

ONVIF™ Media Service Specification

Version 21.12

December, 2021



Copyright © 2008-2020 ONVIF™ All rights reserved.

Recipients of this document may copy, distribute, publish, or display this document so long as this copyright notice, license and disclaimer are retained with all copies of the document. No license is granted to modify this document.

THIS DOCUMENT IS PROVIDED "AS IS," AND THE CORPORATION AND ITS MEMBERS AND THEIR AFFILIATES, MAKE NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT, OR TITLE; THAT THE CONTENTS OF THIS DOCUMENT ARE SUITABLE FOR ANY PURPOSE; OR THAT THE IMPLEMENTATION OF SUCH CONTENTS WILL NOT INFRINGE ANY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.

IN NO EVENT WILL THE CORPORATION OR ITS MEMBERS OR THEIR AFFILIATES BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, PUNITIVE OR CONSEQUENTIAL DAMAGES, ARISING OUT OF OR RELATING TO ANY USE OR DISTRIBUTION OF THIS DOCUMENT, WHETHER OR NOT (1) THE CORPORATION, MEMBERS OR THEIR AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES, OR (2) SUCH DAMAGES WERE REASONABLY FORESEEABLE, AND ARISING OUT OF OR RELATING TO ANY USE OR DISTRIBUTION OF THIS DOCUMENT. THE FOREGOING DISCLAIMER AND LIMITATION ON LIABILITY DO NOT APPLY TO, INVALIDATE, OR LIMIT REPRESENTATIONS AND WARRANTIES MADE BY THE MEMBERS AND THEIR RESPECTIVE AFFILIATES TO THE CORPORATION AND OTHER MEMBERS IN CERTAIN WRITTEN POLICIES OF THE CORPORATION.

CONTENTS

1	Scope	6
2	Normative references	6
3	Terms and Definitions	6
3.1	Definitions	6
3.2	Abbreviations	7
4	Overview	7
4.1	Media profiles	7
4.2	Video source mode	10
5	Service	10
5.1	Audio and video codecs	11
5.2	Media Profile	11
5.2.1	CreateProfile	12
5.2.2	GetProfiles	12
5.2.3	GetProfile	13
5.2.4	AddVideoSourceConfiguration	13
5.2.5	AddVideoEncoderConfiguration	14
5.2.6	AddAudioSourceConfiguration	14
5.2.7	AddAudioEncoderConfiguration	15
5.2.8	AddPTZConfiguration	15
5.2.9	AddVideoAnalyticsConfiguration	16
5.2.10	AddMetadataConfiguration	17
5.2.11	AddAudioOutputConfiguration	17
5.2.12	AddAudioDecoderConfiguration	18
5.2.13	RemoveVideoSourceConfiguration	18
5.2.14	RemoveVideoEncoderConfiguration	19
5.2.15	RemoveAudioSourceConfiguration	19
5.2.16	RemoveAudioEncoderConfiguration	20
5.2.17	RemovePTZConfiguration	21
5.2.18	RemoveVideoAnalyticsConfiguration	21
5.2.19	RemoveMetadataConfiguration	22
5.2.20	RemoveAudioOutputConfiguration	22
5.2.21	RemoveAudioDecoderConfiguration	23
5.2.22	DeleteProfile	23
5.3	Video source	24
5.3.1	GetVideoSources	24
5.4	Video source configuration	24
5.4.1	GetVideoSourceConfigurations	24
5.4.2	GetVideoSourceConfiguration	25
5.4.3	GetCompatibleVideoSourceConfigurations	25
5.4.4	GetVideoSourceConfigurationOptions	25
5.4.5	SetVideoSourceConfiguration	26
5.5	Video encoder configuration	27
5.5.1	GetVideoEncoderConfigurations	27
5.5.2	GetVideoEncoderConfiguration	28
5.5.3	GetCompatibleVideoEncoderConfigurations	28
5.5.4	GetVideoEncoderConfigurationOptions	28
5.5.5	SetVideoEncoderConfiguration	29
5.5.6	GetGuaranteedNumberOfVideoEncoderInstances	30

5.6	Audio source	30
5.6.1	GetAudioSources	30
5.7	Audio source configuration	31
5.7.1	GetAudioSourceConfigurations	31
5.7.2	GetAudioSourceConfiguration	31
5.7.3	GetCompatibleAudioSourceConfigurations	32
5.7.4	GetAudioSourceConfigurationOptions	32
5.7.5	SetAudioSourceConfiguration	33
5.8	Audio encoder configuration	33
5.8.1	GetAudioEncoderConfigurations	34
5.8.2	GetAudioEncoderConfiguration	34
5.8.3	GetCompatibleAudioEncoderConfigurations	34
5.8.4	GetAudioEncoderConfigurationOptions	35
5.8.5	SetAudioEncoderConfiguration	36
5.9	Video analytics configuration	36
5.9.1	GetVideoAnalyticsConfigurations	37
5.9.2	GetVideoAnalyticsConfiguration	37
5.9.3	GetCompatibleVideoAnalyticsConfigurations	37
5.9.4	SetVideoAnalyticsConfiguration	38
5.10	Metadata configuration	39
5.10.1	Efficient XML Interchange (EXI)	39
5.10.2	GetMetadataConfigurations	40
5.10.3	GetMetadataConfiguration	40
5.10.4	GetCompatibleMetadataConfigurations	41
5.10.5	GetMetadataConfigurationOptions	41
5.10.6	SetMetadataConfiguration	42
5.11	Audio outputs	42
5.11.1	GetAudioOutputs	42
5.12	Audio output configuration	43
5.12.1	GetAudioOutputConfigurations	43
5.12.2	GetAudioOutputConfiguration	43
5.12.3	GetCompatibleAudioOutputConfigurations	44
5.12.4	GetAudioOutputConfigurationOptions	44
5.12.5	SetAudioOutputConfiguration	45
5.13	Audio decoder configuration	46
5.13.1	GetAudioDecoderConfigurations	46
5.13.2	GetAudioDecoderConfiguration	46
5.13.3	GetCompatibleAudioDecoderConfigurations	47
5.13.4	GetAudioDecoderConfigurationOptions	47
5.13.5	SetAudioDecoderConfiguration	48
5.14	Audio channel modes	48
5.15	Stream URI	49
5.15.1	GetStreamUri	49
5.16	Snapshot	50
5.16.1	GetSnapshotUri	50
5.17	Multicast	51
5.17.1	StartMulticastStreaming	51
5.17.2	StopMulticastStreaming	52
5.18	Synchronization Points	52
5.18.1	SetSynchronizationPoint	52
5.19	Video source mode	53

5.19.1	GetVideoSourceModes	53
5.19.2	SetVideoSourceMode	53
5.20	OSD (On-Screen Display)	54
5.20.1	CreateOSD	54
5.20.2	DeleteOSD	55
5.20.3	GetOSDs	55
5.20.4	GetOSD	56
5.20.5	SetOSD	56
5.20.6	GetOSDOptions	57
5.21	GetServiceCapabilities	57
5.22	Events	58
5.22.1	Configuration Change	58
5.22.1.1	Profile	58
5.22.1.2	VideoEncoderConfiguration	58
5.22.1.3	VideoSourceConfiguration	58
5.22.1.4	VideoOutputConfiguration	59
5.22.1.5	AudioEncoderConfiguration	59
5.22.1.6	AudioSourceConfiguration	59
5.22.1.7	AudioOutputConfiguration	59
5.22.1.8	MetadataConfiguration	60
5.22.1.9	PTZ Configuration	60
5.22.1.10	VideoAnalyticsConfiguration	60
5.22.2	Active Connections	60
5.22.3	Active Sessions	61
Annex A	Bibliography	63
Annex B	Revision History	64

1 Scope

Scope

This document defines the web service interface for configuration of the so called media profiles. These include the selection of Video and Audio inputs as well as PTZ and Analytics modes and the configuration of Video and Audio encoders.

Media streaming is out of scope of this document and covered by the ONVIF streaming specification.

Web service usage is outside of the scope of this document. Please refer to the ONVIF core specification.

2 Normative references

ONVIF Core Specification

<<http://www.onvif.org/onvif/specs/core/ONVIF-Core-Specification.pdf>>

ONVIF Media Service Specification

<<http://www.onvif.org/onvif/specs/srv/media/ONVIF-Media-Service-Spec.pdf>>

ONVIF Imaging Service Specification

<<http://www.onvif.org/onvif/specs/srv/img/ONVIF-Imaging-Service-Spec.pdf>>

ONVIF PTZ Service Specification

<<http://www.onvif.org/onvif/specs/srv/ptz/ONVIF-PTZ-Service-Spec.pdf>>

ONVIF Streaming Specification

<<http://www.onvif.org/onvif/specs/stream/ONVIF-Streaming-Spec.pdf>>

ONVIF Video Analytics Specification

<<http://www.onvif.org/onvif/specs/srv/analytics/ONVIF-VideoAnalytics-Service-Spec.pdf>>

W3C Efficient XML Interchange (EXI) Format 1.0

<<http://www.w3.org/TR/exi/>>

3 Terms and Definitions

3.1 Definitions

Configuration Entity	A network video device media abstract component that is used to produce a media stream on the network, i.e. video and/or audio stream.
Control Plane	Consists of Media control functions, such as device control, media configuration and PTZ commands.
Digital PTZ	Function that diminishes or crops an image to adjust the image position and ratio.
GZIP	GNU data format for lossless compression.
Media Plane	Consists of media stream, such as video, audio and metadata.
Media Profile	Maps a video or an audio source or an audio output to a video or an audio encoder, a audio decoder configuration and PTZ and analytics configurations.
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).
Video Analytics	Algorithms or programs used to analyze video data and to generate data describing object location and behaviour.

3.2 Abbreviations

RTCP	RTP Control Protocol
RTP	Realtime Transport Protocol
RTSP	Real Time Streaming Protocol
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
EXI	Efficient XML Interchange Format

4 Overview

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. The device provides media configuration through the media service. WSDL for this service is specified in <http://www.onvif.org/ver10/media/wsdl/media.wsdl>.

Table 1: Referenced namespaces (with prefix)

Prefix	Namespace URI
env	http://www.w3.org/2003/05/soap-envelope
ter	http://www.onvif.org/ver10/error
xs	http://www.w3.org/2001/XMLSchema
tt	http://www.onvif.org/ver10/schema
trt	http://www.onvif.org/ver10/media/wsdl
tns1	http://www.onvif.org/ver10/topics

4.1 Media profiles

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 might change dynamically though).

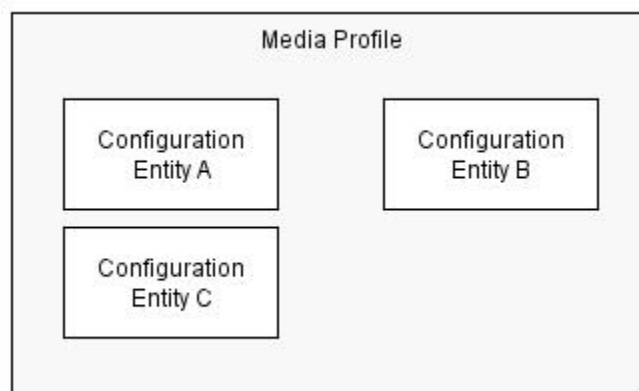


Figure 1: A media profile

A device having the media service provides at least one media profile at boot. 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.

A profile consists of a set of interconnected *configuration entities*. 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
- Video 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.

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

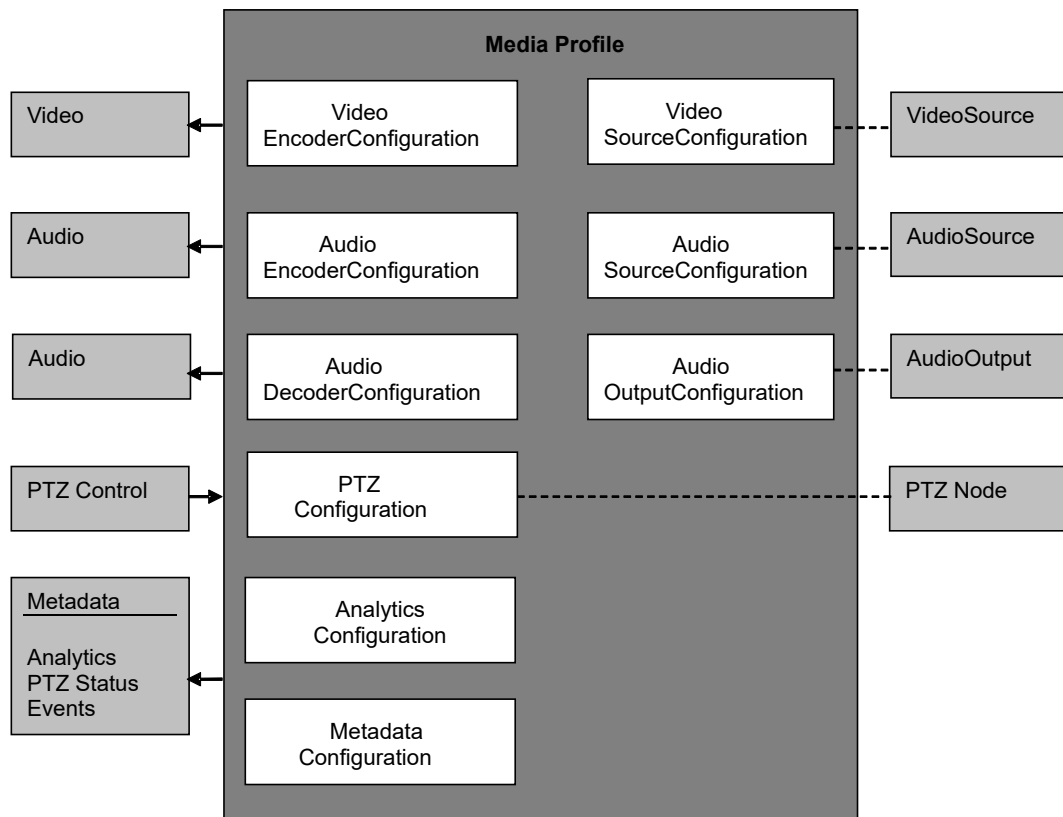


Figure 2: Figure 2: Complete profile configuration

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 list existing sources:

- *GetVideoSources* – Gets all existing video sources in the device.
- *GetAudioSources* – Gets all existing audio sources in the device.
- *GetAudioOutputs* – Gets all existing audio outputs in the device

The following commands manage Media Profiles:

- *CreateProfile* – Creates a new media profile.
- *GetProfiles* – Gets all existing media profiles.
- *GetProfile* – Gets a specific media profile.
- *DeleteProfile* – Deletes a specific media profile.
- *Add<configuration entity>* – Adds a specific configuration entity to the media profile.
- *Remove<configuration entity>* – Removes a specific configuration entity from a media profile.

