ONVIF™
Media Service Specification

Version 19.06

June, 2019

CONTENTS

1 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.wsdll .

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

4.1.1 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).

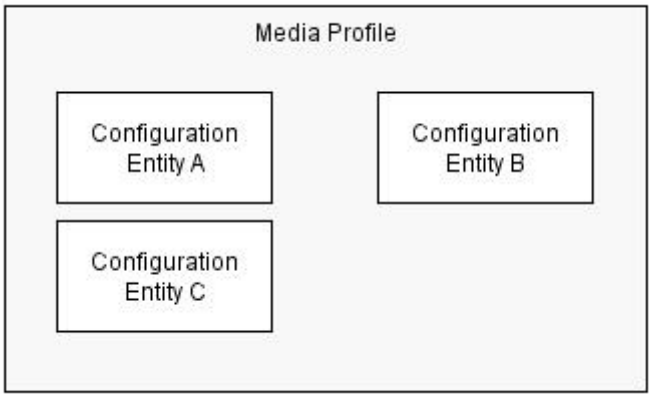

Figure 1: A media profile

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.

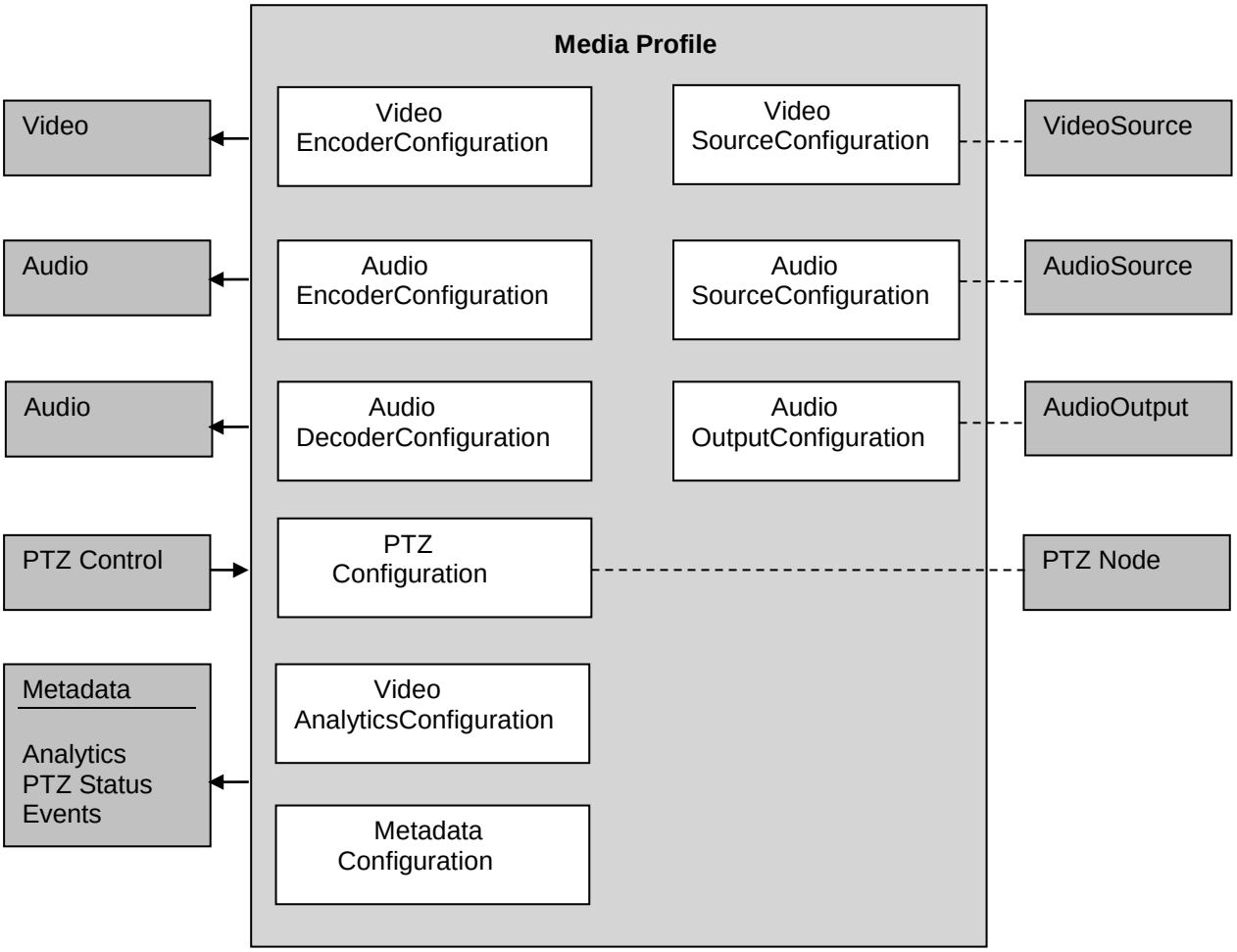

Figure 2: Complete profile configuration

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 1.

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 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.711µ 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 Create media profile

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" | "-.".

Table 2: CreateProfile command

CreateProfile	Access Class: ACTUATE
Message name	**Description**
CreateProfileRequest	*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* tt:Name **Name** [1][1] tt:ReferenceToken **Token** [0][1]
CreateProfileResponse	*Returns an empty Profile structure with no configuration entities.* tt:Profile **Profile** [1][1]
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:ProfileExists	*A profile with the token **ProfileToken** already exists.*
env:Receiver ter:Action ter:MaxNVTProfiles	*The maximum number of supported profiles supported by the device has been reached.*

5.2.2 Get media profiles

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.

Table 3: GetProfiles command

GetProfiles	Access Class: READ_MEDIA
Message name	**Description**
GetProfilesRequest	*This is an empty message.*
GetProfilesResponse	*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.* tt:Profile **Profiles** [0][unbounded]
Fault codes	**Description**
	No command specific faults!

