiguration">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="VideoSourceToken" type="tt:ReferenceToken"/>
            <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="GetVideoSourceConfigurationResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="VideoSourceConfiguration"
type="tt:VideoSourceConfiguration"/>
            <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="GetVideoOutputConfiguration">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="VideoOutputToken" type="tt:ReferenceToken"/>
            <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="GetVideoOutputConfigurationResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="VideoOutputConfiguration"
type="tt:VideoOutputConfiguration"/>
            <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="SetAudioSourceConfiguration">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Configuration" type="tt:AudioSourceConfiguration"/>
            <xs:element name="ForcePersistence" type="xs:boolean"/>
            <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="SetAudioSourceConfigurationResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Configuration" type="tt:AudioSourceConfiguration"/>
            <xs:element name="ForcePersistence" type="xs:boolean"/>
            <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="SetAudioOutputConfiguration">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Configuration" type="tt:AudioOutputConfiguration"/>
            <xs:element name="ForcePersistence" type="xs:boolean"/>
            <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="SetAudioOutputConfigurationResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Configuration" type="tt:AudioOutputConfiguration"/>
            <xs:element name="ForcePersistence" type="xs:boolean"/>
            <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>

```

```

<xs:element name="SetVideoSourceConfiguration">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Configuration" type="tt:VideoSourceConfiguration"/>
      <xs:element name="ForcePersistence" type="xs:boolean"/>
      <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="SetVideoSourceConfigurationResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="SetVideoOutputConfiguration">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Configuration" type="tt:VideoOutputConfiguration"/>
      <xs:element name="ForcePersistence" type="xs:boolean"/>
      <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="SetVideoOutputConfigurationResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="GetVideoSourceConfigurationOptions">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="VideoSourceToken" type="tt:ReferenceToken"/>
      <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="GetVideoSourceConfigurationOptionsResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="VideoSourceConfigurationOptions"
type="tt:VideoSourceConfigurationOptions"/>
      <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="GetVideoOutputConfigurationOptions">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="VideoOutputToken" type="tt:ReferenceToken"/>
      <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="GetVideoOutputConfigurationOptionsResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="VideoOutputConfigurationOptions"
type="tt:VideoOutputConfigurationOptions"/>
      <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="GetAudioSourceConfigurationOptions">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="AudioSourceToken" type="tt:ReferenceToken"/>

```

```

        <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax"/>
    </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="GetAudioSourceConfigurationOptionsResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="AudioSourceOptions"
type="tt:AudioSourceConfigurationOptions"/>
        </xs:sequence>
    </xs:complexType>
    </xs:element>
<xs:element name="GetAudioOutputConfigurationOptions">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="AudioOutputToken" type="tt:ReferenceToken"/>
            <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax"/>
        </xs:sequence>
    </xs:complexType>
    </xs:element>
<xs:element name="GetAudioOutputConfigurationOptionsResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="AudioOutputOptions"
type="tt:AudioOutputConfigurationOptions"/>
            <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax"/>
        </xs:sequence>
    </xs:complexType>
    </xs:element>
<xs:element name="SetRelayOutputSettings">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="RelayOutput" type="tt:RelayOutput"/>
        </xs:sequence>
    </xs:complexType>
    </xs:element>
<xs:element name="SetRelayOutputSettingsResponse">
    <xs:complexType>
        <xs:sequence/>
    </xs:complexType>
    </xs:element>
<xs:element name="GetDigitalInputs">
    <xs:complexType>
        <xs:sequence/>
    </xs:complexType>
    </xs:element>
<xs:element name="GetDigitalInputsResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="DigitalInputs" type="tt:DigitalInput" minOccurs="0"
maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
    </xs:element>
<xs:complexType name="DigitalInputConfigurationInputOptions">
    <xs:sequence>
        <xs:element name="IdleState" type="tt:DigitalIdleState"
maxOccurs="unbounded"/>
        <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
<xs:element name="GetDigitalInputConfigurationOptions">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Token" type="tt:ReferenceToken" minOccurs="0"/>
        </xs:sequence>
    </xs:complexType>
    </xs:element>
<xs:element name="GetDigitalInputConfigurationOptionsResponse">
    <xs:complexType>
        <xs:sequence>

```



```

        <xs:element name="DigitalInputOptions"
type="tmd:DigitalInputConfigurationInputOptions"/>
    </xs:sequence>
</xs:complexType>
</xs:element>
    <xs:element name="SetDigitalInputConfigurations">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="DigitalInputs" type="tt:DigitalInput"
maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="SetDigitalInputConfigurationsResponse">
        <xs:complexType>
            <xs:sequence/>
        </xs:complexType>
    </xs:element>
    <xs:element name="GetSerialPorts">
        <xs:complexType>
            <xs:sequence/>
        </xs:complexType>
    </xs:element>
    <xs:element name="GetSerialPortsResponse">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="SerialPort" type="tmd:SerialPort" minOccurs="0"
maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="GetSerialPortConfiguration">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="SerialPortToken" type="tt:ReferenceToken"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="GetSerialPortConfigurationResponse">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="SerialPortConfiguration"
type="tmd:SerialPortConfiguration"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="SetSerialPortConfiguration">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="SerialPortConfiguration"
type="tmd:SerialPortConfiguration"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="ForcePersistence" type="xs:boolean"/>
    <xs:element name="SetSerialPortConfigurationResponse">
        <xs:complexType>
            <xs:sequence/>
        </xs:complexType>
    </xs:element>
    <xs:element name="GetSerialPortConfigurationOptions">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="SerialPortToken" type="tt:ReferenceToken"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="GetSerialPortConfigurationOptionsResponse">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="SerialPortOptions"
type="tmd:SerialPortConfigurationOptions"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="SendReceiveSerialCommand">
        <xs:complexType>
            <xs:sequence>

```

```

        <xs:element name="SerialData" type="tmd:SerialData" minOccurs="0"/>
        <xs:element name="Timeout" type="xs:duration" minOccurs="0"/>
        <xs:element name="DataLength" type="xs:integer" minOccurs="0"/>
        <xs:element name="Delimiter" type="xs:string" minOccurs="0"/>
    </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="SendReceiveSerialCommandResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="SerialData" type="tmd:SerialData" minOccurs="0"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:complexType name="SerialData">
    <xs:choice>
        <xs:element name="Binary" type="xs:base64Binary"/>
        <xs:element name="String" type="xs:string"/>
    </xs:choice>
    <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="SerialPort">
    <xs:complexContent>
        <xs:extension base="tt:DeviceEntity">
            <xs:sequence>
                <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
            </xs:sequence>
            <xs:anyAttribute processContents="lax"/>
        </xs:extension>
    </xs:complexType>
</xs:complexType>
<xs:simpleType name="SerialPortType">
    <xs:restriction base="xs:string">
        <xs:enumeration value="RS232"/>
        <xs:enumeration value="RS422HalfDuplex"/>
        <xs:enumeration value="RS422FullDuplex"/>
        <xs:enumeration value="RS485HalfDuplex"/>
        <xs:enumeration value="RS485FullDuplex"/>
        <xs:enumeration value="Generic"/>
    </xs:restriction>
</xs:simpleType>
<xs:complexType name="SerialPortConfiguration">
    <xs:sequence>
        <xs:element name="BaudRate" type="xs:int"/>
        <xs:element name="ParityBit" type="tmd:ParityBit"/>
        <xs:element name="CharacterLength" type="xs:int"/>
        <xs:element name="StopBit" type="xs:float"/>
        <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="token" type="tt:ReferenceToken" use="required"/>
    <xs:attribute name="type" type="tmd:SerialPortType" use="required"/>
    <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:simpleType name="ParityBit">
    <xs:restriction base="xs:string">
        <xs:enumeration value="None"/>
        <xs:enumeration value="Even"/>
        <xs:enumeration value="Odd"/>
        <xs:enumeration value="Mark"/>
        <xs:enumeration value="Space"/>
        <xs:enumeration value="Extended"/>
    </xs:restriction>
</xs:simpleType>
<xs:complexType name="SerialPortConfigurationOptions">
    <xs:sequence>
        <xs:element name="BaudRateList" type="tt:IntList"/>
        <xs:element name="ParityBitList" type="tmd:ParityBitList"/>
        <xs:element name="CharacterLengthList" type="tt:IntList"/>
        <xs:element name="StopBitList" type="tt:FloatList"/>
        <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="token" type="tt:ReferenceToken" use="required"/>
    <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="ParityBitList">

```