The following commands manage Configuration Entities:

- *Get<configuration entity>Options* – Gets the valid property values for a specific configuration entity.

- *Set<configuration entity>* – Sets a configuration entity configuration.
- *Get<configuration entity>s* – Gets all existing configuration entities of the type.
- *Get<configuration entity>* – Gets a specific configuration entity.
- *GetCompatible<configuration entity>s* – Gets all configuration entities compatible with a specific media profile.

Where *<configuration 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 RTSP or HTTP stream 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 valid HTTP URI for a specific media profile that can be used to obtain a JPEG snapshot.

4.2 Video source mode

A device can have the capability for changing video source mode which is a setting of video source as exclusion in same time. For example, device's capability for max resolution (1920x1080@16:9 or 2048x1536@4:3) and frame rate (20fps or 30fps) can be changed by selecting each video source modes.

The following commands manage video source mode.

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

5 Service

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 Section the section called "Scope".

The media service commands are divided into two major categories:

- Media configuration:
 - Media profile commands
 - Video source commands
 - Vide 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 support. The detailed requirements are listed under the command descriptions.

5.1 Audio and video codecs

An ONVIF compliant device streams audio and video data using suitable encoding algorithms. The device may also be able to decode audio. A device supports any audio and video codecs, bitrates and resolution according to the manufacturer's choice. In order to ensure interoperability between client and device, this standard mandates the following codec profiles:

- An ONVIF compliant device shall support JPEG QVGA.
- An ONVIF compliant device shall support G.711 μ Law (Simplex-Camera Microphone Only, 1ch) [ITU-T G.711] if the device supports audio.

5.2 Media Profile

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 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
- Video 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 shall support at least one Analytics Configuration if Analytics service is supported. A device shall support at least one PTZ Configuration if PTZ service is supported.

5.2.1 CreateProfile

This operation creates a new empty media profile. The media profile shall be created in the device and shall be persistent (remain after reboot). A device 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.

Optionally the token identifier can be defined by the client. In this case a device shall support at least a token length of 12 characters and characters "A-Z" | "a-z" | "0-9" | "-".

REQUEST:

- **Name [tt:Name]**
Contains the friendly Name of the Profile to create as well as an optional Token parameter, specifying the unique identifier of the new media profile
- **Token optional [tt:ReferenceToken]**

RESPONSE:

- **Profile [tt:Profile]**
Returns an empty Profile structure with no configuration entities.

FAULTS:

- **env:Sender - ter:InvalidArgVal - ter:ProfileExists**
A profile with the token ProfileToken already exists.
- **env:Receiver - ter:Action - ter:MaxNVTPProfiles**
The maximum number of supported profiles supported by the device has been reached.

ACCESS CLASS:

ACTUATE

5.2.2 GetProfiles

Any endpoint can ask for the *existing* media profiles of a device using the GetProfiles command. Pre-configured or dynamically configured profiles can be retrieved using this command. This command lists *all* configured profiles in a device. The client does not need to know the media profile in order to use the command. The device shall support the retrieval of media profiles through the GetProfiles command.

A device shall include the “fixed” attribute in all the returned Profile elements.

REQUEST:

This is an empty message.

RESPONSE:

- **Profiles - optional, unbounded [tt:Profile]**
The response contains a 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:

No command specific faults defined.

ACCESS CLASS:

READ_MEDIA**5.2.3 GetProfile**

If the profile token is already known, a profile can be fetched through the GetProfile command. The device shall support the retrieval of a specific media profile through the GetProfile command.

A device shall include the “fixed” attribute in the returned Profile element.

REQUEST:

- **ProfileToken [tt:ReferenceToken]**
This message contains the token to the requested profile.

RESPONSE:

- **Profile [tt:Profile]**
The response contains the Profile indicated by the Token parameter. A 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 ProfileToken does not exist.

ACCESS CLASS:

READ_MEDIA**5.2.4 AddVideoSourceConfiguration**

This operation adds a VideoSourceConfiguration to an existing media profile. If such a configuration exists in the media profile, it will be replaced. The change shall be persistent. The device shall support addition of a video source configuration to a profile through the AddVideoSourceConfiguration command.

REQUEST:

- **ProfileToken [tt:ReferenceToken]**
Contains a reference to the VideoSourceConfiguration to add and the Profile where it shall be added.
- **ConfigurationToken [tt:ReferenceToken]**

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:NoConfig**
The VideoSourceConfiguration indicated by the 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

5.2.5 AddVideoEncoderConfiguration

This operation adds a VideoEncoderConfiguration to an existing media profile. If a configuration exists in the media profile, it will be replaced. The change shall be persistent. A device shall support addition of a video encoder configuration to a profile through the AddVideoEncoderConfiguration command.

A device shall support adding a compatible VideoEncoderconfiguration to a Profile containing a VideoSource-Configuration and shall support streaming video data of such a Profile.

REQUEST:

- **ProfileToken [tt:ReferenceToken]**
Contains a reference to the VideoEncoderConfiguration to add and the Profile where it shall be added.
- **ConfigurationToken [tt:ReferenceToken]**

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:NoConfig**
The VideoEncoderConfiguration indicated by the 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

5.2.6 AddAudioSourceConfiguration

This operation adds an AudioSourceConfiguration to an existing media profile. If a configuration exists in the media profile, it will be replaced. The change shall be persistent. A device that supports audio streaming from device to client shall support addition of audio source configuration to a profile through the AddAudioSource-Configuration command.

REQUEST:

- **ProfileToken [tt:ReferenceToken]**
Contains a reference to the AudioSourceConfiguration to add and the Profile where it shall be added.
- **ConfigurationToken [tt:ReferenceToken]**

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:NoConfig**
The AudioSourceConfiguration indicated by the 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.
- **env:Receiver - ter:ActionNotSupported - ter:AudioNotSupported**
Audio is not supported.

ACCESS CLASS:

ACTUATE

5.2.7 AddAudioEncoderConfiguration

This operation adds an AudioEncoderConfiguration to an existing media profile. If a configuration exists in the media profile, it will be replaced. The change shall be persistent. A device that supports audio streaming from device to client shall support addition of audio encoder configurations to a profile through the AddAudioEncoderConfiguration command.

A device shall support adding a compatible AudioEncoderConfiguration to a Profile containing an AudioSourceConfiguration and shall support streaming audio data of such a Profile.

REQUEST:

- **ProfileToken [tt:ReferenceToken]**
Contains a reference to the AudioEncoderConfiguration to add and the Profile where it shall be added.
- **ConfigurationToken [tt:ReferenceToken]**

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:NoConfig**
The AudioEncoderConfiguration indicated by the 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.
- **env:Receiver - ter:ActionNotSupported - ter:AudioNotSupported**
Audio is not supported.

ACCESS CLASS:

ACTUATE

5.2.8 AddPTZConfiguration

This operation adds a PTZConfiguration to an existing media profile. If a configuration exists in the media profile, it will be replaced. The change shall be persistent. A device that supports PTZ control shall support addition of PTZ configurations to a profile through the AddPTZConfiguration command.

Adding a PTZConfiguration to a media profile means that streams using that media profile can contain PTZ status (in the metadata), and that the media profile can be used for controlling PTZ movement, see document PTZ Service Specification.

REQUEST:

- **ProfileToken [tt:ReferenceToken]**
Contains a reference to the PTZConfiguration to add and the Profile where it shall be added.
- **ConfigurationToken [tt:ReferenceToken]**

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:NoConfig**
The PTZConfiguration indicated by the 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.
- **env:Receiver - ter:ActionNotSupported - ter:PTZNotSupported**
PTZ is not supported.

ACCESS CLASS:

ACTUATE**5.2.9 AddVideoAnalyticsConfiguration**

This operation adds a VideoAnalytics configuration to an existing media profile. If a configuration exists in the media profile, it will be replaced. The change shall be persistent. A device that supports video analytics shall support addition of video analytics configurations to a profile through the AddVideoAnalyticsConfiguration command.

Adding a VideoAnalyticsConfiguration to a media profile means that streams using that media profile can contain video analytics data (in the metadata) as defined by the submitted configuration reference. Video analytics data is specified in the document Video Analytics Specification and analytics configurations are managed through the commands defined in Section 5.9.

A profile containing only a video analytics configuration but no video source configuration is incomplete. Therefore, a client should first add a video source configuration to a profile before adding a video analytics configuration. The device can deny adding of a video analytics configuration before a video source configuration. In this case, it should respond with a ConfigurationConflict Fault.

REQUEST:

- **ProfileToken [tt:ReferenceToken]**
Contains a reference to the VideoAnalyticsConfiguration to add and the Profile where it shall be added.
- **ConfigurationToken [tt:ReferenceToken]**

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:NoConfig**
The VideoAnalyticsConfiguration indicated by the 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.
- **env:Receiver - ter:ActionNotSupported - ter:VideoAnalyticsNotSupported**
VideoAnalytics is not supported.

ACCESS CLASS:

ACTUATE

5.2.10 AddMetadataConfiguration

This operation adds a Metadata configuration to an existing media profile. If a configuration exists in the media profile, it will be replaced. The change shall be persistent. A device shall support the addition of a metadata configuration to a profile through the AddMetadataConfiguration command.

Adding a MetadataConfiguration to a Profile means that streams using that profile contain metadata. Metadata can consist of events, PTZ status, and/or video analytics data. Metadata configurations are handled through the commands defined in Section 5.10 and 5.9.4.

REQUEST:

- **ProfileToken [tt:ReferenceToken]**
Contains a reference to the MetadataConfiguration to add and the Profile where it shall be added.
- **ConfigurationToken [tt:ReferenceToken]**

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:NoConfig**
The MetadataConfiguration indicated by the 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

5.2.11 AddAudioOutputConfiguration

This operation adds an AudioOutputConfiguration to an existing media profile. If a configuration exists in the media profile, it will be replaced. The change shall be persistent. A device that signals support for Audio outputs via its Device IO AudioOutputs capability shall support the addition of an audio output configuration to a profile through the AddAudioOutputConfiguration command.

REQUEST:

- **ProfileToken [tt:ReferenceToken]**
Contains a reference to the AudioOutputConfiguration to add and the Profile where it shall be added.
- **ConfigurationToken [tt:ReferenceToken]**

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:NoConfig**
The AudioOutputConfiguration indicated by the 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.
- **env:Receiver - ter:ActionNotSupported - ter:AudioOutputNotSupported**
Audio or Audio Output is not supported

ACCESS CLASS:

ACTUATE**5.2.12 AddAudioDecoderConfiguration**

This operation adds an AudioDecoderConfiguration to an existing media profile. If a configuration exists in the media profile, it shall be replaced. The change shall be persistent. An device that signals support for Audio outputs via its Device IO AudioOutputs capability shall support the addition of an audio decoder configuration to a profile through the AddAudioDecoderConfiguration command.

REQUEST:

- **ProfileToken [tt:ReferenceToken]**
Contains a reference to the AudioConfiguration to add and the Profile where it shall be added.
- **ConfigurationToken [tt:ReferenceToken]**

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:NoConfig**
The AudioDecoderConfiguration indicated by the 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.
- **env:Receiver - ter:ActionNotSupported - ter:AudioDecodingNotSupported**
Audio or Audio Decoding is not supported

ACCESS CLASS:

ACTUATE**5.2.13 RemoveVideoSourceConfiguration**

This operation removes a VideoSourceConfiguration from an existing media profile. If the media profile does not contain a VideoSourceConfiguration, the operation has no effect. The removal shall be persistent. The device shall support removal of a video source configuration from a profile through the RemoveVideoSource-Configuration command.

Video source configurations should only be removed after removing a VideoEncoderConfiguration from the media profile.

REQUEST:

- **ProfileToken [tt:ReferenceToken]**
Contains a reference to the media profile from which the VideoSourceConfiguration shall be removed.

RESPONSE:

This is an empty message.