5.2.3 Get media profile

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.

Table 4: GetProfile command

GetProfile	Access Class: READ_MEDIA
Message name	**Description**
GetProfileRequest	*This message contains the token to the requested profile.* tt:ReferenceToken **ProfileToken** [1][1]
GetProfileResponse	*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.* tt:Profile **Profile** [1][1]
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NoProfile	*The requested profile token **ProfileToken** does not exist.*

5.2.4 Add video source configuration to a profile

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.

Table 5: AddVideoSourceConfiguration command

AddVideoSourceConfiguration	Access Class: ACTUATE
Message name	**Description**
AddVideoSourceConfiguration Request	*Contains a reference to the **VideoSourceConfiguration** to add and the **Profile** where it shall be added.* tt:ReferenceToken **ProfileToken** [1][1] tt:ReferenceToken **ConfigurationToken** [1][1]
AddVideoSourceConfiguration Response	*This is an empty message.*
Fault codes	**Description**
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.*

5.2.5 Add video encoder configuration to a profile

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 VideoSourceConfiguration and shall support streaming video data of such a Profile.

Table 6: AddVideoEncoderConfiguration command

AddVideoEncoderConfiguration	Access Class: ACTUATE
Message name	**Description**
AddVideoEncoderConfiguration Request	*Contains a reference to the **VideoEncoderConfiguration** to add and the **Profile** where it shall be added.* tt:ReferenceToken **ProfileToken** [1][1] tt:ReferenceToken **ConfigurationToken** [1][1]
AddVideoEncoderConfiguration Response	*This is an empty message.*
Fault codes	**Description**
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.*

5.2.6 Add audio source configuration to a profile

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 AddAudioSourceConfiguration command.

Table 7: AddAudioSourceConfiguration command

AddAudioSourceConfiguration	Access Class: ACTUATE
Message name	**Description**
AddAudioSourceConfiguration Request	*Contains a reference to the **AudioSourceConfiguration** to add and the **Profile** where it shall be added.* tt:ReferenceToken **ProfileToken** [1][1] tt:ReferenceToken **ConfigurationToken** [1][1]
AddAudioSourceConfiguration Response	*This is an empty message.*
Fault codes	**Description**
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.*

5.2.7 Add audio encoder configuration to a profile

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.

Table 8: AddAudioEncoderConfiguration command

AddAudioEncoderConfiguration	Access Class: ACTUATE
Message name	**Description**
AddAudioEncoderConfiguration Request	*Contains a reference to the **AudioEncoderConfiguration** to add and the **Profile** where it shall be added.* tt:ReferenceToken **ProfileToken** [1][1] tt:ReferenceToken **ConfigurationToken** [1][1]
AddAudioEncoderConfiguration Response	*This is an empty message.*
Fault codes	**Description**
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.*

5.2.8 Add PTZ configuration to a profile

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.

Table 9: AddPTZConfiguration command

AddPTZConfiguration	Access Class: ACTUATE
Message name	**Description**
AddPTZConfigurationRequest	*Contains a reference to the **PTZConfiguration** to add and the **Profile** where it shall be added.* tt:ReferenceToken **ProfileToken** [1][1] tt:ReferenceToken **ConfigurationToken** [1][1]
AddPTZConfigurationResponse	*This is an empty message.*
Fault codes	**Description**
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.*

5.2.9 Add video analytics configuration to a profile

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.

Table 10: AddVideoAnalyticsConfiguration command

AddVideoAnalyticsConfiguration	Access Class: ACTUATE
Message name	**Description**
AddVideoAnalyticsConfigurationRequest	*Contains a reference to the **VideoAnalyticsConfiguration** to add and the **Profile** where it shall be added.* tt:ReferenceToken **ProfileToken** [1][1] tt:ReferenceToken **ConfigurationToken** [1][1]

AddVideoAnalyticsConfigurationResponse	*This is an empty message.*
Fault codes	**Description**
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.*

5.2.10 Add metadata configuration to a profile

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 though 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.

Table 11: AddMetadataConfiguration command

AddMetadataConfiguration	Access Class: ACTUATE
Message name	**Description**
AddMetadataConfiguration Request	*Contains a reference to the **MetadataConfiguration** to add and the **Profile** where it shall be added.* tt:ReferenceToken **ProfileToken** [1][1] tt:ReferenceToken **ConfigurationToken** [1][1]
AddMetadataConfiguration Response	*This is an empty message.*
Fault codes	**Description**
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.*

5.2.11 Add audio output configuration

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. An 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.

Table 12: AddAudioOutputConfiguration

AddAudioOutputConfiguration	Access Class: ACTUATE
Message name	**Description**
AddAudioOutputConfiguration Request	*Contains a reference to the AudioOutputConfiguration to add and the Profile where it shall be added.* tt:ReferenceToken **ProfileToken** [1][1] tt:ReferenceToken **ConfigurationToken** [1][1]
AddAudioOutputConfiguration Response	*This is an empty message.*
Fault codes	**Description**
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*

5.2.12 Add audio decoder configuration

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.

Table 13: AddAudioDecoderConfiguration

AddAudioDecoderConfiguration	Access Class: ACTUATE
Message name	**Description**
AddAudioDecoderConfiguration Request	*Contains a reference to the AudioConfiguration to add and the Profile where it shall be added.* tt:ReferenceToken **ProfileToken** [1][1] tt:ReferenceToken **ConfigurationToken** [1][1]
AddAudioDecoderConfiguration Response	*This is an empty message.*
Fault codes	**Description**
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*

5.2.13 Remove video source configuration from a profile

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 RemoveVideoSourceConfiguration command.

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

Table 14: RemoveVideoSourceConfiguration command

RemoveVideoSourceConfiguration	Access Class: ACTUATE
Message name	**Description**
RemoveVideoSourceConfiguration-Request	*Contains a reference to the media profile from which the VideoSourceConfiguration shall be removed.* tt:ReferenceToken **ProfileToken** [1][1]
RemoveVideoSourceConfiguration-Response	*This is an empty message.*
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NoProfile	*The requested profile token **ProfileToken** 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.*

5.2.14 Remove video encoder configuration from a profile

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.