```

        <xs:sequence>
          <xs:element name="Items" type="tmd:ParityBit" minOccurs="0"
maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:complexType>
    </xs:schema>
  </wsdl:types>
  <wsdl:message name="parameters" element="tmd:GetServiceCapabilitiesRequest">
    <wsdl:part name="parameters" element="tmd:GetServiceCapabilitiesRequest"/>
  </wsdl:message>
  <wsdl:message name="parameters" element="tmd:GetServiceCapabilitiesResponse">
    <wsdl:part name="parameters" element="tmd:GetServiceCapabilitiesResponse"/>
  </wsdl:message>
  <wsdl:message name="parameters" element="tmd:GetRelayOutputOptionsRequest">
    <wsdl:part name="parameters" element="tmd:GetRelayOutputOptionsRequest"/>
  </wsdl:message>
  <wsdl:message name="parameters" element="tmd:GetRelayOutputOptionsResponse">
    <wsdl:part name="parameters" element="tmd:GetRelayOutputOptionsResponse"/>
  </wsdl:message>
  <wsdl:message name="parameters" element="tmd:GetVideoOutputsRequest">
    <wsdl:part name="parameters" element="tmd:GetVideoOutputsRequest"/>
  </wsdl:message>
  <wsdl:message name="parameters" element="tmd:GetVideoOutputsResponse">
    <wsdl:part name="parameters" element="tmd:GetVideoOutputsResponse"/>
  </wsdl:message>
  <wsdl:message name="parameters" element="tmd:GetAudioOutputsRequest">
    <wsdl:part name="parameters" element="tmd:GetAudioOutputsRequest"/>
  </wsdl:message>
  <wsdl:message name="parameters" element="tmd:GetAudioOutputsResponse">
    <wsdl:part name="parameters" element="tmd:GetAudioOutputsResponse"/>
  </wsdl:message>
  <wsdl:message name="parameters" element="tmd:GetVideoSourcesRequest">
    <wsdl:part name="parameters" element="tmd:GetVideoSourcesRequest"/>
  </wsdl:message>
  <wsdl:message name="parameters" element="tmd:GetVideoSourcesResponse">
    <wsdl:part name="parameters" element="tmd:GetVideoSourcesResponse"/>
  </wsdl:message>
  <wsdl:message name="parameters" element="tmd:GetAudioSourcesRequest">
    <wsdl:part name="parameters" element="tmd:GetAudioSourcesRequest"/>
  </wsdl:message>
  <wsdl:message name="parameters" element="tmd:GetAudioSourcesResponse">
    <wsdl:part name="parameters" element="tmd:GetAudioSourcesResponse"/>
  </wsdl:message>
  <wsdl:message name="parameters" element="tmd:GetVideoSourceConfigurationRequest">
    <wsdl:part name="parameters" element="tmd:GetVideoSourceConfigurationRequest"/>
  </wsdl:message>
  <wsdl:message name="parameters" element="tmd:GetVideoSourceConfigurationResponse">
    <wsdl:part name="parameters" element="tmd:GetVideoSourceConfigurationResponse"/>
  </wsdl:message>
  <wsdl:message name="parameters" element="tmd:GetVideoOutputConfigurationRequest">
    <wsdl:part name="parameters" element="tmd:GetVideoOutputConfigurationRequest"/>
  </wsdl:message>
  <wsdl:message name="parameters" element="tmd:GetVideoOutputConfigurationResponse">
    <wsdl:part name="parameters" element="tmd:GetVideoOutputConfigurationResponse"/>
  </wsdl:message>
  <wsdl:message name="parameters" element="tmd:GetAudioSourceConfigurationRequest">
    <wsdl:part name="parameters" element="tmd:GetAudioSourceConfigurationRequest"/>
  </wsdl:message>
  <wsdl:message name="parameters" element="tmd:GetAudioSourceConfigurationResponse">
    <wsdl:part name="parameters" element="tmd:GetAudioSourceConfigurationResponse"/>
  </wsdl:message>
  <wsdl:message name="parameters" element="tmd:GetAudioOutputConfigurationRequest">
    <wsdl:part name="parameters" element="tmd:GetAudioOutputConfigurationRequest"/>
  </wsdl:message>
  <wsdl:message name="parameters" element="tmd:GetAudioOutputConfigurationResponse">
    <wsdl:part name="parameters" element="tmd:GetAudioOutputConfigurationResponse"/>
  </wsdl:message>
  <wsdl:message name="parameters" element="tmd:SetVideoSourceConfigurationRequest">
    <wsdl:part name="parameters" element="tmd:SetVideoSourceConfigurationRequest"/>
  </wsdl:message>
  <wsdl:message name="parameters" element="tmd:SetVideoSourceConfigurationResponse">
    <wsdl:part name="parameters" element="tmd:SetVideoSourceConfigurationResponse"/>
  </wsdl:message>
  <wsdl:message name="parameters" element="tmd:SetVideoOutputConfigurationRequest">
    <wsdl:part name="parameters" element="tmd:SetVideoOutputConfigurationRequest"/>
  </wsdl:message>
  <wsdl:message name="parameters" element="tmd:SetVideoOutputConfigurationResponse">
    <wsdl:part name="parameters" element="tmd:SetVideoOutputConfigurationResponse"/>
  </wsdl:message>

```