FAULTS:

- **env:Sender - ter:InvalidArgVal - ter:NoProfile**
The requested profile tokenProfileToken does not exist.
- **env:Sender - ter:InvalidArgVal - ter:NoConfig**
There exists no video source configuration in the media profile. Note: this fault code has become obsolete to respect the behaviour not to return this error
- **env:Receiver - ter:Action - ter:ConfigurationConflict**
Other configurations of the media profile are dependant on the VideoSourceConfiguration and removing it would cause a conflicting media profile.

ACCESS CLASS:

ACTUATE

5.2.14 RemoveVideoEncoderConfiguration

This operation removes a VideoEncoderConfiguration from an existing media profile. If the media profile does not contain a VideoEncoderConfiguration, the operation has no effect. The removal shall be persistent. The device shall support removal of a video encoder configuration from a profile through the RemoveVideoEncoderConfiguration command.

REQUEST:

- **ProfileToken [tt:ReferenceToken]**
Contains a reference to the media profile from which the VideoEncoderConfiguration shall be removed.

RESPONSE:

This is an empty message.

FAULTS:

- **env:Sender - ter:InvalidArgVal - ter:NoProfile**
The requested profile tokenProfileToken does not exist.
- **env:Sender - ter:InvalidArgVal - ter:NoConfig**
There exists no video encoder configuration in the media profile. Note: this fault code has become obsolete to respect the behaviour not to return this error
- **env:Receiver - ter:Action - ter:ConfigurationConflict**
Other configurations of the media profile are dependant on the VideoEncoderConfiguration and removing it would cause a conflicting media profile.

ACCESS CLASS:

ACTUATE

5.2.15 RemoveAudioSourceConfiguration

This operation removes an AudioSourceConfiguration from an existing media profile. If the media profile does not contain an AudioSourceConfiguration, the operation has no effect. The removal shall be persistent. A device that supports audio streaming from device to client shall support removal of an audio source configuration from a profile through the RemoveAudioSourceConfiguration command.

Audio source configurations should only be removed after removing an AudioEncoderConfiguration from the media profile.

REQUEST:

- **ProfileToken [tt:ReferenceToken]**
Contains a reference to the media profile from which the AudioSourceConfiguration shall be removed.

RESPONSE:

This is an empty message.

FAULTS:

- **env:Sender - ter:InvalidArgVal - ter:NoProfile**
The requested profile tokenProfileToken does not exist.
- **env:Sender - ter:InvalidArgVal - ter:NoConfig**
There exists no audio source configuration in the media profile. Note: this fault code has become obsolete to respect the behaviour not to return this error
- **env:Receiver - ter:Action - ter:ConfigurationConflict**
Other configurations of the media profile are dependant on the AudioSourceConfiguration and removing it would cause a conflicting media profile.
- **env:Receiver - ter:ActionNotSupported - ter:AudioNotSupported**
Audio is not supported.

ACCESS CLASS:

ACTUATE

5.2.16 RemoveAudioEncoderConfiguration

This operation removes an AudioEncoderConfiguration from an existing media profile. If the media profile does not contain an AudioEncoderConfiguration, the operation has no effect. The removal shall be persistent. A device that supports audio streaming from device to client shall support removal of audio encoder configurations from a profile through the RemoveAudioEncoderConfiguration command.

REQUEST:

- **ProfileToken [tt:ReferenceToken]**
Contains a reference to the media profile from which the AudioEncoderConfiguration shall be removed.

RESPONSE:

This is an empty message.

FAULTS:

- **env:Sender - ter:InvalidArgVal - ter:NoProfile**
The requested profile tokenProfileToken does not exist.
- **env:Sender - ter:InvalidArgVal - ter:NoConfig**
There exists no audio encoder configuration in the media profile. Note: this fault code has become obsolete to respect the behaviour not to return this error
- **env:Receiver - ter:Action - ter:ConfigurationConflict**
Other configurations of the media profile are dependant on the AudioEncoderConfiguration and removing it would cause a conflicting media profile.
- **env:Receiver - ter:ActionNotSupported - ter:AudioNotSupported**
Audio is not supported.

ACCESS CLASS:

ACTUATE

5.2.17 RemovePTZConfiguration

This operation removes a PTZConfiguration from an existing media profile. If the media profile does not contain a PTZConfiguration, the operation has no effect. The removal shall be persistent. A device that supports PTZ control shall support removal of PTZ configurations from a profile through the RemovePTZConfiguration command.

REQUEST:

- **ProfileToken [tt:ReferenceToken]**
Contains a reference to the media profile from which the PTZConfiguration shall be removed.

RESPONSE:

This is an empty message.

FAULTS:

- **env:Sender - ter:InvalidArgVal - ter:NoProfile**
The requested profile tokenProfileTokendoes not exist.
- **env:Sender - ter:InvalidArgVal - ter:NoConfig**
There exists no PTZ configuration in the media profile. Note: this fault code has become obsolete to respect the behaviour not to return this error
- **env:Receiver - ter:Action - ter:ConfigurationConflict**
Other configurations of the media profile are dependant on the PTZConfiguration and removing it would cause a conflicting media profile.
- **env:Receiver - ter:ActionNotSupported - ter:PTZNotSupported**
PTZ is not supported.

ACCESS CLASS:

ACTUATE

5.2.18 RemoveVideoAnalyticsConfiguration

This operation removes a VideoAnalyticsConfiguration from an existing media profile. If the media profile does not contain a VideoAnalyticsConfiguration, the operation has no effect. The removal shall be persistent. A device that supports video analytics shall support removal of a video analytics configuration from a profile through the RemoveVideoAnalyticsConfiguration command.

REQUEST:

- **ProfileToken [tt:ReferenceToken]**
Contains a reference to the media profile from which the VideoAnalyticsConfiguration shall be removed.

RESPONSE:

This is an empty message.

FAULTS:

- **env:Sender - ter:InvalidArgVal - ter:NoProfile**
The requested profile tokenProfileTokendoes not exist.
- **env:Sender - ter:InvalidArgVal - ter:NoConfig**
There exists no video analytics configuration in the media profile. Note: this fault code has become obsolete to respect the behaviour not to return this error

- **env:Receiver - ter:Action - ter:ConfigurationConflict**
Other configurations of the media profile are dependant on the VideoAnalyticsConfiguration and removing it would cause a conflicting media profile.
- **env:Receiver - ter:ActionNotSupported - ter:VideoAnalyticsNotSupported**
VideoAnalytics is not supported.

ACCESS CLASS:

ACTUATE

5.2.19 RemoveMetadataConfiguration

This operation removes a MetadataConfiguration from an existing media profile. If the media profile does not contain a MetadataConfiguration, the operation has no effect. The removal shall be persistent. A device shall support the removal of a metadata configuration from a profile through the RemoveMetadataConfiguration command.

REQUEST:

- **ProfileToken [tt:ReferenceToken]**
Contains a reference to the media profile from which the MetadataConfiguration shall be removed.

RESPONSE:

This is an empty message.

FAULTS:

- **env:Sender - ter:InvalidArgVal - ter:NoProfile**
The requested profile tokenProfileTokendoes not exist.
- **env:Sender - ter:InvalidArgVal - ter:NoConfig**
There exists no metadata configuration in the media profile. Note: this fault code has become obsolete to respect the behaviour not to return this error
- **env:Receiver - ter:Action - ter:ConfigurationConflict**
Other configurations of the media profile are dependant on the MetadataConfiguration and removing it would cause a conflicting media profile.

ACCESS CLASS:

ACTUATE

5.2.20 RemoveAudioOutputConfiguration

This operation removes an AudioOutputConfiguration from an existing media profile. If the media profile does not contain an AudioOutputConfiguration, the operation has no effect. The removal shall be persistent. An device that signals support for Audio outputs via its Device IO AudioOutputs capability shall support the removal of an audio output configuration from a profile through the RemoveAudioOutputConfiguration command.

REQUEST:

- **ProfileToken [tt:ReferenceToken]**
Contains a reference to the media profile from which the AudioOutputConfiguration shall be removed.

RESPONSE:

This is an empty message.

FAULTS:

- **env:Sender - ter:InvalidArgVal - ter:NoProfile**
The requested profile tokenProfileTokendoes not exist.

- **env:Sender - ter:InvalidArgVal - ter:NoConfig**
There exists no audio output configuration in the media profile. Note: this fault code has become obsolete to respect the behaviour not to return this error
- **env:Receiver - ter:Action - ter:ConfigurationConflict**
Other configurations of the media profile are dependant on the AudioOutputConfiguration and removing it would cause a conflicting media profile.
- **env: Receiver - ter:ActionNotSupported - ter:AudioOutputNotSupported**
Audio or Audio output is not supported

ACCESS CLASS:

ACTUATE

5.2.21 RemoveAudioDecoderConfiguration

This operation removes an AudioDecoderConfiguration from an existing media profile. If the media profile does not contain an AudioDecoderConfiguration, the operation has no effect. The removal shall be persistent. An device that signals support for Audio outputs via its Device IO AudioOutputs capability shall support the removal of an audio decoder configuration from a profile through the RemoveAudioDecoderConfiguration command.

REQUEST:

- **ProfileToken [tt:ReferenceToken]**
Contains a reference to the media profile from which the AudioDecoderConfiguration shall be removed.

RESPONSE:

This is an empty message.

FAULTS:

- **env:Sender - ter:InvalidArgVal - ter:NoProfile**
The requested profile tokenProfileTokendoes not exist.
- **env:Sender - ter:InvalidArgVal - ter:NoConfig**
There exists no audio decoder configuration in the media profile. Note: this fault code has become obsolete to respect the behaviour not to return this error
- **env:Receiver - ter:Action - ter:ConfigurationConflict**
Other configurations of the media profile are dependant on the AudioDecoder Configuration and removing it would cause a conflicting media profile.
- **env: Receiver - ter:ActionNotSupported - ter::AudioDecodingNotSupported**
Audio or AudioDecoding is not supported

ACCESS CLASS:

ACTUATE

5.2.22 DeleteProfile

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

REQUEST:

- **ProfileToken [tt:ReferenceToken]**
Contains a ProfileToken that indicates what media profile to delete.

RESPONSE:

This is an empty message.

FAULTS:

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

ACCESS CLASS:

ACTUATE**5.3 Video source**

A VideoSource represents unencoded video input. The structure contains the pixel resolution of the video, 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. See the Imaging Service Specification for more information.

5.3.1 GetVideoSources

This operation lists all available video sources for the device. The device shall support the listing of available video sources through the GetVideoSources command.

REQUEST:

This is an empty message.

RESPONSE:

- **VideoSources - optional, unbounded [tt:VideoSource]**
Contains a list of structures describing all available video sources of the device.

FAULTS:

No command specific faults defined.

ACCESS CLASS:

READ_MEDIA**5.4 Video source configuration**

A VideoSourceConfiguration contains a reference to a VideoSource and a Bounds structure containing either the whole VideoSource pixel area or a sub-portion of it. The Bounds and VideoSource define the image that is streamed to a client. If a VideoSourceConfiguration is used inside a profile its UseCount parameter is increased to indicate that changing this configuration could affect other users.

5.4.1 GetVideoSourceConfigurations

This operation lists all *existing* video source configurations for a device. This command lists *all* video source configurations in a device. The client need not know anything about the video source configurations in order to use the command. The device shall support the listing of available video source configurations through the GetVideoSourceConfigurations command.

REQUEST:

This is an empty message.

RESPONSE:

- **Configurations - optional, unbounded [tt:VideoSourceConfiguration]**
This message contains a list of all existing video source configurations in the device. A video source configuration does always point at a real video source with the SourceToken element.

FAULTS:

No command specific faults defined.

ACCESS CLASS:

READ_MEDIA

5.4.2 GetVideoSourceConfiguration

If the video source configuration token is already known, the video source configuration can be fetched through the GetVideoSourceConfiguration command. The device shall support retrieval of specific video source configurations through the GetVideoSourceConfiguration command.

REQUEST:

- **ConfigurationToken [tt:ReferenceToken]**
This message contains the token of the requested video source configuration.

RESPONSE:

- **Configuration [tt:VideoSourceConfiguration]**
This message contains the requested VideoSourceConfiguration with the matching token. A video source configuration does always point at a real video source with the SourceToken element.

FAULTS:

- **env:Sender - ter:InvalidArgVal - ter:NoConfig**
The requested configuration indicated with ConfigurationToken does not exist.

ACCESS CLASS:

READ_MEDIA

5.4.3 GetCompatibleVideoSourceConfigurations

This operation requests all the video source configurations of the device that are compatible with a certain media profile. Each of the returned configurations shall be a valid input parameter for the AddVideoSourceConfiguration command on the media profile. The result will vary depending on the capabilities, configurations and settings in the device. The device shall support the listing of compatible (with a specific profile) video source configurations through the GetCompatibleVideoSourceConfigurations command.

REQUEST:

- **ProfileToken [tt:ReferenceToken]**
Contains the token of an existing media profile.

RESPONSE:

- **Configurations - optional, unbounded [tt:VideoSourceConfiguration]**
Contains a list of video source configurations that are compatible with the media profile.

FAULTS:

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

ACCESS CLASS:

READ_MEDIA

5.4.4 GetVideoSourceConfigurationOptions

This operation returns the available parameters and their valid ranges to the client. Any combination of the parameters obtained using a given media profile and video source configuration shall be a valid input for the

SetVideoSourceConfiguration command. The device shall support the GetVideoSourceConfigurationOptions command.