Table 15: RemoveVideoEncoderConfiguration command

RemoveVideoEncoderConfiguration	Access Class: ACTUATE
Message name	**Description**
RemoveVideoEncoderConfiguration-Request	*Contains a reference to the media profile from which the VideoEncoderConfiguration shall be removed.* tt:ReferenceToken **ProfileToken** [1][1]

RemoveVideoEncoderConfiguration-Response	*This is an empty message.*
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NoProfile	*The requested profile token **ProfileToken** 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.*

5.2.15 Remove audio source configuration from a profile

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.

Table 16: RemoveAudioSourceConfiguration command

RemoveAudioSourceConfiguration	Access Class: ACTUATE
Message name	**Description**
RemoveAudioSourceConfiguration-Request	*Contains a reference to the media profile from which the AudioSourceConfiguration shall be removed.* tt:ReferenceToken **ProfileToken** [1][1]
RemoveAudioSourceConfiguration-Response	*This is an empty message.*
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NoProfile	*The requested profile token **ProfileToken** 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.*

5.2.16 Remove audio encoder configuration from a profile

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.

Table 17: RemoveAudioEncoderConfiguration command

RemoveAudioEncoderConfiguration	Access Class: ACTUATE
Message name	**Description**
RemoveAudioEncoderConfiguration-Request	*Contains a reference to the media profile from which the AudioEncoderConfiguration shall be removed.* tt:ReferenceToken **ProfileToken** [1][1]
RemoveAudioEncoderConfiguration-Response	*This is an empty message.*
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NoProfile	*The requested profile token **ProfileToken** 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.*

5.2.17 Remove PTZ configuration from a profile

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.

Table 18: RemovePTZConfiguration command

RemovePTZConfiguration	Access Class: ACTUATE
Message name	**Description**
RemovePTZConfiguration-Request	*Contains a reference to the media profile from which the PTZConfiguration shall be removed.* tt:ReferenceToken **ProfileToken** [1][1]
RemovePTZConfiguration-Response	*This is an empty message.*
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NoProfile	*The requested profile token **ProfileToken** does 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.*

5.2.18 Remove video analytics configuration from a profile

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.

Table 19: RemoveVideoAnalyticsConfiguration command

RemoveVideoAnalyticsConfiguration	Access Class: ACTUATE
Message name	**Description**
RemoveVideoAnalyticsConfiguration-Request	*Contains a reference to the media profile from which the VideoAnalyticsConfiguration shall be removed.* tt:ReferenceToken **ProfileToken** [1][1]
RemoveVideoAnalyticsConfiguration-Response	*This is an empty message.*
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NoProfile	*The requested profile token **ProfileToken** does 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.*

5.2.19 Remove metadata configuration from a profile

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.

Table 20: RemoveMetadataConfiguration command

RemoveMetadataConfiguration	Access Class: ACTUATE
Message name	**Description**
RemoveMetadataConfiguration-Request	*Contains a reference to the media profile from which the MetadataConfiguration shall be removed.* tt:ReferenceToken **ProfileToken** [1][1]
RemoveMetadataConfiguration-Response	*This is an empty message.*
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NoProfile	*The requested profile token **ProfileToken** does 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.*

5.2.20 Remove audio output configuration

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.

Table 21: RemoveAudioOutputConfiguration

RemoveAudioOutputConfiguration	Access Class: ACTUATE
Message name	**Description**
RemoveAudioOutputConfiguration-Request	*Contains a reference to the media profile from which the AudioOutputConfiguration shall be removed*. tt:ReferenceToken **ProfileToken** [1][1]
RemoveAudioOutputConfiguration-Response	*This is an empty message.*
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NoProfile	*The requested profile token ProfileToken does 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

5.2.21 Remove audio decoder configuration

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.

Table 22: RemoveAudioDecoderConfiguration

RemoveAudioDecoderConfiguration	Access Class: ACTUATE
Message name	**Description**
RemoveAudioDecoderConfiguration-Request	*Contains a reference to the media profile from which the AudioDecoderConfiguration shall be removed*. tt:ReferenceToken **ProfileToken** [1][1]
RemoveAudioDecoderConfiguration-Response	*This is an empty message.*

Fault codes	Description
env:Sender ter:InvalidArgVal ter:NoProfile	*The requested profile token ProfileToken does 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*

5.2.22 Delete media profile

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.

Table 23: DeleteProfile command

DeleteProfile	Access Class: ACTUATE
Message name	**Description**
DeleteProfileRequest	*Contains a **ProfileToken** that indicates what media profile to delete.* tt:ReferenceToken **ProfileToken** [1][1]
DeleteProfileResponse	*This is an empty message.*
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NoProfile	*The requested profile token **ProfileToken** does not exist.*
env:Sender ter:Action ter:DeletionOfFixedProfile	*The fixed Profile cannot be deleted.*

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 Imagaing 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.

Table 24: GetVideoSources command

GetVideoSources	Access Class: READ_MEDIA
Message name	**Description**

GetVideoSourcesRequest	*This is an empty message.*
GetVideoSourcesResponse	*Contains a list of structures describing all available video sources of the device.* tt:VideoSource **VideoSources** [0][unbounded]
Fault codes	**Description**
	No command specific faults!

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 Get video source configurations

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.

Table 25: GetVideoSourceConfigurations command

GetVideoSourceConfigurations	Access Class: READ_MEDIA
Message name	**Description**
GetVideoSourceConfigurations-Request	*This is an empty message.*
GetVideoSourceConfigurations-Response	*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.* tt:VideoSourceConfiguration **Configurations** [0][unbounded]
Fault codes	**Description**
	No command specific faults!

5.4.2 Get video source configuration

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.