```

    <wsdl:part name="parameters" element="tmd:SetVideoOutputConfigurationResponse"/>
</wsdl:message>
<wsdl:message name="SetAudioSourceConfigurationRequest">
    <wsdl:part name="parameters" element="tmd:SetAudioSourceConfiguration"/>
</wsdl:message>
<wsdl:message name="SetAudioSourceConfigurationResponse">
    <wsdl:part name="parameters" element="tmd:SetAudioSourceConfigurationResponse"/>
</wsdl:message>
<wsdl:message name="SetAudioOutputConfigurationRequest">
    <wsdl:part name="parameters" element="tmd:SetAudioOutputConfiguration"/>
</wsdl:message>
<wsdl:message name="SetAudioOutputConfigurationResponse">
    <wsdl:part name="parameters" element="tmd:SetAudioOutputConfigurationResponse"/>
</wsdl:message>
<wsdl:message name="GetVideoSourceConfigurationOptionsRequest">
    <wsdl:part name="parameters" element="tmd:GetVideoSourceConfigurationOptions"/>
</wsdl:message>
<wsdl:message name="GetVideoSourceConfigurationOptionsResponse">
    <wsdl:part name="parameters" element="tmd:GetVideoSourceConfigurationOptionsResponse"/>
</wsdl:message>
<wsdl:message name="GetVideoOutputConfigurationOptionsRequest">
    <wsdl:part name="parameters" element="tmd:GetVideoOutputConfigurationOptions"/>
</wsdl:message>
<wsdl:message name="GetVideoOutputConfigurationOptionsResponse">
    <wsdl:part name="parameters" element="tmd:GetVideoOutputConfigurationOptionsResponse"/>
</wsdl:message>
<wsdl:message name="GetAudioSourceConfigurationOptionsRequest">
    <wsdl:part name="parameters" element="tmd:GetAudioSourceConfigurationOptions"/>
</wsdl:message>
<wsdl:message name="GetAudioSourceConfigurationOptionsResponse">
    <wsdl:part name="parameters" element="tmd:GetAudioSourceConfigurationOptionsResponse"/>
</wsdl:message>
<wsdl:message name="GetAudioOutputConfigurationOptionsRequest">
    <wsdl:part name="parameters" element="tmd:GetAudioOutputConfigurationOptions"/>
</wsdl:message>
<wsdl:message name="GetAudioOutputConfigurationOptionsResponse">
    <wsdl:part name="parameters" element="tmd:GetAudioOutputConfigurationOptionsResponse"/>
</wsdl:message>
<wsdl:message name="GetRelayOutputsRequest">
    <wsdl:part name="parameters" element="tds:GetRelayOutputs"/>
</wsdl:message>
<wsdl:message name="GetRelayOutputsResponse">
    <wsdl:part name="parameters" element="tds:GetRelayOutputsResponse"/>
</wsdl:message>
<wsdl:message name="SetRelayOutputSettingsRequest">
    <wsdl:part name="parameters" element="tmd:SetRelayOutputSettings"/>
</wsdl:message>
<wsdl:message name="SetRelayOutputSettingsResponse">
    <wsdl:part name="parameters" element="tmd:SetRelayOutputSettingsResponse"/>
</wsdl:message>
<wsdl:message name="SetRelayOutputStateRequest">
    <wsdl:part name="parameters" element="tds:SetRelayOutputState"/>
</wsdl:message>
<wsdl:message name="SetRelayOutputStateResponse">
    <wsdl:part name="parameters" element="tds:SetRelayOutputStateResponse"/>
</wsdl:message>
<wsdl:message name="GetDigitalInputsRequest">
    <wsdl:part name="parameters" element="tmd:GetDigitalInputs"/>
</wsdl:message>
<wsdl:message name="GetDigitalInputsResponse">
    <wsdl:part name="parameters" element="tmd:GetDigitalInputsResponse"/>
</wsdl:message>
<wsdl:message name="GetDigitalInputConfigurationOptionsRequest">
    <wsdl:part name="parameters" element="tmd:GetDigitalInputConfigurationOptions"/>
</wsdl:message>
<wsdl:message name="GetDigitalInputConfigurationOptionsResponse">
    <wsdl:part name="parameters" element="tmd:GetDigitalInputConfigurationOptionsResponse"/>
</wsdl:message>
<wsdl:message name="SetDigitalInputConfigurationsRequest">
    <wsdl:part name="parameters" element="tmd:SetDigitalInputConfigurations"/>
</wsdl:message>
<wsdl:message name="SetDigitalInputConfigurationsResponse">
    <wsdl:part name="parameters" element="tmd:SetDigitalInputConfigurationsResponse"/>
</wsdl:message>

```

```

element="tmd:SetDigitalInputConfigurationsResponse"/>
</wsdl:message>
<wsdl:message
  <wsdl:part
    name="parameters"
  >
    name="GetSerialPortsRequest">
    element="tmd:GetSerialPorts"/>
  </wsdl:part>
</wsdl:message>
<wsdl:message
  <wsdl:part
    name="parameters"
  >
    name="GetSerialPortsResponse">
    element="tmd:GetSerialPortsResponse"/>
  </wsdl:part>
</wsdl:message>
<wsdl:message
  <wsdl:part
    name="parameters"
  >
    name="GetSerialPortConfigurationRequest">
    element="tmd:GetSerialPortConfiguration"/>
  </wsdl:part>
</wsdl:message>
<wsdl:message
  <wsdl:part
    name="parameters"
  >
    name="GetSerialPortConfigurationResponse">
    element="tmd:GetSerialPortConfigurationResponse"/>
  </wsdl:part>
</wsdl:message>
<wsdl:message
  <wsdl:part
    name="parameters"
  >
    name="SetSerialPortConfigurationRequest">
    element="tmd:SetSerialPortConfiguration"/>
  </wsdl:part>
</wsdl:message>
<wsdl:message
  <wsdl:part
    name="parameters"
  >
    name="SetSerialPortConfigurationResponse">
    element="tmd:SetSerialPortConfigurationResponse"/>
  </wsdl:part>
</wsdl:message>
<wsdl:message
  <wsdl:part
    name="parameters"
  >
    name="GetSerialPortConfigurationOptionsRequest">
    element="tmd:GetSerialPortConfigurationOptions"/>
  </wsdl:part>
</wsdl:message>
<wsdl:message
  <wsdl:part
    name="parameters"
  >
    name="GetSerialPortConfigurationOptionsResponse">
    element="tmd:GetSerialPortConfigurationOptionsResponse"/>
  </wsdl:part>
</wsdl:message>
<wsdl:portType
  <wsdl:operation
    <wsdl:input
      name="parameters"
    >
      name="SendReceiveSerialCommandRequest">
      element="tmd:SendReceiveSerialCommand"/>
    <wsdl:output
      name="parameters"
    >
      name="SendReceiveSerialCommandResponse">
      element="tmd:SendReceiveSerialCommandResponse"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation
    <wsdl:input
      name="parameters"
    >
      name="DeviceIOPort">
      message="tmd:GetServiceCapabilitiesRequest"/>
    <wsdl:output
      name="parameters"
    >
      message="tmd:GetServiceCapabilitiesResponse"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation
    <wsdl:input
      name="parameters"
    >
      name="GetRelayOutputOptions">
      message="tmd:GetRelayOutputOptionsRequest"/>
    <wsdl:output
      name="parameters"
    >
      message="tmd:GetRelayOutputOptionsResponse"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation
    <wsdl:input
      name="parameters"
    >
      name="GetAudioSources">
      message="tmd:GetAudioSourcesRequest"/>
    <wsdl:output
      name="parameters"
    >
      message="tmd:GetAudioSourcesResponse"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation
    <wsdl:input
      name="parameters"
    >
      name="GetAudioOutputs">
      message="tmd:GetAudioOutputsRequest"/>
    <wsdl:output
      name="parameters"
    >
      message="tmd:GetAudioOutputsResponse"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation
    <wsdl:input
      name="parameters"
    >
      name="GetVideoSources">
      message="tmd:GetVideoSourcesRequest"/>
    <wsdl:output
      name="parameters"
    >
      message="tmd:GetVideoSourcesResponse"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation
    <wsdl:input
      name="parameters"
    >
      name="GetVideoOutputs">
      message="tmd:GetVideoOutputsRequest"/>
    <wsdl:output
      name="parameters"
    >
      message="tmd:GetVideoOutputsResponse"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation
    <wsdl:input
      name="parameters"
    >
      name="GetVideoSourceConfiguration">
      message="tmd:GetVideoSourceConfigurationRequest"/>
    <wsdl:output
      name="parameters"
    >
      message="tmd:GetVideoSourceConfigurationResponse"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation
    <wsdl:input
      name="parameters"
    >
      name="GetVideoOutputConfiguration">
      message="tmd:GetVideoOutputConfigurationRequest"/>
    <wsdl:output
      name="parameters"
    >
      message="tmd:GetVideoOutputConfigurationResponse"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation
    <wsdl:input
      name="parameters"
    >
      name="GetAudioSourceConfiguration">
      message="tmd:GetAudioSourceConfigurationRequest"/>
    <wsdl:output
      name="parameters"
    >
      message="tmd:GetAudioSourceConfigurationResponse"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation
    <wsdl:input
      name="parameters"
    >
      name="GetAudioOutputConfiguration">
      message="tmd:GetAudioOutputConfigurationRequest"/>
    <wsdl:output
      name="parameters"
    >
      message="tmd:GetAudioOutputConfigurationResponse"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation
    <wsdl:input
      name="parameters"
    >
      name="SetVideoSourceConfiguration">
      message="tmd:SetVideoSourceConfigurationRequest"/>
    <wsdl:output
      name="parameters"
    >
      message="tmd:SetVideoSourceConfigurationResponse"/>
    </wsdl:output>
  </wsdl:operation>

```