If a video source 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 video source 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]**
This message may contain a media profile or video source configuration token, or both.
- **ProfileToken optional [tt:ReferenceToken]**

RESPONSE:

- **Options [tt:VideoSourceConfigurationOptions]**
This message contains the video configuration options. If a video source configuration is specified, the options shall concern that particular configuration. If a media profile is specified, the options shall be compatible with that media profile. If no tokens are specified, the options shall be considered generic for the device.

FAULTS:

- **env:Sender - ter:InvalidArgVal - ter:NoProfile**
The requested profile tokenProfileToken does not exist.
- **env:Sender - ter:InvalidArgVal - ter:NoConfig**
The requested configuration does not exist.

ACCESS CLASS:

READ_MEDIA

5.4.5 SetVideoSourceConfiguration

This operation modifies a video source configuration. The ForcePersistence flag indicates if the changes shall remain after reboot of the device. Running streams using this configuration may be immediately updated according to the new settings. The changes are not guaranteed to take effect unless the client requests a new stream URI and restarts any affected stream. Client methods for changing a running stream are out of scope for this specification. The device shall support the modification of video source parameters through the SetVideoSourceConfiguration command.

REQUEST:

- **Configuration [tt:VideoSourceConfiguration]**
The Configuration element contains the modified video source configuration. The configuration shall exist in the device. The ForcePersistence element is obsolete and should always be assumed to be true.
- **ForcePersistence [xs:boolean]**

RESPONSE:

This is an empty message.

FAULTS:

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

- **env:Sender - ter:InvalidArgVal - ter:ConfigModify**
The configuration parameters are not possible to set.
- **env:Receiver - ter:Action - ter:ConfigurationConflict**
The new settings conflicts with other uses of the configuration.

ACCESS CLASS:

ACTUATE

5.5 Video encoder configuration

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

- Encoder – The encoding used for the video data.
- 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 FrameRateLimit [fps] parameter to configure the output framerate.
- MPEG4/H264 specifics – Defines the encoding profile and GOV length [frame].

TheVideoEncoderConfiguration structure also contains multicast parameters and a session timeout to define video streaming behaviour. If a VideoEncoderConfiguration is used inside a profile its UseCount parameter is increased to indicate that changing this configuration could affect other users.

If the whole RateControl parameter structure is missing the current state of rate control is undefined and vendor specific. A device, supporting disabling rate control mechanisms shall reflect that by omitting the RateControl element after removal by a client 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.

5.5.1 GetVideoEncoderConfigurations

This operation lists all *existing* video encoder configurations of a device. This command lists *all* configured video encoder configurations in a device. The client does not need to know anything apriori about the video encoder configurations in order to use the command. The device shall support the listing of available video encoder configurations through the GetVideoEncoderConfigurations command.

REQUEST:

This is an empty message.

RESPONSE:

- **Configurations - optional, unbounded [tt:VideoEncoderConfiguration]**
This message contains a list of all existing video encoder configurations in the device.

FAULTS:

No command specific faults defined.

ACCESS CLASS:

READ_MEDIA

5.5.2 GetVideoEncoderConfiguration

If the video encoder configuration token is already known, the encoder configuration can be fetched through the GetVideoEncoderConfiguration command. The device shall support the retrieval of a specific video encoder configuration through the GetVideoEncoderConfiguration command.

REQUEST:

- **ConfigurationToken [tt:ReferenceToken]**
This message contains the token of the requested video encoder configuration.

RESPONSE:

- **Configuration [tt:VideoEncoderConfiguration]**
This message contains the requested VideoEncoderConfiguration with the matching token.

FAULTS:

- **env:Sender - ter:InvalidArgVal - ter:NoConfig**
The requested configuration indicated with ConfigurationToken does not exist.

ACCESS CLASS:

READ_MEDIA

5.5.3 GetCompatibleVideoEncoderConfigurations

This operation lists all the video encoder configurations of the device that are compatible with a certain media profile. Each of the returned configurations shall be a valid input parameter for the AddVideoEncoderConfiguration command on the media profile. The result will vary depending on the capabilities, configurations and settings in the device. The device shall support the listing of compatible (with a specific profile) video encoder configurations through the GetCompatibleVideoEncoderConfigurations command.

REQUEST:

- **ProfileToken [tt:ReferenceToken]**
Contains the token of an existing media profile.

RESPONSE:

- **Configurations - optional, unbounded [tt:VideoEncoderConfiguration]**
Contains a list of video encoder configurations that are compatible with the given media profile.

FAULTS:

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

ACCESS CLASS:

READ_MEDIA

5.5.4 GetVideoEncoderConfigurationOptions

This operation returns the available parameters and their valid ranges to the client. Any combination of the parameters obtained using a given media profile and video encoder configuration shall be a valid input for the SetVideoEncoderConfiguration command. The device shall support the GetVideoEncoderConfigurationOptions command.

If a video encoder 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 video encoder 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]**
This message may contain a media profile or video encoder configuration token, or both.
- **ProfileToken optional [tt:ReferenceToken]**

RESPONSE:

- **Options [tt:VideoEncoderConfigurationOptions]**
This message contains the video configuration options.

FAULTS:

- **env:Sender - ter:InvalidArgVal - ter:NoProfile**
The requested profile tokenProfileToken does not exist.
- **env:Sender - ter:InvalidArgVal - ter:NoConfig**
The requested configuration does not exist.

ACCESS CLASS:

READ_MEDIA

5.5.5 SetVideoEncoderConfiguration

This operation modifies a video encoder configuration. The ForcePersistence flag indicates if the changes shall remain after reboot of the device. Changes in the Multicast settings shall always be persistent. Running streams using this configuration may be immediately updated according to the new settings, but the changes are not guaranteed to take effect unless the client requests a new stream URI and restarts any affected stream. If the new settings invalidate any parameters already negotiated using RTSP, for example by changing codec type, the device must not apply these settings to existing streams. Instead it must either continue to stream using the old settings or stop sending data on the affected streams.

Client methods for changing a running stream are out of scope for this specification. The device shall support the modification of video encoder parameters through the SetVideoEncoderConfiguration command.

A device shall accept any combination of parameters that it returned in the GetVideoEncoderConfigurationOptionsResponse. 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.

REQUEST:

- **Configuration [tt:VideoEncoderConfiguration]**
The Configuration element contains the modified video encoder configuration. The configuration shall exist in the device. The ForcePersistence element is obsolete and should always be assumed to be true.
- **ForcePersistence [xs:boolean]**

RESPONSE:

This is an empty message.

FAULTS:

- **env:Sender - ter:InvalidArgVal - ter:NoConfig**
The configuration does not exist.
- **env:Sender - ter:InvalidArgVal - ter:ConfigModify**
The configuration parameters are not possible to set.

- **env:Receiver - ter:Action - ter:ConfigurationConflict**
The new settings conflicts with other uses of the configuration.

ACCESS CLASS:

ACTUATE

5.5.6 GetGuaranteedNumberOfVideoEncoderInstances

The GetGuaranteedNumberOfVideoEncoderInstances command can be used to request the minimum number of guaranteed video encoder instances (applications) per Video Source Configuration. A device SHALL support this command. This command was added in ONVIF 1.02.

REQUEST:

- **tt:ReferenceToken ConfigurationToken [1][1]**
This request contains a token to the video source configuration.

RESPONSE:

- **TotalNumber [xs:int]**
This message contains the minimum guaranteed TotalNumber of encoder instances (applications) per VideoSourceConfiguration. If a device limits the number of instances for respective Video Codecs the response contains the information how many Jpeg, H264 and Mpeg4 can be set up at the same time. In all other cases the device is able to deliver the TotalNumber of streams independent from the configured VideoCodec at the same time.
- **JPEG optional [xs:int]**
- **H264 optional [xs:int]**
- **MPEG4 optional [xs:int]**

FAULTS:

- **env:Sender - ter:InvalidArgVal - ter:NoConfig**
The requested configuration indicated with ConfigurationTokendoes not exist.

ACCESS CLASS:

READ_MEDIA

5.6 Audio source

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

5.6.1 GetAudioSources

This operation lists all available audio sources of the device. A device that supports audio streaming from device to client shall support listing of available audio sources through the GetAudioSources command.

REQUEST:

This is an empty message.

RESPONSE:

- **AudioSources - optional, unbounded [tt:AudioSource]**
Contains a list of structures describing all available audio sources of the device.

FAULTS:

- **env:Receiver - ter:ActionNotSupported - ter:AudioNotSupported**
The device does not support audio.

ACCESS CLASS:

READ_MEDIA

5.7 Audio source configuration

An AudioSourceConfiguration contains a reference to an AudioSource that is to be used for input in a media profile. If an AudioSourceConfiguration is used inside a profile its UseCount parameter is increased to indicate that changing this configuration could affect other users.

5.7.1 GetAudioSourceConfigurations

This operation lists all *existing* audio source configurations of a device. This command lists *all* audio source configurations in a device. The client does not need to know anything apriori about the audio source configurations in order to use the command. A device that supports audio streaming from device to client shall support listing of available audio source configurations through the GetAudioSourceConfigurations command.

REQUEST:

This is an empty message.

RESPONSE:

- **Configurations - optional, unbounded [tt:AudioSourceConfiguration]**
This message contains a list of all existing audio source configurations in the device. An audio source configuration does always point at a real audio source with the SourceToken element.

FAULTS:

- **env:Receiver - ter:ActionNotSupported - ter:AudioNotSupported**
The device does not support audio.

ACCESS CLASS:

READ_MEDIA

5.7.2 GetAudioSourceConfiguration

The GetAudioSourceConfiguration command fetches the audio source configurations if the audio source configuration token is already known. A device that supports audio streaming from device to client shall support the retrieval of a specific audio source configuration through the GetAudioSourceConfiguration command.

REQUEST:

- **ConfigurationToken [tt:ReferenceToken]**
This message contains the token of the requested audio source configuration. An audio source configuration does always point at a real audio source with the SourceToken element.

RESPONSE:

- **Configuration [tt:AudioSourceConfiguration]**
This message contains the requested AudioSourceConfiguration with the matching token.

FAULTS:

- **env:Sender - ter:InvalidArgVal - ter:NoConfig**
The requested configuration indicated with ConfigurationToken does not exist.
- **env:Receiver - ter:ActionNotSupported - ter:AudioNotSupported**
The device does not support audio.

ACCESS CLASS:

READ_MEDIA

5.7.3 GetCompatibleAudioSourceConfigurations

This operation requests all audio source configurations of a device that are compatible with a certain media profile. Each of the returned configurations shall be a valid input parameter for the AddAudioSourceConfiguration command on the media profile. The result varies depending on the capabilities, configurations and settings in the device. A device that supports audio streaming from device to client shall support listing of compatible (with a specific profile) audio source configurations through the GetCompatibleAudioSourceConfigurations command.

REQUEST:

- **ProfileToken [tt:ReferenceToken]**
Contains the token of an existing media profile.

RESPONSE:

- **Configurations - optional, unbounded [tt:AudioSourceConfiguration]**
Contains a list of audio source configurations that are compatible with the media profile.

FAULTS:

- **env:Sender - ter:InvalidArgVal - ter:NoProfile**
The requested profile tokenProfileTokendoes not exist.
- **env:Receiver - ter:ActionNotSupported - ter:AudioNotSupported**
The device does not support audio.

ACCESS CLASS:

READ_MEDIA

5.7.4 GetAudioSourceConfigurationOptions

This operation returns the available parameters and their valid ranges to the client. Any combination of the parameters obtained using a given media profile and audio source configuration shall be a valid input for the SetAudioSourceConfiguration command. A device that supports audio streaming from device to client shall support the GetAudioSourceConfigurationOptions command.

If an audio source 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 an audio source 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]**
This message may contain a media profile or audio source configuration token, or both.
- **ProfileToken optional [tt:ReferenceToken]**

RESPONSE:

- **Options [tt:AudioSourceConfigurationOptions]**
This message contains the audio configuration options. If an audio source configuration is specified, the options shall concern that particular configuration. If a media profile is specified, the options shall be compatible with that media profile. If no tokens are specified, the options shall be considered generic for the device.

FAULTS:

- **env:Sender - ter:InvalidArgVal - ter:NoProfile**
The requested profile tokenProfileTokendoes not exist.

- **env:Sender - ter:InvalidArgVal - ter:NoConfig**
The requested configuration does not exist.
- **env:Receiver - ter:ActionNotSupported - ter:AudioNotSupported**
The device does not support audio.

ACCESS CLASS:

READ_MEDIA

5.7.5 SetAudioSourceConfiguration

This operation modifies an audio source configuration. The ForcePersistence flag indicates if the changes shall remain after reboot of the device. Running streams using this configuration may be immediately updated according to the new settings, but the changes are not guaranteed to take effect unless the client requests a new stream URI and restarts any affected stream. If the new settings invalidate any parameters already negotiated using RTSP, for example by changing codec type, the device must not apply these settings to existing streams. Instead it must either continue to stream using the old settings or stop sending data on the affected streams.

Client methods for changing a running stream are out of scope for this specification. A device that supports audio streaming from device to client shall support the configuration of audio source parameters through the SetAudioSourceConfiguration command.

REQUEST:

- **Configuration [tt:AudioSourceConfiguration]**
The Configuration element contains the modified audio source configuration. The configuration shall exist in the device. The ForcePersistence element is obsolete and should always assumed to be true.
- **ForcePersistence [xs:boolean]**

RESPONSE:

This is an empty message.