Table 26: GetVideoSourceConfiguration command

GetVideoSourceConfiguration	Access Class: READ_MEDIA
Message name	**Description**

GetVideoSourceConfiguration-Request	*This message contains the token of the requested video source configuration.* tt:ReferenceToken **ConfigurationToken** [1][1]
GetVideoSourceConfiguration-Response	*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.* tt:VideoSourceConfiguration **Configuration** [1][1]
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NoConfig	*The requested configuration indicated with **ConfigurationToken** does not exist.*

5.4.3 Get compatible video source configurations

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.

Table 27: GetCompatibleVideoSourceConfigurations command

GetCompatibleVideoSourceConfigurations	Access Class: READ_MEDIA
Message name	**Description**
GetCompatibleVideoSource-ConfigurationsRequest	*Contains the token of an existing media profile.* tt:ReferenceToken **ProfileToken** [1][1]
GetCompatibleVideoSource-ConfigurationsResponse	*Contains a list of video source configurations that are compatible with the media profile.* tt:VideoSourceConfiguration **Configurations** [0][unbounded]
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NoProfile	*The requested profile token **ProfileToken** does not exist.*

5.4.4 Get video source configuration options

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.

Table 28: GetVideoSourceConfigurationOptions command

GetVideoSourceConfigurationOptions	Access Class: READ_MEDIA
Message name	**Description**
GetVideoSourceConfiguration-OptionsRequest	*This message may contain a media profile or video source configuration token, or both.* tt:ReferenceToken **ConfigurationToken** [0][1] tt:ReferenceToken **ProfileToken** [0][1]
GetVideoSourceConfiguration-OptionsResponse	*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.* tt:VideoSourceConfigurationOptions **Options** [1][1]
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NoProfile	*The requested profile token **ProfileToken** does not exist.*
env:Sender ter:InvalidArgVal ter:NoConfig	*The requested configuration does not exist.*

5.4.5 Modify a video source configuration

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.

Table 29: SetVideoSourceConfiguration command

SetVideoSourceConfiguration	Access Class: ACTUATE
Message name	**Description**
SetVideoSourceConfiguration-Request	*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.* tt:VideoSourceConfiguration **Configuration** [1][1] xs:boolean **ForcePersistence** [1][1]
SetVideoSourceConfiguration-Response	*This message is empty.*
Fault codes	**Description**
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.*

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 Get video encoder configurations

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.

Table 30: GetVideoEncoderConfigurations command

GetVideoEncoderConfigurations	Access Class: READ_MEDIA
Message name	**Description**
GetVideoEncoderConfigurations-Request	*This is an empty message.*
GetVideoEncoderConfigurations-Response	*This message contains a list of* all *existing video encoder configurations in the* device. tt:VideoEncoderConfiguration **Configurations** [0][unbounded]
Fault codes	**Description**
	No command specific faults!

5.5.2 Get video encoder configuration

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.

Table 31: GetVideoEncoderConfiguration command

GetVideoEncoderConfiguration	Access Class: READ_MEDIA
Message name	**Description**
GetVideoEncoderConfiguration-Request	*This message contains the token of the requested video encoder configuration.* tt:ReferenceToken **ConfigurationToken** [1][1]
GetVideoEncoderConfiguration-Response	*This message contains the requested VideoEncoderConfiguration with the matching token.* tt:VideoEncoderConfiguration **Configuration** [1][1]
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NoConfig	*The requested configuration indicated with **ConfigurationToken** does not exist.*

5.5.3 Get compatible video encoder configurations

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.

Table 32: GetCompatibleVideoEncoderConfigurations command

GetCompatibleVideoEncoderConfigurations	Access Class: READ_MEDIA
Message name	**Description**
GetCompatibleVideoEncoder-ConfigurationsRequest	*Contains the token of an existing media profile.* tt:ReferenceToken **ProfileToken** [1][1]
GetCompatibleVideoEncoder-ConfigurationsResponse	*Contains a list of video encoder configurations that are compatible with the given media profile.* tt:VideoEncoderConfiguration **Configurations** [0][unbounded]
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NoProfile	*The requested profile token **ProfileToken** does not exist.*

5.5.4 Get video encoder configuration options

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.

Table 33: GetVideoEncoderConfigurationOptions command

GetVideoEncoderConfigurationOptions	Access Class: READ_MEDIA
Message name	**Description**
GetVideoEncoderConfiguration-OptionsRequest	*This message may contain a media profile or video encoder configuration token, or both.* tt:ReferenceToken **ConfigurationToken** [0][1] tt:ReferenceToken **ProfileToken** [0][1]
GetVideoEncoderConfiguration-OptionsResponse	*This message contains the video configuration options.* tt:VideoEncoderConfigurationOptions **Options** [1][1]
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NoProfile	*The requested profile token **ProfileToken** does not exist.*

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

5.5.5 Modify a video encoder configuration

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.

Clientmethods 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.

Table 34: SetVideoEncoderConfiguration command

SetVideoEncoderConfiguration	Access Class: ACTUATE
Message name	**Description**
SetVideoEncoderConfiguration-Request	*The **Configuration** element contains the modified video encoder configuration. The configuration shall exist in the* device*.* *The **ForcePersistence** element is obsolete and should always assumed to be true.* tt:VideoEncoderConfiguration **Configuration** [1][1] xs:boolean **ForcePersistence** [1][1]
SetVideoEncoderConfiguration-Response	*This message is empty.*
Fault codes	**Description**
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.*

5.5.6 Get guaranteed number of video encoder instances

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.