```

<wsdl:operation                                name="SetVideoOutputConfiguration">
  <wsdl:input                                  message="tmd:SetVideoOutputConfigurationRequest"/>
  <wsdl:output                                  message="tmd:SetVideoOutputConfigurationResponse"/>
</wsdl:operation>
<wsdl:operation                                name="SetAudioSourceConfiguration">
  <wsdl:input                                  message="tmd:SetAudioSourceConfigurationRequest"/>
  <wsdl:output                                  message="tmd:SetAudioSourceConfigurationResponse"/>
</wsdl:operation>
<wsdl:operation                                name="SetAudioOutputConfiguration">
  <wsdl:input                                  message="tmd:SetAudioOutputConfigurationRequest"/>
  <wsdl:output                                  message="tmd:SetAudioOutputConfigurationResponse"/>
</wsdl:operation>
<wsdl:operation                                name="GetVideoSourceConfigurationOptions">
  <wsdl:input                                  message="tmd:GetVideoSourceConfigurationOptionsRequest"/>
  <wsdl:output                                  message="tmd:GetVideoSourceConfigurationOptionsResponse"/>
</wsdl:operation>
<wsdl:operation                                name="GetVideoOutputConfigurationOptions">
  <wsdl:input                                  message="tmd:GetVideoOutputConfigurationOptionsRequest"/>
  <wsdl:output                                  message="tmd:GetVideoOutputConfigurationOptionsResponse"/>
</wsdl:operation>
<wsdl:operation                                name="GetAudioSourceConfigurationOptions">
  <wsdl:input                                  message="tmd:GetAudioSourceConfigurationOptionsRequest"/>
  <wsdl:output                                  message="tmd:GetAudioSourceConfigurationOptionsResponse"/>
</wsdl:operation>
<wsdl:operation                                name="GetAudioOutputConfigurationOptions">
  <wsdl:input                                  message="tmd:GetAudioOutputConfigurationOptionsRequest"/>
  <wsdl:output                                  message="tmd:GetAudioOutputConfigurationOptionsResponse"/>
</wsdl:operation>
<wsdl:operation                                name="GetRelayOutputs">
  <wsdl:input                                  message="tmd:GetRelayOutputsRequest"/>
  <wsdl:output                                  message="tmd:GetRelayOutputsResponse"/>
</wsdl:operation>
<wsdl:operation                                name="SetRelayOutputSettings">
  <wsdl:input                                  message="tmd:SetRelayOutputSettingsRequest"/>
  <wsdl:output                                  message="tmd:SetRelayOutputSettingsResponse"/>
</wsdl:operation>
<wsdl:operation                                name="SetRelayOutputState">
  <wsdl:input                                  message="tmd:SetRelayOutputStateRequest"/>
  <wsdl:output                                  message="tmd:SetRelayOutputStateResponse"/>
</wsdl:operation>
<wsdl:operation                                name="GetDigitalInputs">
  <wsdl:input                                  message="tmd:GetDigitalInputsRequest"/>
  <wsdl:output                                  message="tmd:GetDigitalInputsResponse"/>
</wsdl:operation>
<wsdl:operation                                name="GetDigitalInputConfigurationOptions">
  <wsdl:input                                  message="tmd:GetDigitalInputConfigurationOptionsRequest"/>
  <wsdl:output                                  message="tmd:GetDigitalInputConfigurationOptionsResponse"/>
</wsdl:operation>
<wsdl:operation                                name="SetDigitalInputConfigurations">
  <wsdl:input                                  message="tmd:SetDigitalInputConfigurationsRequest"/>
  <wsdl:output                                  message="tmd:SetDigitalInputConfigurationsResponse"/>
</wsdl:operation>
<wsdl:operation                                name="GetSerialPorts">
  <wsdl:input                                  message="tmd:GetSerialPortsRequest"/>
  <wsdl:output                                  message="tmd:GetSerialPortsResponse"/>
</wsdl:operation>
<wsdl:operation                                name="GetSerialPortConfiguration">
  <wsdl:input                                  message="tmd:GetSerialPortConfigurationRequest"/>
  <wsdl:output                                  message="tmd:GetSerialPortConfigurationResponse"/>
</wsdl:operation>
<wsdl:operation                                name="SetSerialPortConfiguration">
  <wsdl:input                                  message="tmd:SetSerialPortConfigurationRequest"/>
  <wsdl:output                                  message="tmd:SetSerialPortConfigurationResponse"/>
</wsdl:operation>
<wsdl:operation                                name="GetSerialPortConfigurationOptions">
  <wsdl:input                                  message="tmd:GetSerialPortConfigurationOptionsRequest"/>
  <wsdl:output                                  message="tmd:GetSerialPortConfigurationOptionsResponse"/>
</wsdl:operation>
<wsdl:operation                                name="SendReceiveSerialCommand">
  <wsdl:input                                  message="tmd:SendReceiveSerialCommandRequest"/>
  <wsdl:output                                  message="tmd:SendReceiveSerialCommandResponse"/>
</wsdl:operation>
</wsdl:portType>
<wsdl:binding                                name="DeviceIOBinding"                                type="tmd:DeviceIOPort">
  <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
  <wsdl:operation                                name="GetServiceCapabilities">
    <soap:operation>
      soapAction="http://www.onvif.org/ver10/deviceio/wsdl/GetServiceCapabilities"/>
    </soap:operation>
  </wsdl:operation>

```



```

        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetRelayOutputOptions">
        <soap:operation
soapAction="http://www.onvif.org/ver10/deviceio/wsdl/GetRelayOutputOptions"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetAudioSources">
        <soap:operation
soapAction="http://www.onvif.org/ver10/deviceio/wsdl/GetAudioSources"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetAudioOutputs">
        <soap:operation
soapAction="http://www.onvif.org/ver10/deviceio/wsdl/GetAudioOutputs"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetVideoSources">
        <soap:operation
soapAction="http://www.onvif.org/ver10/deviceio/wsdl/GetVideoSources"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetVideoOutputs">
        <soap:operation
soapAction="http://www.onvif.org/ver10/deviceio/wsdl/GetVideoOutputs"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetVideoSourceConfiguration">
        <soap:operation
soapAction="http://www.onvif.org/ver10/deviceio/wsdl/GetVideoSourceConfiguration"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetVideoOutputConfiguration">
        <soap:operation
soapAction="http://www.onvif.org/ver10/deviceio/wsdl/GetVideoOutputConfiguration"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetAudioSourceConfiguration">

```

```

        <soap:operation
soapAction="http://www.onvif.org/ver10/deviceio/wsdl/GetAudioSourceConfiguration"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetAudioOutputConfiguration">
        <soap:operation
soapAction="http://www.onvif.org/ver10/deviceio/wsdl/GetAudioOutputConfiguration"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="SetVideoSourceConfiguration">
        <soap:operation
soapAction="http://www.onvif.org/ver10/deviceio/wsdl/SetVideoSourceConfiguration"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="SetVideoOutputConfiguration">
        <soap:operation
soapAction="http://www.onvif.org/ver10/deviceio/wsdl/SetVideoOutputConfiguration"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="SetAudioSourceConfiguration">
        <soap:operation
soapAction="http://www.onvif.org/ver10/deviceio/wsdl/SetAudioSourceConfiguration"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="SetAudioOutputConfiguration">
        <soap:operation
soapAction="http://www.onvif.org/ver10/deviceio/wsdl/SetAudioOutputConfiguration"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetVideoSourceConfigurationOptions">
        <soap:operation
soapAction="http://www.onvif.org/ver10/deviceio/wsdl/GetVideoSourceConfigurationOptions
"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetVideoOutputConfigurationOptions">
        <soap:operation
soapAction="http://www.onvif.org/ver10/deviceio/wsdl/GetVideoOutputConfigurationOptions
"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>

```