FAULTS:

- **env:Sender - ter:InvalidArgVal - ter:NoConfig**
The configuration does not exist.
- **env:Sender - ter:InvalidArgVal - ter:ConfigModify**
The configuration parameters are not possible to set.
- **env:Receiver - ter:Action - ter:ConfigurationConflict**
The new settings conflicts with other uses of the configuration.
- **env:Receiver - ter:ActionNotSupported - ter:AudioNotSupported**
The device does not support audio.

ACCESS CLASS:

ACTUATE

5.8 Audio encoder configuration

An AudioEncoderConfiguration contains the following parameters for encoding audio data:

- Encoder – The encoding used for audio data.
- Bitrate – The output bitrate [kbps].
- SampleRate – The output sample rate [kHz].

The AudioEncoderConfiguration structure also contains multicast parameters and a session timeout to define audio streaming behaviour.

If an AudioEncoderConfiguration is used inside a profile its UseCount parameter is increased to indicate that changing this configuration could affect other users.

5.8.1 GetAudioEncoderConfigurations

This operation lists all *existing* device audio encoder configurations. The client does not need to know anything a priori about the audio encoder configurations in order to use the command. A device that supports audio streaming from device to client shall support the listing of available audio encoder configurations through the GetAudioEncoderConfigurations command.

REQUEST:

This is an empty message.

RESPONSE:

- **Configurations - optional, unbounded [tt:AudioEncoderConfiguration]**
This message contains a list of all existing audio encoder configurations in the device.

FAULTS:

- **env:Receiver - ter:ActionNotSupported - ter:AudioNotSupported**
The device does not support audio.

ACCESS CLASS:

READ_MEDIA

5.8.2 GetAudioEncoderConfiguration

The GetAudioEncoderConfiguration command fetches the encoder configuration if the audio encoder configuration token is known. A device that supports audio streaming from device to client shall support the listing of a specific audio encoder configuration through the GetAudioEncoderConfiguration command.

REQUEST:

- **ConfigurationToken [tt:ReferenceToken]**
This message contains the token of the requested audio encoder configuration.

RESPONSE:

- **Configuration [tt:AudioEncoderConfiguration]**
This message contains the requested AudioEncoderConfiguration with the matching token.

FAULTS:

- **env:Sender - ter:InvalidArgVal - ter:NoConfig**
The configuration does not exist.
- **env:Receiver - ter:ActionNotSupported - ter:AudioNotSupported**
The device does not support audio.

ACCESS CLASS:

READ_MEDIA

5.8.3 GetCompatibleAudioEncoderConfigurations

This operation requests all audio encoder configurations of the device that are compatible with a certain media profile. Each of the returned configurations shall be a valid input parameter for the AddAudioEncoderConfigura-

tion command on the media profile. The result varies depending on the capabilities, configurations and settings in the device. A device that supports audio streaming from device to client shall support listing of compatible (with a specific profile) audio encoder configurations through the `GetCompatibleAudioEncoderConfigurations` command.

REQUEST:

- **ProfileToken [tt:ReferenceToken]**
Contains the token of an existing media profile.

RESPONSE:

- **Configurations - optional, unbounded [tt:AudioEncoderConfiguration]**
Contains a list of audio encoder configurations that are compatible with the given media profile.

FAULTS:

- **env:Sender - ter:InvalidArgVal - ter:NoProfile**
The requested profile token `ProfileToken` does not exist.
- **env:Receiver - ter:ActionNotSupported - ter:AudioNotSupported**
The device does not support audio.

ACCESS CLASS:

READ_MEDIA

5.8.4 GetAudioEncoderConfigurationOptions

This operation returns the available parameters and their valid ranges to the client. Any combination of the parameters obtained using a given media profile and audio encoder configuration shall be a valid input for the `SetAudioEncoderConfiguration` command. A device that supports audio streaming from device to client shall support the `GetAudioEncoderConfigurationOptions` command.

If an audio encoder 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 an audio encoder 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]**
This message may contain a media profile or audio encoder configuration token, or both.
- **ProfileToken optional [tt:ReferenceToken]**

RESPONSE:

- **Options [tt:AudioEncoderConfigurationOptions]**
This message contains the audio configuration options. If a audio encoder configuration is specified, the options shall concern that particular configuration. If a media profile is specified, the options shall be compatible with that media profile. If no tokens are specified, the options shall be considered generic for the device.

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.

- **env:Receiver - ter:ActionNotSupported - ter:AudioNotSupported**
The device does not support audio.

ACCESS CLASS:

READ_MEDIA

5.8.5 SetAudioEncoderConfiguration

This operation modifies an audio encoder configuration. The ForcePersistence flag indicates if the changes shall remain after reboot of the device. Changes in the Multicast settings shall always be persistent. Running streams using this configuration may be immediately updated according to the new settings. The changes are not guaranteed to take effect unless the client requests a new stream URI and restarts any affected streams. Client methods for changing a running stream are out of scope for this specification. A device that supports audio streaming from device to client shall support the configuration of audio encoder parameters through the SetAudioEncoderConfiguration command.

REQUEST:

- **Configuration [tt:AudioEncoderConfiguration]**
The Configuration element contains the modified audio encoder configuration. The configuration shall exist in the device. The ForcePersistence element is obsolete and should always assumed to be true.
- **ForcePersistence [xs:boolean]**

RESPONSE:

This is an empty message.

FAULTS:

- **env:Sender - ter:InvalidArgVal - ter:NoConfig**
The configuration does not exist.
- **env:Sender - ter:InvalidArgVal - ter:ConfigModify**
The configuration parameters are not possible to set.
- **env:Receiver - ter:Action - ter:ConfigurationConflict**
The new settings conflicts with other uses of the configuration.
- **env:Receiver - ter:ActionNotSupported - ter:AudioNotSupported**
The device does not support audio.

ACCESS CLASS:

ACTUATE

5.9 Video analytics configuration

VideoAnalyticsConfiguration contains parameters for an *analytics engine* and a *rule engine* (see the document Video Analytics Service Specification). Thereby, the analytics engine consists of multiple modules which can be managed by the analytics module part of the analytics service. Similarly, the rule engine consists of multiple rules which can be managed by the rule engine part of the analytics service. The subsequent commands are introduced to handle complete video analytics configuration in an atomar way. For instance, the ModifyVideoAnalyticsConfiguration command changes analytics and rule engine configuration in an atomar operation. When a video analytics configuration is present in a profile, the metadata configuration can activate the streaming of the scene description within the RTP streams (see Section 5.10).

A device MAY NOT allow referencing the very same VideoAnalyticsConfiguration 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 MAY require separate processing of the whole video analytics for each VideoSourceConfiguration, even if they refer to the very same VideoSource.

Since the options of a VideoAnalyticsConfiguration are dynamic and often vendor specific, they can only be retrieved via the video analytics service.

5.9.1 GetVideoAnalyticsConfigurations

This operation lists all video analytics configurations of a device. This command lists *all* configured video analytics in a device. The client does not need to know anything apriori about the video analytics in order to use the command. A device that supports video analytics shall support the listing of available video analytics configuration through the GetVideoAnalyticsConfigurations command.

REQUEST:

This is an empty message.

RESPONSE:

- **Configurations - optional, unbounded [tt:VideoAnalyticsConfiguration]**
This message contains a list of all existing video analytics configurations in the device.

FAULTS:

- **env:Sender - ter:ActionNotSupported - ter:VideoAnalyticsNot-Supported**
Device does not support video analytics.

ACCESS CLASS:

READ_MEDIA

5.9.2 GetVideoAnalyticsConfiguration

The GetVideoAnalyticsConfiguration command fetches the video analytics configuration if the video analytics token is known. A device that supports video analytics shall support the listing of a specific video analytics configuration through the GetVideoAnalyticsConfiguration command.

REQUEST:

- **ConfigurationToken [tt:ReferenceToken]**
This message contains the token of an existing video analytics configuration.

RESPONSE:

- **Configuration [tt:VideoAnalyticsConfiguration]**
This message contains the requested video analytics configuration.

FAULTS:

- **env:Sender - ter:InvalidArgVal - ter:NoConfig**
The requested configuration indicated with ConfigurationTokendoes not exist.
- **env:Sender - ter:ActionNotSupported - ter:VideoAnalyticsNot-Supported**
The device does not support video analytics.

ACCESS CLASS:

READ_MEDIA

5.9.3 GetCompatibleVideoAnalyticsConfigurations

This operation requests all video analytic configurations of the device that are compatible with a certain media profile. Each of the returned configurations shall be a valid input parameter for the AddVideoAnalyticsConfigu-

ration command on the media profile. The result varies depending on the capabilities, configurations and settings in the device. A device that supports video analytics shall support the listing of compatible (with a specific profile) video analytics configuration through the GetCompatibleVideoAnalyticsConfigurations command.

REQUEST:

- **ProfileToken [tt:ReferenceToken]**
Contains the token of an existing media profile.

RESPONSE:

- **Configurations - optional, unbounded [tt:VideoAnalyticsConfiguration]**
Contains a list of video analytics configurations that are compatible with the given media profile.

FAULTS:

- **env:Sender - ter:InvalidArgVal - ter:NoProfile**
The requested profile tokenProfileToken does not exist.
- **env:Sender - ter:ActionNotSupported - ter:VideoAnalyticsNot-Supported**
The device does not support video analytics.

ACCESS CLASS:

READ_MEDIA

5.9.4 SetVideoAnalyticsConfiguration

A video analytics configuration is modified using this command. The ForcePersistence flag indicates if the changes shall remain after reboot of the device or not. Running streams using this configuration shall be immediately updated according to the new settings. Otherwise inconsistencies can occur between the scene description processed by the rule engine and the notifications produced by analytics engine and rule engine which reference the very same video analytics configuration token. A device that supports video analytics shall support the configuration of video analytics parameters through the SetVideoAnalyticsConfiguration command.

REQUEST:

- **Configuration [tt:VideoAnalyticsConfiguration]**
The Configuration element contains the modified video analytics configuration. The configuration shall exist in the device. The ForcePersistence element is obsolete and should always assumed to be true.
- **ForcePersistence [xs:boolean]**

RESPONSE:

This is an empty message.

FAULTS:

- **env:Sender - ter:InvalidArgVal - ter:NoConfig**
The configuration does not exist.
- **env:Sender - ter:InvalidArgVal - ter:ConfigModify**
The configuration parameters are not possible to set.
- **env:Receiver - ter:Action - ter:ConfigurationConflict**
The new settings conflicts with other uses of the configuration.
- **env:Sender - ter:ActionNotSupported - ter:VideoAnalyticsNot-Supported**
The device does not support video analytics.

ACCESS CLASS:

ACTUATE

5.10 Metadata configuration

For PTZ transmission of status and position change information can be enabled separately.

Event streaming can be enabled and controlled using topic filters. For topic filter configuration refer to section “Event Handling” of the ONVIF Core Specification.

Streaming of scene description can be enabled. Optionally the AnalyticsEngineConfiguration allows to restrict streaming of scene description to the provided list of AnalyticsModules. Note that analytics modules only generate scene description if they are configured in the AnalyticsConfiguration of the profile as defined in section 5.9.

A device shall ignore any analytics module parameters passed to the SetMetadataConfiguration command and should not list AnalyticsModule/Parameters.

The structure also contains multicast parameters used to configure and control multicast of the metadata stream. A session timeout parameter defines the session timeout (see ONVIF Streaming Specification)

If a MetadataConfiguration is used inside a profile its UseCount parameter is increased to indicate that changing this configuration could affect other users.

Devices supporting compressed metadata shall signal available compression algorithm as defined in the MetadataCompressionType. Currently defined compression types are "GZIP" and "EXI".

5.10.1 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 2 and Table 3) shall be supported by a devices signalling the support for EXI compression via GetMetadataConfigurationOptions.

The EXI header shall only be transmitted if a setting different then 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 2) shall always be used.

Table 2: Table 51 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 3
Padding Bits	If present must be “0”.

Table 3: 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)

Exi Option	Value
schemaID	Insert reference to schema obtained from device here.
datatypeRepresentationMap [http://www.w3.org/TR/2013/PER-exi-20131022/#key-datatypeRepresentationOption]	none
blockSize	default (1,000,000)
valueMaxLength [http://www.w3.org/TR/2013/PER-exi-20131022/#key-valueMaxLengthOption]	default (unbounded)
valuePartitionCapacity [http://www.w3.org/TR/2013/PER-exi-20131022/#key-valuePartitionCapacityOption]	default (unbounded)
user defined meta-data [http://www.w3.org/TR/2013/PER-exi-20131022/#key-userMetaOption]	none

5.10.2 GetMetadataConfigurations

This operation lists all *existing* metadata configurations. The client does not need to know anything apriori about the metadata in order to use the command. A device or another device that supports metadata streaming shall support the listing of existing metadata configurations through the GetMetadataConfigurations command.

REQUEST:

This is an empty message.

RESPONSE:

- **Configurations - optional, unbounded [tt:MetadataConfiguration]**
This message contains a list of all existing metadata configurations in the device.

FAULTS:

No command specific faults defined.

ACCESS CLASS:

READ_MEDIA

5.10.3 GetMetadataConfiguration

The GetMetadataConfiguration command fetches the metadata configuration if the metadata token is known. A device or another device that supports metadata streaming shall support the listing of a specific metadata configuration through the GetMetadataConfiguration command.