Table 35: GetGuaranteedNumberOfVideoEncoderInstances command

GetGuaranteedNumberOfVideoEncoderInstances	Access Class: READ_MEDIA
Message name	**Description**
GetGuaranteedNumberOf-EncoderInstancesRequest	*This request contains a token to the video source configuration.* tt: ReferenceToken **ConfigurationToken** [1][1]
GetGuaranteedNumberOf-EncoderInstancesResponse	*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.* xs:int **TotalNumber** [1][1] xs:int **JPEG** [0][1] xs:int **H264** [0][1] xs:int **MPEG4** [0][1]
Fault codes	**Description**
env:Sender *ter:InvalidArgVal* *ter:NoConfig*	*The requested configuration indicated with **ConfigurationToken** does not exist.*

5.6 Audio source

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

5.6.1 Get audio sources

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.

Table 36: GetAudioSources command

GetAudioSources	Access Class: READ_MEDIA
Message name	**Description**
GetAudioSourcesRequest	*This message is empty.*
GetAudioSourcesResponse	*Contains a list of structures describing all available audio sources of the device.* tt:AudioSource **AudioSources** [0][unbounded]
Fault codes	**Description**
env:Receiver ter:ActionNotSupported ter:AudioNotSupported	The device *does not support audio.*

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 Get audio source configurations

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.

Table 37: GetAudioSourceConfigurations command

GetAudioSourceConfigurations	Access Class: READ_MEDIA
Message name	**Description**
GetAudioSourceConfigurations-Request	*This is an empty message.*
GetAudioSourceConfigurations-Response	*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.* tt:AudioSourceConfiguration **Configurations** [0][unbounded]
Fault codes	**Description**
env:Receiver ter:ActionNotSupported ter:AudioNotSupported	*The* device *does not support audio.*

5.7.2 Get audio source configuration

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.

Table 38: GetAudioSourceConfiguration command

GetAudioSourceConfiguration	Access Class: READ_MEDIA
Message name	**Description**
GetAudioSourceConfiguration-Request	*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.* tt:ReferenceToken **ConfigurationToken** [1][1]
GetAudioSourceConfiguration-Response	*This message contains the requested AudioSourceConfiguration with the matching token.* tt:AudioSourceConfiguration **Configuration** [1][1]
Fault codes	**Description**
env:Sender ter:InvalidArgVal	*The requested configuration indicated with **ConfigurationToken** does not exist.*

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

5.7.3 Get compatible audio source configurations

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.

Table 39: GetCompatibleAudioSourceConfigurations command

GetCompatibleAudioSourceConfigurations	Access Class: READ_MEDIA
Message name	**Description**
GetCompatibleAudioSource-ConfigurationsRequest	*Contains the token of an existing media profile.* tt:ReferenceToken **ProfileToken** [1][1]
GetCompatibleAudioSource-ConfigurationsResponse	*Contains a list of audio source configurations that are compatible with the media profile.* tt:AudioSourceConfiguration **Configurations** [0][unbounded]
Fault codes	**Description**
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.*

5.7.4 Get audio source configuration options

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.

Table 40: GetAudioSourceConfigurationOptions command

GetAudioSourceConfigurationOptions	Access Class: READ_MEDIA
Message name	**Description**

GetAudioSourceConfiguration-OptionsRequest	*This message may contain a media profile or audio source configuration token, or both.* tt:ReferenceToken **ConfigurationToken** [0][1] tt:ReferenceToken **ProfileToken** [0][1]
GetAudioSourceConfiguration-OptionsResponse	*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.* tt:AudioSourceConfigurationOptions **Options** [1][1]
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NoProfile	*The requested profile token **ProfileToken** 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.*

5.7.5 Modify an audio source configuration

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.

Table 41: SetAudioSourceConfiguration command

SetAudioSourceConfiguration	Access Class: ACTUATE
Message name	**Description**
SetAudioSourceConfiguration-Request	*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.* tt:AudioSourceConfiguration **Configuration** [1][1] xs:boolean **ForcePersistence** [1][1]
SetAudioSourceConfiguration-Response	*This message is empty.*
Fault codes	**Description**

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.*

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 Get audio encoder configurations

This operation lists all *existing* device audio encoder configurations. The client does not need to know anything apriori 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.

Table 42: GetAudioEncoderConfigurations command

GetAudioEncoderConfigurations	Access Class: READ_MEDIA
Message name	**Description**
GetAudioEncoderConfigurations-Request	*This is an empty message.*
GetAudioEncoderConfigurations-Response	*This message contains a list of* all *existing audio encoder configurations in the* device*.* tt:AudioEncoderConfiguration **Configurations** [0][unbounded]
Fault codes	**Description**
env:Receiver ter:ActionNotSupported ter:AudioNotSupported	*The* device *does not support audio.*

5.8.2 Get audio encoder configuration

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.

Table 43: GetAudioEncoderConfiguration command

GetAudioEncoderConfiguration	Access Class: READ_MEDIA
Message name	**Description**
GetAudioEncoderConfiguration-Request	*This message contains the token of the requested audio encoder configuration.* tt:ReferenceToken **ConfigurationToken** [1][1]
GetAudioEncoderConfiguration-Response	*This message contains the requested AudioEncoderConfiguration with the matching token.* tt:AudioEncoderConfiguration **Configuration** [1][1]
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NoConfig	*The configuration does not exist.*
env:Receiver ter:ActionNotSupported ter:AudioNotSupported	*The* device *does not support audio.*

5.8.3 Get compatible audio encoder configurations

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 AddAudioEncoderConfiguration 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.

Table 44: GetCompatibleAudioEncoderConfigurations command

GetCompatibleAudioEncoderConfigurations	Access Class: READ_MEDIA
Message name	**Description**
GetCompatibleAudioEncoder-ConfigurationsRequest	*Contains the token of an existing media profile.* tt:ReferenceToken **ProfileToken** [1][1]
GetCompatibleAudioEncoder-ConfigurationsResponse	*Contains a list of audio encoder configurations that are compatible with the given media profile.* tt:AudioEncoderConfiguration **Configurations** [0][unbounded]
Fault codes	**Description**
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.*

5.8.4 Get audio encoder configuration options

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.