```

        <soap:body use="literal"/>
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetAudioSourceConfigurationOptions">
    <soap:operation
soapAction="http://www.onvif.org/ver10/deviceio/wsdl/GetAudioSourceConfigurationOptions
"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
<wsdl:operation name="GetAudioOutputConfigurationOptions">
    <soap:operation
soapAction="http://www.onvif.org/ver10/deviceio/wsdl/GetAudioOutputConfigurationOptions
"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
<wsdl:operation name="GetRelayOutputs">
    <soap:operation
soapAction="http://www.onvif.org/ver10/deviceio/wsdl/GetRelayOutputs"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
<wsdl:operation name="SetRelayOutputSettings">
    <soap:operation
soapAction="http://www.onvif.org/ver10/deviceio/wsdl/SetRelayOutputSettings"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
<wsdl:operation name="SetRelayOutputState">
    <soap:operation
soapAction="http://www.onvif.org/ver10/deviceio/wsdl/SetRelayOutputState"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
<wsdl:operation name="GetDigitalInputs">
    <soap:operation
soapAction="http://www.onvif.org/ver10/deviceio/wsdl/GetDigitalInputs"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
<wsdl:operation name="GetDigitalInputConfigurationOptions">
    <soap:operation
soapAction="http://www.onvif.org/ver10/deviceio/wsdl/GetDigitalInputConfigurationOption
s"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
<wsdl:operation name="SetDigitalInputConfigurations">
    <soap:operation

```

```

soapAction="http://www.onvif.org/ver10/deviceio/wsdl/SetDigitalInputConfigurations"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetSerialPorts">
  <soap:operation
soapAction="http://www.onvif.org/ver10/deviceio/wsdl/GetSerialPorts"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetSerialPortConfiguration">
  <soap:operation
soapAction="http://www.onvif.org/ver10/deviceio/wsdl/GetSerialPortConfigurations"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="SetSerialPortConfiguration">
  <soap:operation
soapAction="http://www.onvif.org/ver10/deviceio/wsdl/SetSerialPortConfiguration"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetSerialPortConfigurationOptions">
  <soap:operation
soapAction="http://www.onvif.org/ver10/deviceio/wsdl/GetSerialPortConfigurationOptions"
/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="SendReceiveSerialCommand">
  <soap:operation
soapAction="http://www.onvif.org/ver10/deviceio/wsdl/SendReceiveSerialCommand"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
</wsdl:binding>
</wsdl:definitions>

```

F.2 Imagerie

```
<?xml version="1.0" encoding="utf-8"?><?xml-stylesheet type="text/xsl"
href="../../../ver20/util/onvif-wsdl-viewer.xsl"?><wsdl:definitions
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:timg="http://www.onvif.org/ver20/imaging/wsdl"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap12/"
xmlns:tt="http://www.onvif.org/ver10/schema" name="ImagingService"
targetNamespace="http://www.onvif.org/ver20/imaging/wsdl">
  <wsdl:types>
    <xs:schema targetNamespace="http://www.onvif.org/ver20/imaging/wsdl"
elementFormDefault="qualified" version="16.06">
      <xs:import namespace="http://www.onvif.org/ver10/schema"
schemaLocation="../../../ver10/schema/onvif.xsd"/>
      <xs:element name="GetServiceCapabilities">
        <xs:complexType>
          <xs:sequence/>
        </xs:complexType>
      </xs:element>
      <xs:element name="GetServiceCapabilitiesResponse">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Capabilities" type="timg:Capabilities"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="Capabilities">
        <xs:sequence>
          <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
          <xs:attribute name="ImageStabilization" type="xs:boolean"/>
          <xs:attribute name="Presets" type="xs:boolean"/>
          <xs:anyAttribute processContents="lax"/>
        </xs:sequence>
      </xs:complexType>
      <xs:element name="Capabilities" type="timg:Capabilities"/>
      <xs:element name="GetImagingSettings">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="VideoSourceToken" type="tt:ReferenceToken"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="GetImagingSettingsResponse">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="ImagingSettings" type="tt:ImagingSettings20"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="SetImagingSettings">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="VideoSourceToken" type="tt:ReferenceToken"/>
            <xs:element name="ImagingSettings" type="tt:ImagingSettings20"/>
            <xs:element name="ForcePersistence" type="xs:boolean" maxOccurs="1"
minOccurs="0"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="SetImagingSettingsResponse">
        <xs:complexType/>
      </xs:element>
      <xs:element name="GetOptions">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="VideoSourceToken" type="tt:ReferenceToken"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="GetOptionsResponse">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="ImagingOptions" type="tt:ImagingOptions20"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:schema>
  </wsdl:types>
</wsdl:definitions>
```

```

</xs:element>
<xs:element name="Move">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="VideoSourceToken" type="tt:ReferenceToken"/>
      <xs:element name="Focus" type="tt:FocusMove"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="MoveResponse">
  <xs:complexType>
    <xs:sequence/>
  </xs:complexType>
</xs:element>
<xs:element name="GetMoveOptions">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="VideoSourceToken" type="tt:ReferenceToken"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="GetMoveOptionsResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="MoveOptions" type="tt:MoveOptions20"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="Stop">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="VideoSourceToken" type="tt:ReferenceToken"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="StopResponse">
  <xs:complexType>
    <xs:sequence/>
  </xs:complexType>
</xs:element>
<xs:element name="GetStatus">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="VideoSourceToken" type="tt:ReferenceToken"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="GetStatusResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Status" type="tt:ImagingStatus20"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:simpleType name="ImagingPresetType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Custom"/>
    <xs:enumeration value="ClearWeather"/>
    <xs:enumeration value="Cloudy"/>
    <xs:enumeration value="Fog"/>
    <xs:enumeration value="Rain"/>
    <xs:enumeration value="Snowing"/>
    <xs:enumeration value="Snow"/>
    <xs:enumeration value="WDR"/>
    <xs:enumeration value="Shade"/>
    <xs:enumeration value="Night"/>
    <xs:enumeration value="Indoor"/>
    <xs:enumeration value="Fluorescent"/>
    <xs:enumeration value="Incandescent"/>
    <xs:enumeration value="Sodium (Natrium)"/>
    <xs:enumeration value="Sunrise (Horizon)"/>
    <xs:enumeration value="Sunset (Rear)"/>
    <xs:enumeration value="ExtremeHot"/>
    <xs:enumeration value="ExtremeCold"/>
    <xs:enumeration value="Underwater"/>
    <xs:enumeration value="CloseUp"/>
    <xs:enumeration value="Motion"/>
    <xs:enumeration value="FlickerFree50"/>
  </xs:restriction>
</xs:simpleType>

```