REQUEST:

- **ConfigurationToken [tt:ReferenceToken]**
This message contains the token of an existing metadata configuration.

RESPONSE:

- **Configuration [tt:MetadataConfiguration]**
This message contains the requested metadata configuration.

FAULTS:

- **env:Sender - ter:InvalidArgVal - ter:NoConfig**
The requested configuration indicated with ConfigurationToken does not exist.

ACCESS CLASS:

READ_MEDIA

5.10.4 GetCompatibleMetadataConfigurations

This operation requests all the metadata configurations of the device that are compatible with a certain media profile. Each of the returned configurations shall be a valid input parameter for the AddMetadataConfiguration command on the media profile. The result varies depending on the capabilities, configurations and settings in the device. A device or other device that supports metadata streaming shall support the listing of compatible (with a specific profile) metadata configuration through the GetCompatibleMetadataConfigurations command.

REQUEST:

- **ProfileToken [tt:ReferenceToken]**
Contains the token of an existing media profile.

RESPONSE:

- **Configurations - optional, unbounded [tt:MetadataConfiguration]**
Contains a list of metadata configurations that are compatible with the given media profile.

FAULTS:

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

ACCESS CLASS:

READ_MEDIA

5.10.5 GetMetadataConfigurationOptions

This operation returns the available parameters and their valid ranges to the client. Any combination of the parameters obtained using a given media profile and metadata configuration shall be a valid input for the SetMetadataConfiguration command. A device that supports metadata streaming shall support the GetMetadataConfigurationOptions command.

If a metadata 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 metadata 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]**
This message may contain a media profile or metadata configuration token, or both.
- **ProfileToken optional [tt:ReferenceToken]**

RESPONSE:

- **Options [tt:MetadataConfigurationOptions]**
This message contains the metadata configuration options. If a metadata configuration is specified, the options shall concern that particular configuration. If a media profile is specified, the options shall be

compatible with that media profile. If no tokens are specified, the options shall be considered generic for the device.

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.

ACCESS CLASS:

READ_MEDIA

5.10.6 SetMetadataConfiguration

This operation modifies a metadata configuration. The ForcePersistence flag indicates if the changes shall remain after reboot of the device. Changes in the Multicast settings shall always be persistent. Running streams using this configuration may be updated immediately according to the new settings. The changes are not guaranteed to take effect unless the client requests a new stream URI and restarts any affected streams. Client methods for changing a running stream are out of scope for this specification. A device or another device that supports metadata streaming shall support the configuration of metadata parameters through the SetMetadataConfiguration command.

REQUEST:

- **Configuration [tt:MetadataConfiguration]**
The Configuration element contains multicast settings as well as a set of filters determining what data to include in the metadata stream. The ForcePersistence element is obsolete and should always be assumed to be true.
- **ForcePersistence [xs:boolean]**

RESPONSE:

This is an empty message.

FAULTS:

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

5.11 Audio outputs

The Audio Output represents the physical audio outputs that can be connected to a loudspeaker.

5.11.1 GetAudioOutputs

This command lists all available audio outputs of a device. An device that signals support for Audio outputs via its Device IO AudioOutputs capability shall support listing of available audio outputs through the GetAudioOutputs command.

REQUEST:

This is an empty message.

RESPONSE:

- **AudioOutputs - optional, unbounded [tt:AudioOutput]**
Contains a list of structures describing all available audio outputs of the device. If a device has no AudioOutputs an empty list is returned.

FAULTS:

- **env:Receiver - ter:ActionNotSupported - ter:AudioOutputNotSupported**
Audio or Audio Outputs are not supported by the device

ACCESS CLASS:

READ_MEDIA

5.12 Audio output configuration

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 Section 5.14).

If an AudioOutputConfiguration is used inside a profile its UseCount parameter is increased to indicate that changing this configuration could affect other users.

5.12.1 GetAudioOutputConfigurations

This command lists all existing AudioOutputConfigurations of a device. The client does not need to know anything a priori about the audio configurations to use this command. A device that signals support for Audio outputs via its Device IO AudioOutputs capability shall support the listing of AudioOutputConfigurations through this command.

REQUEST:

This is an empty message.

RESPONSE:

- **Configurations - optional, unbounded [tt:AudioOutputConfiguration]**
Contains a list of AudioOutputConfigurations that are available on the device

FAULTS:

- **env: Receiver - ter:ActionNotSupported - ter:AudioOutputNotSupported**
Audio or Audio Outputs are not supported by the device

ACCESS CLASS:

READ_MEDIA

5.12.2 GetAudioOutputConfiguration

If the audio output configuration token is already known, the output configuration can be fetched through the GetAudioOutputConfiguration command. An device that signals support for Audio outputs via its Device IO

AudioOutputs capability shall support the retrieval of a specific audio output configuration through the GetAudioOutputConfiguration command.

REQUEST:

- **ConfigurationToken [tt:ReferenceToken]**
This message contains the token of the requested AudioOutput configuration.

RESPONSE:

- **Configuration [tt:AudioOutputConfiguration]**
This message contains the requested AudioOutputConfiguration with the matching token.

FAULTS:

- **env:Sender - ter:InvalidArgVal - ter:NoConfig**
The requested configuration indicated with ConfigurationToken does not exist.
- **env:Receiver - ter:ActionNotSupported - ter::AudioOutputNotSupported**
Audio or Audio Outputs are not supported by the device

ACCESS CLASS:

READ_MEDIA

5.12.3 GetCompatibleAudioOutputConfigurations

This command lists all audio output configurations of a device that are compatible with a certain media profile. Each returned configuration shall be a valid input for the AddAudioOutputConfiguration command. A device that signals support for Audio outputs via its Device IO AudioOutputs capability shall support the listing of compatible (with a specific profile) AudioOutputConfigurations through the GetCompatibleAudioOutputConfigurations command.

REQUEST:

- **ProfileToken [tt:ReferenceToken]**
Contains the token of an existing media profile.

RESPONSE:

- **Configurations - optional, unbounded [tt:AudioOutputConfiguration]**
Contains a list of audio output configurations that are compatible with the given media profile.

FAULTS:

- **env:Sender - ter:InvalidArgVal - ter:NoProfile**
The requested profile token ProfileToken does not exist.
- **env:Receiver - ter:ActionNotSupported - ter:AudioOutputNotSupported**
Audio or Audio Outputs are not supported by the device

ACCESS CLASS:

READ_MEDIA

5.12.4 GetAudioOutputConfigurationOptions

This operation returns the available parameters and their valid ranges to the client. Any combination of the parameters obtained using a given media profile and audio output configuration shall be a valid input for the SetAudioOutputConfiguration command. A device that supports audio streaming from client to device shall support the GetAudioOutputConfigurationOptions command.

If an audio output 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 an audio output 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]**
This message may contain a media profile or audio output configuration token, or both.
- **ProfileToken optional [tt:ReferenceToken]**

RESPONSE:

- **Options [tt:AudioOutputConfigurationOptions]**
This message contains the audio output configuration options. If a audio output configuration is specified, the options shall concern that particular configuration. If a media profile is specified, the options shall be compatible with that media profile. If no tokens are specified, the options shall be considered generic for the device.

FAULTS:

- **env:Sender - ter:InvalidArgVal - ter:NoProfile**
The requested profile tokenProfileToken does not exist.
- **env:Sender - ter:InvalidArgVal - ter:NoConfig**
The requested configuration does not exist.
- **env:Receiver - ter:ActionNotSupported - ter:AudioOutputNotSupported**
Audio or Audio Outputs are not supported by the device

ACCESS CLASS:

READ_MEDIA

5.12.5 SetAudioOutputConfiguration

This operation modifies an audio output configuration. The ForcePersistence flag indicates if the changes shall remain after reboot of the device. An device that signals support for Audio outputs via its Device IO AudioOutputs capability shall support the modification of audio output parameters through the SetAudioOutputConfiguration command.

REQUEST:

- **Configuration [tt:AudioOutputConfiguration]**
The Configuration element contains the modified Audio Output configuration. The configuration must exist in the device. The ForcePersistence element is obsolete and should always assumed to be true.
- **ForcePersistence [xs:boolean]**

RESPONSE:

This is an empty message.

FAULTS:

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

- **env: Receiver - ter:ActionNotSupported - ter:AudioOutputNotSupported**
Audio or Audio Outputs are not supported by the device

ACCESS CLASS:

ACTUATE

5.13 Audio decoder configuration

The Audio Decoder Configuration does not contain any parameter to configure the decoding .A decoder shall decode every data it receives (according to its capabilities).

If an AudioDecoderConfiguration is used inside a profile its UseCount parameter is increased to indicate that changing this configuration could affect other users.

5.13.1 GetAudioDecoderConfigurations

This command lists all existing AudioDecoderConfigurations of a device.

The client does not need to know anything apriori about the audio decoder configurations in order to use this command. An device that signals support for Audio outputs via its Device IO AudioOutputs capability shall support the listing of AudioOutputConfigurations through this command.

REQUEST:

This is an empty message.

RESPONSE:

- **Configurations - optional, unbounded [tt:AudioDecoderConfiguration]**
Contains a list of AudioDecoderConfigurations that are available on the device

FAULTS:

- **env:Receiver - ter:ActionNotSupported - ter:AudioDecodingNotSupported**
Audio or Audio decoding is not supported by the device

ACCESS CLASS:

READ_MEDIA

5.13.2 GetAudioDecoderConfiguration

If the audio decoder configuration token is already known, the decoder configuration can be fetched through the GetAudioDecoderConfiguration command. An device that signals support for Audio outputs via its Device IO AudioOutputs capability shall support the retrieval of a specific audio decoder configuration through the GetAudioDecoderConfiguration command.

REQUEST:

- **ConfigurationToken [tt:ReferenceToken]**
This message contains the token of the requested AudioDecoder configuration.

RESPONSE:

- **Configuration [tt:AudioDecoderConfiguration]**
This message contains the requested AudioDecoder Configuration with the matching token.

FAULTS:

- **env:Sender - ter:InvalidArgVal - ter:NoConfig**
The requested configuration indicated with ConfigurationToken does not exist.

- **env:Receiver - ter:ActionNotSupported - ter:AudioDecodingNotSupported**
Audio or Audio decoding is not supported by the device

ACCESS CLASS:

READ_MEDIA

5.13.3 GetCompatibleAudioDecoderConfigurations

This operation lists all the audio decoder configurations of the device that are compatible with a certain media profile. Each of the returned configurations shall be a valid input parameter for the AddAudioDecoderConfiguration command on the media profile. A device that signals support for Audio outputs via its Device IO AudioOutputs capability shall support the listing of compatible (with a specific profile) audio decoder configurations through the GetCompatibleAudioDecoderConfigurations command.

REQUEST:

- **ProfileToken [tt:ReferenceToken]**
Contains the token of an existing media profile.

RESPONSE:

- **Configurations - optional, unbounded [tt:AudioDecoderConfiguration]**
Contains a list of audiodecoder configurations that are compatible with the given media profile.

FAULTS:

- **env:Sender - ter:InvalidArgVal - ter:NoProfile**
The requested profile token ProfileToken does not exist.
- **env:Receiver - ter:ActionNotSupported - ter:AudioDecodingNotSupported**
Audio or Audio decoding is not supported by the device

ACCESS CLASS:

READ_MEDIA

5.13.4 GetAudioDecoderConfigurationOptions

This operation returns the available parameters and their valid ranges to the client. Any combination of the parameters obtained using a given media profile and audio decoder configuration shall be a valid input for the SetAudioDecoderConfiguration command. A device that supports audio streaming from client to device shall support the GetAudioDecoderConfigurationOptions command.

If an audio decoder 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 an audio decoder 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]**
This message may contain a media profile or audio decoder configuration token, or both.
- **ProfileToken optional [tt:ReferenceToken]**

RESPONSE:

- **Options [tt:AudioDecoderConfigurationOptions]**
This message contains the audio decoder configuration options. If a audio decoder configuration is specified, the options shall concern that particular configuration. If a media profile is specified, the

options shall be compatible with that media profile. If no tokens are specified, the options shall be considered generic for the device.

FAULTS:

- **env:Sender - ter:InvalidArgVal - ter:NoProfile**
The requested profile tokenProfileToken does not exist.
- **env:Sender - ter:InvalidArgVal - ter:NoConfig**
The requested configuration does not exist.
- **env:Receiver - ter:ActionNotSupported - ter:AudioDecodingNotSupported**
Audio or Audio decoding is not supported by the device

ACCESS CLASS:**READ_MEDIA****5.13.5 SetAudioDecoderConfiguration**

This operation modifies an audio decoder configuration. The ForcePersistence flag indicates if the changes shall remain after reboot of the device. An device that signals support for Audio outputs via its Device IO AudioOutputs capability shall support the modification of audio decoder parameters through the SetAudioDecoderConfiguration command.

REQUEST:

- **Configuration [tt:AudioDecoderConfiguration]**
The Configuration element contains the modified AudioDecoder configuration. The configuration must exist in the device. The ForcePersistence element is obsolete and should always assumed to be true.
- **ForcePersistence [xs:boolean]**

RESPONSE:

This is an empty message.