Table 45: GetAudioEncoderConfigurationOptions command

GetAudioEncoderConfigurationOptions	Access Class: READ_MEDIA
Message name	**Description**
GetAudioEncoderConfiguration-OptionsRequest	*This message may contain a media profile or audio encoder configuration token, or both.* tt:ReferenceToken **ConfigurationToken** [0][1] tt:ReferenceToken **ProfileToken** [0][1]
GetAudioEncoderConfiguration-OptionsResponse	*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.* tt:AudioEncoderConfigurationOptions **Options** [1][1]
Fault codes	**Description**
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.*

5.8.5 Modify audio encoder configurations

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.

Table 46: SetAudioEncoderConfiguration command

SetAudioEncoderConfiguration	Access Class: ACTUATE
Message name	**Description**
SetAudioEncoderConfiguration-Request	*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.* tt:AudioEncoderConfiguration **Configuration** [1][1] xs:boolean **ForcePersistence** [1][1]
SetAudioEncoderConfiguration-Response	*This message is empty.*
Fault codes	**Description**
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.*

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 Get video analytics configurations

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.

Table 47: GetVideoAnalyticsConfigurations command

GetVideoAnalyticsConfigurations	Access Class: READ_MEDIA
Message name	**Description**
GetVideoAnalyticsConfigurations-Request	*This message is empty.*
GetVideoAnalyticsConfigurations-Response	*This message contains a list of* all *existing video analytics configurations in the device.* tt:VideoAnalyticsConfiguration **Configurations** [0][unbounded]
Fault codes	**Description**
env:Sender ter:ActionNotSupported ter:VideoAnalyticsNot-Supported	*Device does not support video analytics.*

5.9.2 Get video analytics configuration

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.

Table 48: GetVideoAnalyticsConfiguration command

GetVideoAnalyticsConfiguration	Access Class: READ_MEDIA
Message name	**Description**
GetVideoAnalyticsConfiguration-Request	*This message contains the token of an existing video analytics configuration.* tt:ReferenceToken **ConfigurationToken** [1][1]
GetVideoAnalyticsConfiguration-Response	*This message contains the requested video analytics configuration.* tt:VideoAnalyticsConfiguration **Configuration** [1][1]
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NoConfig	*The requested configuration indicated with **ConfigurationToken** does not exist.*
env:Sender ter:ActionNotSupported ter:VideoAnalyticsNot-Supported	*The device does not support video analytics.*

5.9.3 Get compatible video analytics configurations

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 AddVideoAnalyticsConfiguration 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.

Table 49: GetCompatibleVideoAnalyticsConfigurations command

GetCompatibleVideoAnalyticsConfigurations	Access Class: READ_MEDIA
Message name	**Description**
GetCompatibleVideoAnalytics-ConfigurationsRequest	*Contains the token of an existing media profile.* tt:ReferenceToken **ProfileToken** [1][1]
GetCompatibleVideoAnalytics-ConfigurationsResponse	*Contains a list of video analytics configurations that are compatible with the given media profile.* tt:VideoAnalyticsConfiguration **Configurations** [0][unbounded]
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NoProfile	*The requested profile token **ProfileToken** does not exist.*
env:Sender ter:ActionNotSupported ter:VideoAnalyticsNot-Supported	*The device does not support video analytics.*

5.9.4 Modify a video analytics configuration

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.

Table 50: SetVideoAnalyticsConfiguration command

SetVideoAnalyticsConfiguration	Access Class: ACTUATE
Message name	**Description**
SetVideoAnalyticsConfiguration-Request	*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.* tt:VideoAnalyticsConfiguration **Configuration** [1][1] xs:boolean **ForcePersistence** [1][1]
SetVideoAnalyticsConfiguration-Response	*This message is empty.*
Fault codes	**Description**
env:Sender ter:InvalidArgVal	*The configuration does not exist.*

ter:NoConfig	
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.*

5.10 Metadata configuration

A MetadataConfiguration contains parameters for selecting the data to include in the metadata stream. The choices include PTZ status, PTZ position, events as defined by a subscription and analytics data . The event subscription data is described in the section “Event Handling” of the ONVIF Core Specification. The analytics parameters define which data to include from the analytics engine part of the profile, see Section 5.9.

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 51 and Table 52) shall be supported by a devices signalling the support for EXI compression via GetMetadataConfigurationOptions. Schema based EXI encoding shall be used. The required schema may be obtained from a device using the GetWsdlUrl command.

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 51) shall always be used.

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 52
Padding Bits	If present must be “0“.

Table 52 ONVIF defined EXI configuration settings

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

5.10.2 Get metadata configurations

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.

Table 53: GetMetadataConfigurations command

GetMetadataConfigurations	Access Class: READ_MEDIA
Message name	**Description**
GetMetadataConfigurations-Request	*This message is empty.*
GetMetadataConfigurations-Response	*This message contains a list of* all *existing metadata configurations in the device.* tt:MetadataConfiguration **Configurations** [0][unbounded]
Fault codes	**Description**
	No command specific faults!

5.10.3 Get metadata configuration

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.

Table 54: GetMetadataConfiguration command

GetMetadataConfiguration	Access Class: READ_MEDIA
Message name	**Description**
GetMetadataConfiguration-Request	*This message contains the token of an existing metadata configuration.* tt:ReferenceToken **ConfigurationToken** [1][1]
GetMetadataConfiguration-Response	*This message contains the requested metadata configuration.* tt:MetadataConfiguration **Configuration** [1][1]
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NoConfig	*The requested configuration indicated with **ConfigurationToken** does not exist.*

5.10.4 Get compatible metadata configurations

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.

Table 55: GetCompatibleMetadataConfigurations command

GetCompatibleMetadataConfigurations	Access Class: READ_MEDIA
Message name	**Description**
GetCompatibleMetadata-ConfigurationsRequest	*Contains the token of an existing media profile.* tt:ReferenceToken **ProfileToken** [1][1]
GetCompatibleMetadata-ConfigurationsResponse	*Contains a list of metadata configurations that are compatible with the given media profile.* tt:MetadataConfiguration **Configurations** [0][unbounded]
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NoProfile	*The requested profile token **ProfileToken** does not exist.*

5.10.5 Get metadata configuration options

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.