```

        <xs:enumeration value="FlickerFree60"/>
      </xs:restriction>
    </xs:simpleType>
    <xs:complexType name="ImagingPreset">
      <xs:sequence>
        <xs:element name="Name" type="tt:Name"/>
      </xs:sequence>
      <xs:attribute name="token" type="tt:ReferenceToken" use="required"/>
      <xs:attribute name="type" type="xs:string" use="required"/>
      <xs:anyAttribute processContents="lax"/>
    </xs:complexType>
    <xs:element name="GetPresets">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="VideoSourceToken" type="tt:ReferenceToken"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="GetPresetsResponse">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="Preset" type="timg:ImagingPreset"
maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="GetCurrentPreset">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="VideoSourceToken" type="tt:ReferenceToken"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="GetCurrentPresetResponse">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="Preset" type="timg:ImagingPreset" minOccurs="0"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="SetCurrentPreset">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="VideoSourceToken" type="tt:ReferenceToken"/>
          <xs:element name="PresetToken" type="tt:ReferenceToken"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="SetCurrentPresetResponse">
      <xs:complexType>
        <xs:sequence/>
      </xs:complexType>
    </xs:element>
  </xs:schema>
</wsdl:types>
<wsdl:message name="GetServiceCapabilitiesRequest">
  <wsdl:part name="parameters" element="timg:GetServiceCapabilities"/>
</wsdl:message>
<wsdl:message name="GetServiceCapabilitiesResponse">
  <wsdl:part name="parameters" element="timg:GetServiceCapabilitiesResponse"/>
</wsdl:message>
<wsdl:message name="GetImagingSettingsRequest">
  <wsdl:part name="parameters" element="timg:GetImagingSettings"/>
</wsdl:message>
<wsdl:message name="GetImagingSettingsResponse">
  <wsdl:part name="parameters" element="timg:GetImagingSettingsResponse"/>
</wsdl:message>
<wsdl:message name="SetImagingSettingsRequest">
  <wsdl:part name="parameters" element="timg:SetImagingSettings"/>
</wsdl:message>
<wsdl:message name="SetImagingSettingsResponse">
  <wsdl:part name="parameters" element="timg:SetImagingSettingsResponse"/>
</wsdl:message>
<wsdl:message name="GetOptionsRequest">
  <wsdl:part name="parameters" element="timg:GetOptions"/>
</wsdl:message>
<wsdl:message name="GetOptionsResponse">
  <wsdl:part name="parameters" element="timg:GetOptionsResponse"/>

```

```

</wsdl:message>
<wsdl:message name="MoveRequest">
  <wsdl:part name="parameters" element="timg:Move"/>
</wsdl:message>
<wsdl:message name="MoveResponse">
  <wsdl:part name="parameters" element="timg:MoveResponse"/>
</wsdl:message>
<wsdl:message name="GetMoveOptionsRequest">
  <wsdl:part name="parameters" element="timg:GetMoveOptions"/>
</wsdl:message>
<wsdl:message name="GetMoveOptionsResponse">
  <wsdl:part name="parameters" element="timg:GetMoveOptionsResponse"/>
</wsdl:message>
<wsdl:message name="StopRequest">
  <wsdl:part name="parameters" element="timg:Stop"/>
</wsdl:message>
<wsdl:message name="StopResponse">
  <wsdl:part name="parameters" element="timg:StopResponse"/>
</wsdl:message>
<wsdl:message name="GetStatusRequest">
  <wsdl:part name="parameters" element="timg:GetStatus"/>
</wsdl:message>
<wsdl:message name="GetStatusResponse">
  <wsdl:part name="parameters" element="timg:GetStatusResponse"/>
</wsdl:message>
<wsdl:message name="GetPresetsRequest">
  <wsdl:part name="parameters" element="timg:GetPresets"/>
</wsdl:message>
<wsdl:message name="GetPresetsResponse">
  <wsdl:part name="parameters" element="timg:GetPresetsResponse"/>
</wsdl:message>
<wsdl:message name="GetCurrentPresetRequest">
  <wsdl:part name="parameters" element="timg:GetCurrentPreset"/>
</wsdl:message>
<wsdl:message name="GetCurrentPresetResponse">
  <wsdl:part name="parameters" element="timg:GetCurrentPresetResponse"/>
</wsdl:message>
<wsdl:message name="SetCurrentPresetRequest">
  <wsdl:part name="parameters" element="timg:SetCurrentPreset"/>
</wsdl:message>
<wsdl:message name="SetCurrentPresetResponse">
  <wsdl:part name="parameters" element="timg:SetCurrentPresetResponse"/>
</wsdl:message>
<wsdl:portType name="ImagingPort">
  <wsdl:operation name="GetServiceCapabilities">
    <wsdl:input message="timg:GetServiceCapabilitiesRequest"/>
    <wsdl:output message="timg:GetServiceCapabilitiesResponse"/>
  </wsdl:operation>
  <wsdl:operation name="GetImagingSettings">
    <wsdl:input message="timg:GetImagingSettingsRequest"/>
    <wsdl:output message="timg:GetImagingSettingsResponse"/>
  </wsdl:operation>
  <wsdl:operation name="SetImagingSettings">
    <wsdl:input message="timg:SetImagingSettingsRequest"/>
    <wsdl:output message="timg:SetImagingSettingsResponse"/>
  </wsdl:operation>
  <wsdl:operation name="GetOptions">
    <wsdl:input message="timg:GetOptionsRequest"/>
    <wsdl:output message="timg:GetOptionsResponse"/>
  </wsdl:operation>
  <wsdl:operation name="Move">
    <wsdl:input message="timg:MoveRequest"/>
    <wsdl:output message="timg:MoveResponse"/>
  </wsdl:operation>
  <wsdl:operation name="GetMoveOptions">
    <wsdl:input message="timg:GetMoveOptionsRequest"/>
    <wsdl:output message="timg:GetMoveOptionsResponse"/>
  </wsdl:operation>
  <wsdl:operation name="Stop">
    <wsdl:input message="timg:StopRequest"/>
    <wsdl:output message="timg:StopResponse"/>
  </wsdl:operation>
  <wsdl:operation name="GetStatus">
    <wsdl:input message="timg:GetStatusRequest"/>
    <wsdl:output message="timg:GetStatusResponse"/>
  </wsdl:operation>
  <wsdl:operation name="GetPresets">
    <wsdl:input message="timg:GetPresetsRequest"/>
  </wsdl:operation>

```

```
<wsdl:output message="timg:GetPresetsResponse"/>
</wsdl:operation>
<wsdl:operation name="GetCurrentPreset">
  <wsdl:input message="timg:GetCurrentPresetRequest"/>
  <wsdl:output message="timg:GetCurrentPresetResponse"/>
</wsdl:operation>
<wsdl:operation name="SetCurrentPreset">
  <wsdl:input message="timg:SetCurrentPresetRequest"/>
  <wsdl:output message="timg:SetCurrentPresetResponse"/>
</wsdl:operation>
</wsdl:portType>
<wsdl:binding name="ImagingBinding" type="timg:ImagingPort">
  <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
  <wsdl:operation name="GetServiceCapabilities">
    <soap:operation
soapAction="http://www.onvif.org/ver20/imaging/wsdl/GetServiceCapabilities"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="GetImagingSettings">
    <soap:operation
soapAction="http://www.onvif.org/ver20/imaging/wsdl/GetImagingSettings"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="SetImagingSettings">
    <soap:operation
soapAction="http://www.onvif.org/ver20/imaging/wsdl/SetImagingSettings"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="GetOptions">
    <soap:operation
soapAction="http://www.onvif.org/ver20/imaging/wsdl/GetOptions"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="Move">
    <soap:operation soapAction="http://www.onvif.org/ver20/imaging/wsdl/Move"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="Stop">
    <soap:operation
soapAction="http://www.onvif.org/ver20/imaging/wsdl/FocusStop"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="GetStatus">
    <soap:operation
soapAction="http://www.onvif.org/ver20/imaging/wsdl/GetStatus"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
```