FAULTS:

- **env:Sender - ter:InvalidArgVal - ter:NoConfig**
The configuration does not exist.
- **env:Sender - ter:InvalidArgVal - ter:ConfigModify**
The configuration parameters are not possible to set.
- **env:Receiver - ter:Action - ter:ConfigurationConflict**
The new settings conflicts with other uses of the configuration.
- **env: Receiver - ter:ActionNotSupported - ter:AudioDecodingNotSupported**
Audio or Audio decoding is not supported by the device

ACCESS CLASS:**ACTUATE****5.14 Audio channel modes**

An audio channel MAY support different types of audio transmission. While 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 the 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 out of ONVIF scope.

5.15 Stream URI

5.15.1 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 profile is changed. The `InvalidAfterConnect`, `InvalidAfterReboot` and `Timeout` Parameter should be set accordingly (`InvalidAfterConnect=false`, `InvalidAfterReboot=false`, `timeout=PT0S`). A device shall support the retrieval of a media stream URI for a specific media profile through the `GetStreamUri` command unless the `NoRTSPStreaming` capability is set.

The correct syntax for the `StreamSetup` element for the media stream setups as defined in 5.1.1 of the ONVIF Streaming Specification are defined in Table 4.

Table 4: Valid setup parameter combinations

Mode	StreamType	Transport Protocol
RTP unicast over UDP	RTP_unicast	UDP
RTP over RTSP over HTTP over TCP	RTP_unicast	HTTP
RTP over RTSP over TCP	RTP_unicast	RTSP

If a multicast stream is requested at least one of `VideoEncoderConfiguration`, `AudioEncoderConfiguration` and `MetadataConfiguration` shall have a valid multicast setting.

For full compatibility with other ONVIF services a device should not generate `Uris` longer than 128 octets.

On a request for transport protocol `http` a device shall return a url that uses the same port as the web service. This enables seamless NAT traversal.

REQUEST:

- **StreamSetup [tt:StreamSetup]**
The `StreamSetup` element contains two parts. `StreamType` defines if a unicast or multicast media stream is requested. `Transport` specifies a chain of transport protocols defining the tunnelling of the media stream over different network protocols.
- **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]**
Contains the stable `Uri` to be used for requesting the media stream as well as parameters defining the lifetime of the `Uri`. The `InvalidAfterConnect` and `InvalidAfterReboot` parameter shall be set to `false`, the `timeout` parameter shall be set to `PT0S` to indicate that this stream `Uri` is indefinitely valid even if the profile changes.

- **InvalidAfterConnect [xs:boolean]**
The parameter shall be set to false.
- **InvalidAfterReboot [xs:boolean]**
- **Timeout [xs:duration]**
The parameter shall be set to false.
The timeout parameter shall be set to PT0S to indicate that this stream URI is indefinitely valid even if the profile changes.

FAULTS:

- **env:Sender - ter:InvalidArgVal - ter:NoProfile**
The media profile does not exist.
- **env:Sender - ter:InvalidArgVal - ter:InvalidStreamSetup**
Specification of StreamType or Transport part in StreamSetup is not supported.
- **env:Sender - ter:OperationProhibited - ter:StreamConflict**
Specification of StreamType or Transport part in StreamSetup causes conflict with other streams.
- **env:Receiver - ter:Action - ter:IncompleteConfiguration**
The specified media profile does not have the minimum amount of configurations to have streams. Please add at least one source configuration and one matching encoder configuration.
- **env:Sender - ter:InvalidArgVal - ter:InvalidMulticastSettings**
No configuration is configured for multicast.

ACCESS CLASS:

READ_MEDIA

5.16 Snapshot

5.16.1 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 is changed. The ValidUntilConnect, ValidUntilReboot and Timeout Parameter shall be set accordingly (ValidUntilConnect=false, ValidUntilReboot=false, timeout=PT0S). The URI can be used for acquiring a JPEG image through a HTTP GET operation.

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

A device supporting the media service should support this command. 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 source and dimensions of the snapshot.

RESPONSE:

- **Uri [xs:anyURI]**
Contains a stable Uri to be used for acquiring a snapshot in JPEG format as well as parameters defining the lifetime of the Uri.
- **InvalidAfterConnect [xs:boolean]**
The parameter shall be set to false.

- **InvalidAfterReboot [xs:boolean]**
The parameter shall be set to false.
- **Timeout [xs:duration]**
The timeout parameter shall be set to PT0S to indicate that this stream URI is indefinitely valid even if the profile changes.

FAULTS:

- **env:Sender - ter:InvalidArgVal - ter:NoProfile**
The media profile does not exist.
- **env:Receiver - ter:Action - ter:IncompleteConfiguration**
The specified media profile does not contain either a reference to a video encoder configuration or a reference to a video source configuration.

ACCESS CLASS:

READ_MEDIA**5.17 Multicast**

See the ONVIF Streaming Specification for a detailed discussion of device and client multicast streaming.

A device supporting multicast streaming (indicated by the RTPMulticast capability) shall support:

- multicast RTSP setup, see GetStreamUri section 5.15
- web service multicast setup, see StartMulticastStreaming and StopMulticastStreaming

5.17.1 StartMulticastStreaming

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 continue after a reboot of the device until a StopMulticastStreaming request is received. 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 case of H.264 video, the SPS/PPS header shall be sent inband.

REQUEST:

- **ProfileToken [tt:ReferenceToken]**
Contains the token of the Profile that is used to define the multicast stream.

RESPONSE:

This is an empty message.

FAULTS:

- **env:Sender - ter:InvalidArgVal - ter:NoProfile**
The profile does not exist.
- **env:Receiver - ter:Action - ter:IncompleteConfiguration**
The specified media profile does not contain either a reference to a video encoder a video source configuration, to a audio source or to audio encoder configuration or a reference to a metadata configuration

ACCESS CLASS:

ACTUATE

5.17.2 StopMulticastStreaming

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

REQUEST:

- **ProfileToken [tt:ReferenceToken]**
Contains the token of the Profile that is used to define the multicast stream.

RESPONSE:

This is an empty message.

FAULTS:

- **env:Sender - ter:InvalidArgVal - ter:NoProfile**
The profile does not exist.
- **env:Receiver - ter:Action - ter:IncompleteConfiguration**
The specified media profile does not contain either a reference to a video encoder a video source configuration, to a audio source or to audio encoder configuration or a reference to a metadata configuration

ACCESS CLASS:

ACTUATE

5.18 Synchronization Points

5.18.1 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 enforces 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 the section “Synchronization Point” of the ONVIF Core Specification). If a PTZ metadata stream is associated to the profile, the PTZ position shall be repeated within the metadata stream.

A device that supports MPEG-4 or H.264 shall support the request for an I-frame through the SetSynchronizationPoint command unless the NoRTSPStreaming capability is set.

REQUEST:

- **ProfileToken [tt:ReferenceToken]**
Contains a Profile reference for which a Synchronization Point is requested.

RESPONSE:

This is an empty message.

FAULTS:

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

ACCESS CLASS:

ACTUATE**5.19 Video source mode**

A device can have the capability for changing video source mode which means a unit which can indicate media profile structure of video sensor in same time. In case that device indicate the capability for video source mode, the configured video source mode is relating to only current media profile structure for video source, video source configuration and video encoder configuration. After setting video source mode a client can see the detail information of settable configuration for the video source configuration and the video encoder configuration from `GetVideoSourceConfigurationOptions` and `GetVideoEncoderConfigurationOptions` commands. In other words the possible configuration of un-setting mode is not seen from any commands, so `GetVideoSourceModes` command provides summary information of possible configuration including video encoder.

5.19.1 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 request message specifies video source.

RESPONSE:

- **VideoSouceMode - unbounded [trt:VideoSourceMode]**
The response contains list of mode information for seeing capabilities of video source.

FAULTS:

- **env:Sender - ter:InvalidArgVal - ter:NoVideoSource**
The requested video source does not exist.

ACCESS CLASS:

READ_SYSTEM**5.19.2 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 behavior after changing the mode is not defined in this specification.

REQUEST:

- **VideoSourceToken [tt:ReferenceToken]**
The request message specifies video source.
- **VideoSourceModeToken [tt:ReferenceToken]**

RESPONSE:

- **Reboot [xs:boolean]**
The response contains information about rebooting after returning response. When `Reboot` is set "true", a device will reboot automatically after setting 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

5.20 OSD (On-Screen Display)

The OSD service provides functions to enable a client to control and configure 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 content of OSD.

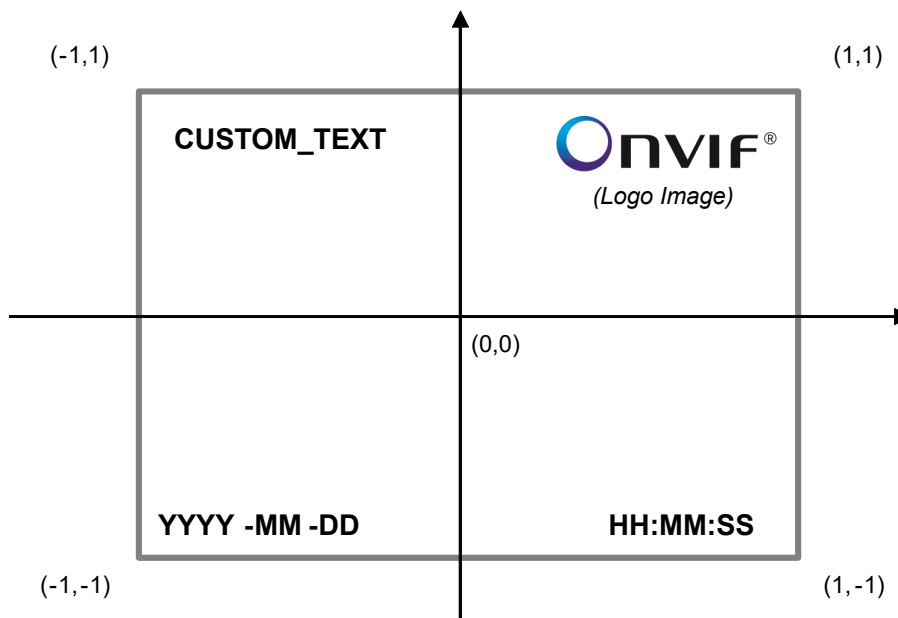


Figure 3: Example of screen which have four OSD configurations and coordinate system

Device supporting temporary OSDTextConfiguration, shall notify TemporaryOSDText capability as defined in Section 5.21. 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.

5.20.1 CreateOSD

This operation creates a new OSD configuration with specified values and also make the association between the new OSD and an existing VideoSourceConfiguration identified by the 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. The OSD shall be created in the device and shall be persistent (remain after reboot). 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.

When creating a OSDTextConfiguration, if the IsPersistentText attribute is missing, device shall assume IsPersistentText attribute as true.

A created OSD 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:

- **Token [xs:string]**
Return the newly created OSD token.

FAULTS:

- **env:Receiver - ter:Action - ter:MaxOSDs**
The maximum number of supported OSDs by the specific VideoSourceConfiguration has been reached.

ACCESS CLASS:

ACTUATE

5.20.2 DeleteOSD

This operation deletes an OSD. This change shall always be persistent. The device shall support the deletion of an OSD through the DeleteOSD command.

REQUEST:

- **OSDToken [tt:ReferenceToken]**
The request message contains an OSD token that indicate which OSD shall be deleted

RESPONSE:

This is an empty message.

FAULTS:

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

ACCESS CLASS:

ACTUATE

5.20.3 GetOSDs

This operation lists all existing OSD configurations for the device. The device shall support the listing of existing OSD configurations through the GetOSDs command.

REQUEST:

- **VideoSourceConfigurationToken optional [tt:ReferenceToken]**
The request message specifies the VideoSourceConfiguration token for which the OSD should be associated with.

RESPONSE:

- **OSD - optional, unbounded [tt:OSDConfiguration]**
The response contains a list of requested OSD for the video source configuration; If no VideoSourceConfiguration token specified, just return all OSDs. If a device has no OSD for specified VideoSourceConfiguration an empty list is returned.

FAULTS:

- **env:Sender - ter:InvalidArgVal - ter:NoConfig**
The requested configuration indicated with VideoSourceConfigurationToken does not exist.

ACCESS CLASS:

READ_MEDIA

5.20.4 GetOSD

If the OSD configuration token is already known, the OSD configuration can be fetched through the GetOSD command. The device shall support retrieval of specific OSD configurations through the GetOSD command.

REQUEST:

- **OSDToken [tt:ReferenceToken]**
This message contains the token of the requested OSD.

RESPONSE:

- **OSD [tt:OSDConfiguration]**
The message contains the requested OSD with the matching token.

FAULTS:

- **env:Sender - ter:InvalidArgVal - ter:NoConfig**
The requested configuration indicated with OSDToken does not exist.

ACCESS CLASS:

READ_MEDIA

5.20.5 SetOSD

This operation modifies an OSD configuration. Running streams using this configuration may be immediately updated according to the new settings. The device shall support the modification of OSD parameters through the SetOSD command.

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.

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 does not exist
- **env:Sender - ter:InvalidArgVal - ter:ConfigModify**
The configuration parameters are not possible to set.

ACCESS CLASS:

ACTUATE

5.20.6 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:

- **VideoSourceConfigurationToken [tt:ReferenceToken]**
The VideoSourceConfigurationToken element specifies the video source configuration of which the suitable OSD options are requested. The VideoSourceConfigurationToken shall exist in the device

RESPONSE:

- **Options [tt:OSDConfigurationOptions]**
This message contains the OSD options which is 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

5.21 GetServiceCapabilities

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

REQUEST:

This is an empty message.