Table 56: GetMetadataConfigurationOptions command

GetMetadataConfigurationOptions	Access Class: READ_MEDIA
Message name	**Description**
GetMetadataConfiguration-OptionsRequest	*This message may contain a media profile or metadata configuration token, or both.* tt:ReferenceToken **ConfigurationToken** [0][1] tt:ReferenceToken **ProfileToken** [0][1]
GetMetadataConfiguration-OptionsResponse	*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.* tt:MetadataConfigurationOptions **Options** [1][1]
Fault codes	**Description**
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.*

5.10.6 Modify a metadata configuration

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.

Table 57: SetMetadataConfiguration command

SetMetadataConfiguration	Access Class: ACTUATE
Message name	**Description**
SetMetadataConfiguration-Request	*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 assumed to be true.* tt:MetadataConfiguration **Configuration** [1][1] xs:boolean **ForcePersistence** [1][1]
SetMetadataConfiguration-Response	*This message is empty.*
Fault codes	**Description**
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.*

5.11 Audio outputs

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

5.11.1 Get audio outputs

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.

Table 58: GetAudioOutputs

GetAudioOutputs	Access Class: READ_MEDIA
Message name	**Description**
GetAudioOutputsRequest	*This is an empty message.*
GetAudioOutputsResponse	*Contains a list of structures describing all available audio outputs of the device. If a device has no AudioOutputs an empty list is returned.* tt:AudioOutput **AudioOutputs** [0][unbounded]
Fault codes	**Description**
env:Receiver ter:ActionNotSupported ter:AudioOutputNotSupported	*Audio or Audio Outputs are not supported by the* device

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 Get audio output configurations

This command lists all existing AudioOutputConfigurations of a device. The client does not need to know anything apriori 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.

Table 59: GetAudioOutputConfigurations

GetAudioOutputConfigurations	Access Class: READ_MEDIA
Message name	**Description**
GetAudioOutputConfigurationsRequest	*This is an empty message.*
GetAudioOutputConfigurationsResponse	*Contains a list of AudioOutputConfigurations that are available on the device* tt:AudioOutputConfiguration **Configurations** [0][unbounded]
Fault codes	**Description**
env: Receiver ter:ActionNotSupported ter:AudioOutputNotSupported	*Audio or Audio Outputs are not supported by the device*

5.12.2 Get audio output configuration

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.

Table 60: GetAudioOutputConfiguration

GetAudioOutputConfiguration	Access Class: READ_MEDIA
Message name	**Description**
GetAudioOutputConfigurationRequest	*This message contains the token of the requested AudioOutput configuration.* tt:ReferenceToken **ConfigurationToken** [1][1]
GetAudioOutputConfigurationResponse	*This message contains the requested AudioOutputConfiguration with the matching token.* tt:AudioOutputConfiguration **Configuration** [1][1]
Fault codes	**Description**
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*

5.12.3 Get compatible audio output configurations

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. An 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.

Table 61: GetCompatibleAudioOutputConfiguration

GetCompatibleAudioOutputConfigurations	Access Class: READ_MEDIA
Message name	**Description**
GetCompatibleAudioOutputConfigurations Request	*Contains the token of an existing media profile.* tt:ReferenceToken **ProfileToken** [1][1]
GetCompatibleAudioOutputConfigurations Response	*Contains a list of audio output configurations that are compatible with the given media profile.* tt:AudioOutputConfiguration **Configurations** [0][unbounded]
Fault codes	**Description**
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*

5.12.4 Get audio output configuration options

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.

Table 62: GetAudioOutputConfigurationOptions

GetAudioOutputConfigurationOptions	Access Class: READ_MEDIA
Message name	**Description**
GetAudioOutputConfiguration-OptionsRequest	*This message may contain a media profile or audio output configuration token, or both.* tt:ReferenceToken **ConfigurationToken** [0][1] tt:ReferenceToken **ProfileToken** [0][1]
GetAudioOutputConfiguration-OptionsResponse	*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.* tt:AudioOutputConfigurationOptions **Options** [1][1]
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NoProfile	*The requested profile token ProfileToken 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*

5.12.5 Modify audio output configuration

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.

Table 63: SetAudioOutputConfiguration

SetAudioOutputConfiguration	Access Class: ACTUATE
Message name	**Description**
SetAudioOutputConfiguration-Request	*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.* tt:AudioOutputConfiguration **Configuration** [1][1] xs:boolean **ForcePersistence** [1][1]
SetAudioOutputConfiguration-Response	*This message is empty.*
Fault codes	**Description**
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*

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 Get audio decoder configurations

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.

Table 64: GetAudioDecoderConfigurations

GetAudioDecoderConfigurations	Access Class: READ_MEDIA
Message name	**Description**
GetAudioDecoderConfigurationsRequest	*This is an empty message.*
GetAudioDecoderConfigurationsResponse	*Contains a list of AudioDecoderConfigurations that are available on the device* tt:AudioDecoderConfiguration **Configurations** [0][unbounded]
Fault codes	**Description**
env:Receiver ter:ActionNotSupported ter:AudioDecodingNotSupported	*Audio or Audio decoding is not supported by the device*

5.13.2 Get audio decoder configuration

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.

Table 65: GetAudioDecoderConfiguration

GetAudioDecoderConfiguration	Access Class: READ_MEDIA
Message name	**Description**
GetAudioDecoderConfigurationRequest	*This message contains the token of the requested AudioDecoder configuration.* tt:ReferenceToken **ConfigurationToken** [1][1]
GetAudioDecoderConfigurationResponse	*This message contains the requested AudioDecoder Configuration with the matching token.* tt:AudioDecoderConfiguration **Configuration** [1][1]
Fault codes	**Description**
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*

5.13.3 Get compatible audio decoder configurations

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. An 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.