```
<wsdl:output>
  <soap:body use="literal"/>
</wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetMoveOptions">
  <soap:operation
soapAction="http://www.onvif.org/ver20/imaging/wsdl/GetMoveOptions"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetPresets">
  <soap:operation
soapAction="http://www.onvif.org/ver20/imaging/wsdl/GetPresets"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetCurrentPreset">
  <soap:operation
soapAction="http://www.onvif.org/ver20/imaging/wsdl/GetCurrentPreset"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="SetCurrentPreset">
  <soap:operation
soapAction="http://www.onvif.org/ver20/imaging/wsdl/SetCurrentPreset"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
</wsdl:binding>
</wsdl:definitions>
```

IECNORM.COM : Click to view the full PDF of IEC 62676-2-31:2019

F.3 Multimédia

```

<?xml version="1.0" encoding="utf-8"?><?xml-stylesheet type="text/xsl"
href="../../../ver20/util/onvif-wsdl-viewer.xsl"?><wsdl:definitions
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap12/"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:trt="http://www.onvif.org/ver10/media/wsdl"
targetNamespace="http://www.onvif.org/ver10/media/wsdl">
  <wsdl:types>
    <xs:schema xmlns:tt="http://www.onvif.org/ver10/schema"
targetNamespace="http://www.onvif.org/ver10/media/wsdl" elementFormDefault="qualified"
version="2.6">
      <xs:import namespace="http://www.onvif.org/ver10/schema"
schemaLocation="../../../ver10/schema/onvif.xsd"/>
      <xs:element name="GetServiceCapabilities">
        <xs:complexType>
          <xs:sequence/>
        </xs:complexType>
      </xs:element>
      <xs:element name="GetServiceCapabilitiesResponse">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Capabilities" type="trt:Capabilities"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="Capabilities">
        <xs:sequence>
          <xs:element name="ProfileCapabilities" type="trt:ProfileCapabilities"/>
          <xs:element name="StreamingCapabilities"
type="trt:StreamingCapabilities"/>
          <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="SnapshotUri" type="xs:boolean"/>
        <xs:attribute name="Rotation" type="xs:boolean"/>
        <xs:attribute name="VideoSourceMode" type="xs:boolean"/>
        <xs:attribute name="OSD" type="xs:boolean"/>
        <xs:attribute name="EXIFCompression" type="xs:boolean"/>
        <xs:anyAttribute processContents="lax"/>
      </xs:complexType>
      <xs:element name="Capabilities" type="trt:Capabilities"/>
      <xs:complexType name="ProfileCapabilities">
        <xs:sequence>
          <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="MaximumNumberOfProfiles" type="xs:int"/>
        <xs:anyAttribute processContents="lax"/>
      </xs:complexType>
      <xs:complexType name="StreamingCapabilities">
        <xs:sequence>
          <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="RTPMulticast" type="xs:boolean"/>
        <xs:attribute name="RTP_TCP" type="xs:boolean"/>
        <xs:attribute name="RTP_RTSP_TCP" type="xs:boolean"/>
        <xs:attribute name="NonAggregateControl" type="xs:boolean"/>
        <xs:attribute name="NoRTSPStreaming" type="xs:boolean"/>
        <xs:anyAttribute processContents="lax"/>
      </xs:complexType>
      <xs:element name="GetVideoSources">
        <xs:complexType>
          <xs:sequence/>
        </xs:complexType>
      </xs:element>
      <xs:element name="GetVideoSourcesResponse">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="VideoSources" type="tt:VideoSource" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="GetAudioSources">

```

```

        <xs:complexType>
            <xs:sequence/>
        </xs:complexType>
    </xs:element>
    <xs:element name="GetAudioSourcesResponse">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="AudioSources" type="tt:AudioSource" minOccurs="0"
maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="GetAudioOutputs">
        <xs:complexType>
            <xs:sequence/>
        </xs:complexType>
    </xs:element>
    <xs:element name="GetAudioOutputsResponse">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="AudioOutputs" type="tt:AudioOutput" minOccurs="0"
maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="CreateProfile">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="Name" type="tt:Name"/>
                <xs:element name="Token" type="tt:ReferenceToken" minOccurs="0"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="CreateProfileResponse">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="Profile" type="tt:Profile"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="GetProfile">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="ProfileToken" type="tt:ReferenceToken"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="GetProfileResponse">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="Profile" type="tt:Profile"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="GetProfiles">
        <xs:complexType>
            <xs:sequence/>
        </xs:complexType>
    </xs:element>
    <xs:element name="GetProfilesResponse">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="Profiles" type="tt:Profile" minOccurs="0"
maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="AddVideoEncoderConfiguration">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="ProfileToken" type="tt:ReferenceToken"/>
                <xs:element name="ConfigurationToken" type="tt:ReferenceToken"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="AddVideoEncoderConfigurationResponse">
        <xs:complexType>
            <xs:sequence/>

```

```
</xs:complexType>
</xs:element>
<xs:element name="RemoveVideoEncoderConfiguration">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="ProfileToken" type="tt:ReferenceToken"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="RemoveVideoEncoderConfigurationResponse">
  <xs:complexType>
    <xs:sequence/>
  </xs:complexType>
</xs:element>
<xs:element name="AddVideoSourceConfiguration">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="ProfileToken" type="tt:ReferenceToken"/>
      <xs:element name="ConfigurationToken" type="tt:ReferenceToken"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="AddVideoSourceConfigurationResponse">
  <xs:complexType>
    <xs:sequence/>
  </xs:complexType>
</xs:element>
<xs:element name="RemoveVideoSourceConfiguration">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="ProfileToken" type="tt:ReferenceToken"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="RemoveVideoSourceConfigurationResponse">
  <xs:complexType>
    <xs:sequence/>
  </xs:complexType>
</xs:element>
<xs:element name="AddAudioEncoderConfiguration">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="ProfileToken" type="tt:ReferenceToken"/>
      <xs:element name="ConfigurationToken" type="tt:ReferenceToken"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="AddAudioEncoderConfigurationResponse">
  <xs:complexType>
    <xs:sequence/>
  </xs:complexType>
</xs:element>
<xs:element name="RemoveAudioEncoderConfiguration">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="ProfileToken" type="tt:ReferenceToken"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="RemoveAudioEncoderConfigurationResponse">
  <xs:complexType>
    <xs:sequence/>
  </xs:complexType>
</xs:element>
<xs:element name="AddAudioSourceConfiguration">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="ProfileToken" type="tt:ReferenceToken"/>
      <xs:element name="ConfigurationToken" type="tt:ReferenceToken"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="AddAudioSourceConfigurationResponse">
  <xs:complexType>
    <xs:sequence/>
  </xs:complexType>
</xs:element>
<xs:element name="RemoveAudioSourceConfiguration">
```