RESPONSE:

- **trt: Capabilities Capabilities [1][1]**
The capability response message contains the requested service capabilities using a hierarchical XML capability structure.

FAULTS:

No command specific faults defined.

ACCESS CLASS:

PRE_AUTH

The following capabilities are available:

RTPMulticast	Indication of support of UDP multicasting as described in Section 5.17.
RTP_TCP	Indication if the device supports RTP over TCP, see Section 5.1.1.2 of the ONVIF Streaming Specification.
RTP_RTSP_TCP	Indication if the device supports RTP/RTSP/TCP transport, see Section 5.1.1.3 of the ONVIF Streaming Specification.
NonAggregateControl	Indicates support for non aggregate RTSP control as described in section 5.2.1.1 of the ONVIF Streaming Specification.

NoRTSPStreaming	Indicates the device does not support live media streaming via RTSP.
MaximumNumberOfProfiles	The maximum Number of MediaProfiles the device supports.
SnapshotUri	Indicates the support for GetSnapshotUri.
Rotation	Indicates the support for the Rotation feature.
VideoSourceMode	Indicates the support for changing video source mode.
OSD	Indication of support of OSD feature.
TemporaryOSDText	Indicates the support for temporary osd text configuration.

5.22 Events

5.22.1 Configuration Change

A device should provide an event to inform subscribed clients when important configurations in the devices change. An ONVIF compliant device shall use the topics defined in the following sections associated with the respective message description.

5.22.1.1 Profile

Whenever a change in the profiles of a device supporting the media service occurs the device should provide the following event. The Profile change could be caused by Creation or Deletion of a Profile or by Adding or Removing a Configuration to or from a Profile.

Topic: tns1:Configuration/Profile

```
<tt:MessageDescription>
  <tt:Source>
    <tt:SimpleItemDescription Name="Token" Type="tt:ReferenceToken" />
  </tt:Source>
  <tt>Data>
    <tt:ElementItemDescription Name="Configuration" Type="tt:Profile" />
  </tt>Data>
</tt:MessageDescription>
```

5.22.1.2 VideoEncoderConfiguration

Whenever a VideoEncoderConfiguration of a device changes the device should provide the following event:

Topic: tns1:Configuration/VideoEncoderConfiguration

```
<tt:MessageDescription>
  <tt:Source>
    <tt:SimpleItemDescription Name="Token" Type="tt:ReferenceToken" />
  </tt:Source>
  <tt>Data>
    <tt:ElementItemDescription Name="Configuration" Type="tt:VideoEncoderConfiguration" />
  </tt>Data>
</tt:MessageDescription>
```

5.22.1.3 VideoSourceConfiguration

Whenever a VideoSourceConfiguration of a device changes the device should provide the following event:

Topic: ns1:Configuration/VideoSourceConfiguration/MediaService

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

```

</tt:Source>
<tt>Data>
  <tt:ElementItemDescription Name="Configuration" Type="tt:VideoSourceConfiguration"/>
</tt>Data>
</tt:MessageDescription>

```

5.22.1.4 VideoOutputConfiguration

Whenever a VideoOutputConfiguration of a device changes the device should provide the following event:

Topic: tns1:Configuration/VideoOutputConfiguration/MediaService

```

<tt:MessageDescription>
  <tt:Source>
    <tt:SimpleItemDescription Name="Token" Type="tt:ReferenceToken"/>
  </tt:Source>
  <tt>Data>
    <tt:ElementItemDescription Name="Configuration" Type="tt:VideoOutputConfiguration"/>
  </tt>Data>
</tt:MessageDescription>

```

5.22.1.5 AudioEncoderConfiguration

Whenever an AudioEncoderConfiguration of a device changes the device should provide the following event:

Topic tns1:Configuration/AudioEncoderConfiguration

```

<tt:MessageDescription>
  <tt:Source>
    <tt:SimpleItemDescription Name="Token" Type="tt:ReferenceToken"/>
  </tt:Source>
  <tt>Data>
    <tt:ElementItemDescription Name="Configuration" Type="tt:AudioEncoderConfiguration"/>
  </tt>Data>
</tt:MessageDescription>

```

5.22.1.6 AudioSourceConfiguration

Whenever an AudioSourceConfiguration of a device changes the device should provide the following event:

Topic tns1:Configuration/AudioSourceConfiguration/MediaService

```

<tt:MessageDescription>
  <tt:Source>
    <tt:SimpleItemDescription Name="Token" Type="tt:ReferenceToken"/>
  </tt:Source>
  <tt>Data>
    <tt:ElementItemDescription Name="Configuration" Type="tt:AudioSourceConfiguration"/>
  </tt>Data>
</tt:MessageDescription>

```

5.22.1.7 AudioOutputConfiguration

Whenever an AudioOutputConfiguration of a device changes the device should provide the following event:

Topic: tns1:Configuration/AudioOutputConfiguration/MediaService

```

<tt:MessageDescription>
  <tt:Source>
    <tt:SimpleItemDescription Name="Token" Type="tt:ReferenceToken"/>
  </tt:Source>
  <tt>Data>
    <tt:ElementItemDescription Name="Configuration" Type="tt:AudioOutputConfiguration"/>
  </tt>Data>

```

```
</tt:MessageDescription>
```

5.22.1.8 MetadataConfiguration

Whenever a MetadataConfiguration of a device changes the device should provide the following event:

Topic: tns1:Configuration/MetadataConfiguration

```
<tt:MessageDescription>
  <tt:Source>
    <tt:SimpleItemDescription Name="Token" Type="tt:ReferenceToken" />
  </tt:Source>
  <tt>Data>
    <tt:ElementItemDescription Name="Configuration" Type="tt:MetadataConfiguration" />
  </tt>Data>
</tt:MessageDescription>
```

5.22.1.9 PTZ Configuration

Whenever a PTZConfiguration of a PTZ capable device changes the device should provide the following event:

Topic: tns1:Configuration/PTZConfiguration

```
<tt:MessageDescription>
  <tt:Source>
    <tt:SimpleItemDescription Name="Token" Type="tt:ReferenceToken" />
  </tt:Source>
  <tt>Data>
    <tt:ElementItemDescription Name="Configuration" Type="tt:PTZConfiguration" />
  </tt>Data>
</tt:MessageDescription>
```

5.22.1.10 VideoAnalyticsConfiguration

Whenever a VideoAnalyticsConfiguration of device changes the device should provide the following event:

Topic: tns1:Configuration/VideoAnalyticsConfiguration

```
<tt:MessageDescription>
  <tt:Source>
    <tt:SimpleItemDescription Name="Token" Type="tt:ReferenceToken" />
  </tt:Source>
  <tt>Data>
    <tt:ElementItemDescription Name="Configuration" Type="tt:VideoAnalyticsConfiguration" />
  </tt>Data>
</tt:MessageDescription>
```

5.22.2 Active Connections

A device that supports the media service should provide the “Active Connections” 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:sequence>
</xs:complexType>
```

```

    <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>

```

NOTE: Active Connections Event is deprecated and its replaced by Active Sessions Event.

5.22.3 Active Sessions

A device that supports the media service should provide the "Active Sessions" monitoring events to inform a client about the current usage of its Media Streams. The monitoring events are sent every time a client connects to or disconnects from a unicast stream. An ONVIF compliant device shall use the following topics and message format:

```

Topics: tns1:Monitoring/ActiveSessions/VideoEncoder
tns1:Monitoring/ActiveSessions/AudioEncoder
tns1:Monitoring/ActiveSessions/AudioDecoder
tns1:Monitoring/ActiveSessions/Metadata
<tt:MessageDescription IsProperty="true">
  <tt:Source>
    <tt:SimpleItemDescription Name="Token" Type="tt:ReferenceToken"/>
  </tt:Source>
  <tt>Data>
    <tt:SimpleItemDescription Name="Sessions" Type="tt:StringAttrList"/>
  </tt>Data>
</tt:MessageDescription>

```

Token refers to the appropriate Video Encoder Configuration, Audio Encoder Configuration, Audio Decoder Configuration, or Metadata Configuration token.

Sessions is a space-delimited list of IPv4 and/or IPv6 addresses of active streaming clients. Multiple clients at an IP address, regardless of streaming protocol, shall be repeated once for every client. Sort order of the list is not defined.

When the first session associated with an encoding resource connects, the event type is Initialized. When all sessions associated with an encoding resource have disconnected, the event type is Deleted.

Example of event for a Video Encoder Configuration with a stream to IPv4 10.220.232.202 and a stream to IPv6 fc80::2934:4e3e:e559:83e9, and then connecting a second stream to 10.220.232.202 (order of Sessions list is undefined; these addresses can appear in any order, but 10.220.232.202 shall appear twice to represent the two streams):

```

<wsnt:Topic Dialect="...">
  tns1:Monitoring/ActiveSessions/VideoEncoder
</wsnt:Topic>
<wsnt:Message>
  <tt:Message UtcTime="..." PropertyOperation="Changed">
    <tt:Source>
      <tt:SimpleItem Name="Token" Value="vec0" />
    </tt:Source>
    <tt>Data>
      <tt:SimpleItem Name="Sessions"
        Value="10.220.232.202 fc80::2934:4e3e:e559:83e9 10.220.232.202" />
    </tt>Data>
  </tt:Message>

```

```
</wsnt:Message>
```

Example of event for a Metadata Configuration when connecting its first active stream to IPv4 10.220.232.202:

```
<wsnt:Topic Dialect="...">
  tns1:Monitoring/ActiveSessions/Metadata
</wsnt:Topic>
<wsnt:Message>
  <tt:Message UtcTime="..." PropertyOperation="Initialized">
    <tt:Source>
      <tt:SimpleItem Name="Token" Value="af16a847-cd62-4923-9ccd-3108a16afae" />
    </tt:Source>
    <tt>Data>
      <tt:SimpleItem Name="Sessions" Value="10.220.232.202" />
    </tt>Data>
  </tt:Message>
</wsnt:Message>
```

Example of event for an Audio Encoder Configuration when all active connections are closed:

```
<wsnt:Topic Dialect="...">
  tns1:Monitoring/ActiveSessions/AudioEncoder
</wsnt:Topic>
<wsnt:Message>
  <tt:Message UtcTime="..." PropertyOperation="Deleted">
    <tt:Source>
      <tt:SimpleItem Name="Token" Value="audio" />
    </tt:Source>
    <tt>Data>
      <tt:SimpleItem Name="Sessions" Value="" />
    </tt>Data>
  </tt:Message>
</wsnt:Message>
```

Annex A. Bibliography

[ONVIF Display WSDL] ONVIF Media WSDL, ver 2.0, 2010.

URL:<http://www.onvif.org/onvif/ver10/network/wsd1/media.wsd1>

[ONVIF Schema] ONVIF Schema, ver 2.0, 2010.

URL:<http://www.onvif.org/onvif/ver10/schema/onvif.xsd>

[ONVIF Topic Namespace] ONVIF Topic Namespace XML, ver 2.0, 2010.

URL:<http://www.onvif.org/onvif/ver10/topics/topicns.xml>

Annex B. Revision History

Rev.	Date	Editor	Changes
2.1	Jul-2011	Hans Busch	Split from Core 2.0 Change Requests 65, 185, 197, 198, 225, 250
2.1.1	Jan-2012	Hans Busch	Change Requests 274, 281, 315, 387, 424, 493, 528, 535, 551, 571, 586
2.2	May-2012	Hans Busch	Change Requests 544, 552, 580, 641, 637, 642, 657
2.2.1	Dec-2012	Michio Hirai	Change Request 826, 855, 789
2.2.1	Dec-2012	Hans Busch	Change Request 708
2.3	May-2013	Michio Hirai	Change Request 790, 968, 1049, 1052
2.4	Mar-2013	Hirokazu Kitaoka	Addition of Video Source Mode feature.
2.4	Mar-2013	Hermes Zhang	Addition of OSD and update for change request 945, 954, 955, 956, 957, 958, 959, 960, 961, 962, 963, 964, 965, 966, 967, 979, 1048
2.4	Aug-2013	Takahiro Iwasaki	Change Request 1048, 1117, 1125, 1126, 1127, 1128, 1146, 1149, 1150, 1151, 1156, 1160, 1161, 1183
2.4	Aug-2013	Michio Hirai	Change Request 1087
2.4.2	Jun-2014	Michio Hirai	Change Request 1342, 1050
2.5	Dec-2014	Hans Busch, Michio Hirai	Add gzip compressed metadata Change Request 1413, 1540
2.6	Jun-2015	Michio Hirai	Change Request 1587, 1602
2.6.1	Dec-2015	Hiroyuki Sano	Change Request 1672
16.06	Jun-2016	Hiroyuki Sano	Change Request 1796, 1869
17.06	Jun-2017	Hans Busch, Hiroyuki Sano	Change Request 1843 Change Request 2105
18.06	Jun-2018	Hiroyuki Sano	Change Request 2216, 2250
19.06	Jun-2019	Steve Wolf, Hiroyuki Sano	Added Active Sessions, Guaranteed Framerate Change Request 2430, 2479
19.12	Dec-2019	Hiroyuki Sano	Change Request 2489, 2557, 2598
20.06	Jun-2020	Michio Hirai	Change Request 2571
21.12	Dec-2021	Oksana Tyushkina	Fix typo in commands names