```

<xs:complexType>
  <xs:sequence>
    <xs:element name="ProfileToken" type="tt:ReferenceToken"/>
  </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="RemoveAudioSourceConfigurationResponse">
  <xs:complexType>
    <xs:sequence/>
  </xs:complexType>
</xs:element>
<xs:element name="AddPTZConfiguration">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="ProfileToken" type="tt:ReferenceToken"/>
      <xs:element name="ConfigurationToken" type="tt:ReferenceToken"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="AddPTZConfigurationResponse">
  <xs:complexType>
    <xs:sequence/>
  </xs:complexType>
</xs:element>
<xs:element name="RemovePTZConfiguration">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="ProfileToken" type="tt:ReferenceToken"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="RemovePTZConfigurationResponse">
  <xs:complexType>
    <xs:sequence/>
  </xs:complexType>
</xs:element>
<xs:element name="AddVideoAnalyticsConfiguration">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="ProfileToken" type="tt:ReferenceToken"/>
      <xs:element name="ConfigurationToken" type="tt:ReferenceToken"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="AddVideoAnalyticsConfigurationResponse">
  <xs:complexType>
    <xs:sequence/>
  </xs:complexType>
</xs:element>
<xs:element name="RemoveVideoAnalyticsConfiguration">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="ProfileToken" type="tt:ReferenceToken"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="RemoveVideoAnalyticsConfigurationResponse">
  <xs:complexType>
    <xs:sequence/>
  </xs:complexType>
</xs:element>
<xs:element name="AddMetadataConfiguration">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="ProfileToken" type="tt:ReferenceToken"/>
      <xs:element name="ConfigurationToken" type="tt:ReferenceToken"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="AddMetadataConfigurationResponse">
  <xs:complexType>
    <xs:sequence/>
  </xs:complexType>
</xs:element>
<xs:element name="RemoveMetadataConfiguration">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="ProfileToken" type="tt:ReferenceToken"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

```
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="RemoveMetadataConfigurationResponse">
  <xs:complexType>
    <xs:sequence/>
  </xs:complexType>
</xs:element>
<xs:element name="AddAudioOutputConfiguration">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="ProfileToken" type="tt:ReferenceToken"/>
      <xs:element name="ConfigurationToken" type="tt:ReferenceToken"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="AddAudioOutputConfigurationResponse">
  <xs:complexType>
    <xs:sequence/>
  </xs:complexType>
</xs:element>
<xs:element name="RemoveAudioOutputConfiguration">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="ProfileToken" type="tt:ReferenceToken"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="RemoveAudioOutputConfigurationResponse">
  <xs:complexType>
    <xs:sequence/>
  </xs:complexType>
</xs:element>
<xs:element name="AddAudioDecoderConfiguration">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="ProfileToken" type="tt:ReferenceToken"/>
      <xs:element name="ConfigurationToken" type="tt:ReferenceToken"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="AddAudioDecoderConfigurationResponse">
  <xs:complexType>
    <xs:sequence/>
  </xs:complexType>
</xs:element>
<xs:element name="RemoveAudioDecoderConfiguration">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="ProfileToken" type="tt:ReferenceToken"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="RemoveAudioDecoderConfigurationResponse">
  <xs:complexType>
    <xs:sequence/>
  </xs:complexType>
</xs:element>
<xs:element name="DeleteProfile">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="ProfileToken" type="tt:ReferenceToken"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="DeleteProfileResponse">
  <xs:complexType>
    <xs:sequence/>
  </xs:complexType>
</xs:element>
<xs:element name="GetVideoEncoderConfigurations">
  <xs:complexType>
    <xs:sequence/>
  </xs:complexType>
</xs:element>
<xs:element name="GetVideoEncoderConfigurationsResponse">
  <xs:complexType>
    <xs:sequence>
```

```

        <xs:element name="Configurations" type="tt:VideoEncoderConfiguration"
minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="GetVideoSourceConfigurations">
    <xs:complexType>
        <xs:sequence/>
    </xs:complexType>
</xs:element>
<xs:element name="GetVideoSourceConfigurationsResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Configurations" type="tt:VideoSourceConfiguration"
minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="GetAudioEncoderConfigurations">
    <xs:complexType>
        <xs:sequence/>
    </xs:complexType>
</xs:element>
<xs:element name="GetAudioEncoderConfigurationsResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Configurations" type="tt:AudioEncoderConfiguration"
minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="GetAudioSourceConfigurations">
    <xs:complexType>
        <xs:sequence/>
    </xs:complexType>
</xs:element>
<xs:element name="GetAudioSourceConfigurationsResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Configurations" type="tt:AudioSourceConfiguration"
minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="GetVideoAnalyticsConfigurations">
    <xs:complexType>
        <xs:sequence/>
    </xs:complexType>
</xs:element>
<xs:element name="GetVideoAnalyticsConfigurationsResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Configurations"
type="tt:VideoAnalyticsConfiguration" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="GetMetadataConfigurations">
    <xs:complexType>
        <xs:sequence/>
    </xs:complexType>
</xs:element>
<xs:element name="GetMetadataConfigurationsResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Configurations" type="tt:MetadataConfiguration"
minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="GetAudioOutputConfigurations">
    <xs:complexType>
        <xs:sequence/>
    </xs:complexType>
</xs:element>
<xs:element name="GetAudioOutputConfigurationsResponse">
    <xs:complexType>
        <xs:sequence>

```

```

        <xs:element name="Configurations" type="tt:AudioOutputConfiguration"
minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="GetAudioDecoderConfigurations">
    <xs:complexType>
        <xs:sequence/>
    </xs:complexType>
</xs:element>
<xs:element name="GetAudioDecoderConfigurationsResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Configurations" type="tt:AudioDecoderConfiguration"
minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="GetVideoSourceConfiguration">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="ConfigurationToken" type="tt:ReferenceToken"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="GetVideoSourceConfigurationResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Configuration" type="tt:VideoSourceConfiguration"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="GetVideoEncoderConfiguration">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="ConfigurationToken" type="tt:ReferenceToken"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="GetVideoEncoderConfigurationResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Configuration"
type="tt:VideoEncoderConfiguration"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="GetAudioSourceConfiguration">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="ConfigurationToken" type="tt:ReferenceToken"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="GetAudioSourceConfigurationResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Configuration" type="tt:AudioSourceConfiguration"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="GetAudioEncoderConfiguration">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="ConfigurationToken" type="tt:ReferenceToken"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="GetAudioEncoderConfigurationResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Configuration"
type="tt:AudioEncoderConfiguration"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="GetVideoAnalyticsConfiguration">
    <xs:complexType>

```

```

        <xs:sequence>
            <xs:element name="ConfigurationToken" type="tt:ReferenceToken"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="GetVideoAnalyticsConfigurationResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Configuration"
type="tt:VideoAnalyticsConfiguration"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="GetMetadataConfiguration">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="ConfigurationToken" type="tt:ReferenceToken"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="GetMetadataConfigurationResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Configuration" type="tt:MetadataConfiguration"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="GetAudioOutputConfiguration">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="ConfigurationToken" type="tt:ReferenceToken"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="GetAudioOutputConfigurationResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Configuration" type="tt:AudioOutputConfiguration"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="GetAudioDecoderConfiguration">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="ConfigurationToken" type="tt:ReferenceToken"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="GetAudioDecoderConfigurationResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Configuration"
type="tt:AudioDecoderConfiguration"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="GetCompatibleVideoEncoderConfigurations">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="ProfileToken" type="tt:ReferenceToken"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="GetCompatibleVideoEncoderConfigurationsResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Configurations" type="tt:VideoEncoderConfiguration"
minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="GetCompatibleVideoSourceConfigurations">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="ProfileToken" type="tt:ReferenceToken"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>

```



```

<xs:element name="GetCompatibleVideoSourceConfigurationsResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Configurations" type="tt:VideoSourceConfiguration"
minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="GetCompatibleAudioEncoderConfigurations">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="ProfileToken" type="tt:ReferenceToken"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="GetCompatibleAudioEncoderConfigurationsResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Configurations" type="tt:AudioEncoderConfiguration"
minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="GetCompatibleAudioSourceConfigurations">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="ProfileToken" type="tt:ReferenceToken"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="GetCompatibleAudioSourceConfigurationsResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Configurations" type="tt:AudioSourceConfiguration"
minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="GetCompatibleVideoAnalyticsConfigurations">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="ProfileToken" type="tt:ReferenceToken"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="GetCompatibleVideoAnalyticsConfigurationsResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Configurations"
type="tt:VideoAnalyticsConfiguration" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="GetCompatibleMetadataConfigurations">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="ProfileToken" type="tt:ReferenceToken"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="GetCompatibleMetadataConfigurationsResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Configurations" type="tt:MetadataConfiguration"
minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="GetCompatibleAudioOutputConfigurations">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="ProfileToken" type="tt:ReferenceToken"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="GetCompatibleAudioOutputConfigurationsResponse">
  <xs:complexType>
    <xs:sequence>

```