Table 66: GetCompatibleAudioDecoderConfigurations

GetCompatibleAudioDecoderConfigurations	Access Class: READ_MEDIA
Message name	**Description**
GetCompatibleAudioDecoderConfigurations Request	*Contains the token of an existing media profile.* tt:ReferenceToken **ProfileToken** [1][1]
GetCompatibleAudioDecoderConfigurations Response	*Contains a list of audiodecoder configurations that are compatible with the given media profile.* tt:AudioDecoderConfiguration **Configurations** [0][unbounded]
Fault codes	**Description**
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*

5.13.4 Get audio decoder configuration options

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.

Table 67: GetAudioDecoderConfigurationOptions

GetAudioDecoderConfigurationOptions	Access Class: READ_MEDIA
Message name	**Description**
GetAudioDecoderConfiguration-OptionsRequest	*This message may contain a media profile or audio decoder configuration token, or both.* tt:ReferenceToken **ConfigurationToken** [0][1] tt:ReferenceToken **ProfileToken** [0][1]
GetAudioDecoderConfiguration-OptionsResponse	*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.* tt:AudioDecoderConfigurationOptions **Options** [1][1]
Fault codes	**Description**

env:Sender ter:InvalidArgVal ter:NoProfile	*The requested profile token ProfileToken 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*

5.13.5 Modify audio decoder configuration

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.

Table 68: SetAudioDecoderConfiguration

SetAudioDecoderConfiguration	Access Class: ACTUATE
Message name	Description
SetAudioDecoderConfiguration-Request	*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.* tt:AudioDecoderConfiguration **Configuration** [1][1] xs:boolean **ForcePersistence** [1][1]
SetAudioDecoderConfiguration-Response	*This message is empty.*
Fault codes	Description
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*

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 Request stream URI

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 69.

Table 69: Valid setup parameter conbinations

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.

Table 70: GetStreamUri command

GetStreamUri	Access Class: READ_MEDIA
Message name	**Description**
GetStreamUriRequest	*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.* *The **ProfileToken** element indicates the media profile to use and will define the configuration of the content of the stream.* tt:StreamSetup **StreamSetup** [1][1] tt:ReferenceToken **ProfileToken** [1][1]
GetStreamUriResponse	*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.* xs:anyURI **Uri** [1][1] xs:boolean **InvalidAfterConnect** [1][1] xs:boolean **InvalidAfterReboot** [1][1] xs:duration **Timeout** [1][1]
Fault codes	**Description**
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.*

5.16 Snapshot

5.16.1 Request snapshot URI

A Network client uses the GetSnapshotUri command to obtain a JPEG snhapshot 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.

Table 71: GetSnapshotUri command

GetSnapshotUri	Access Class: READ_MEDIA
Message name	**Description**
GetSnapshotUriRequest	*The **ProfileToken** element indicates the media profile to use and will define the source and dimensions of the snapshot.* tt:ReferenceToken **ProfileToken** [1][1]
GetSnapshotUriResponse	*Contains a stable **Uri** to be used for acquiring a snapshot in JPEG format as well as parameters defining the lifetime of the Uri. The **ValidUntilConnect** and **ValidUntilReboot** 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.* xs:anyURI **Uri** [1][1] xs:boolean **InvalidAfterConnect** [1][1] xs:boolean **InvalidAfterReboot** [1][1] xs:duration **Timeout** [1][1]
Fault codes	**Description**
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.*

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 Start multicast streaming

This command starts multicast streaming using a specified media profile of a device. Streaming continues until StopMulticastStreaming is called for the same Profile. The streaming shall 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.

Table 72: StartMulticastStreaming command

StartMulticastStreaming	Access Class: ACTUATE
Message name	**Description**
StartMulticastStreaming-Request	*Contains the token of the Profile that is used to define the multicast stream.* tt:ReferenceToken **ProfileToken** [1][1]
StartMulticastStreaming-Response	*This message is empty.*
Fault codes	**Description**
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*

5.17.2 Stop multicast streaming

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.

Table 73: StopMulticastStreaming command

StopMulticastStreaming	Access Class: ACTUATE
Message name	**Description**
StopMulticastStreaming-Request	*Contains the token of the Profile that is used to define the multicast stream.* tt:ReferenceToken **ProfileToken** [1][1]
StopMulticastStreaming-Response	*This message is empty.*
Fault codes	**Description**

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*

5.18 Synchronization Points

5.18.1 Set synchronization point

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.

Table 74: SetSynchronizationPoint command

SetSynchronizationPoint	Access Class: ACTUATE
Message name	**Description**
SetSynchronizationPointRequest	Contains a Profile reference for which a Synchronization Point is requested. tt:ReferenceToken **ProfileToken** [1][1]
SetSynchronizationPointResponse	This message is empty.
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NoProfile	The profile does not exist.

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.

Table 75: GetVideoSourceModes command

GetVideoSourceModes	Access Class: READ_SYSTEM
Message name	**Description**
GetVideoSourceModesRequest	*The request message specifies video source* *.* tt:ReferenceToken VideoSourceToken[1][1]
GetVideoSourceModesResponse	*The response contains list of mode information for seeing capabilities of video source.* trt:VideoSourceMode VideoSouceMode[1][unbounded]
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NoVideoSource	*The requested video source does not exist.*

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.

Table 76: SetVideoSourceMode command

SetVideoSourceMode	Access Class: WRITE_SYSTEM
Message name	**Description**
SetVideoSourceModeRequest	*The request message specifies video source.* tt:ReferenceToken VideoSourceToken[1][1] tt:ReferenceToken VideoSourceModeToken[1][1]
SetVideoSourceModeResponse	*The response contains information about rebooting after returning response. When Reboot is set "true", a device will reboot automatically after setting mode.* xs:boolean Reboot[1][1]
Fault codes	**Description**
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.*

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.

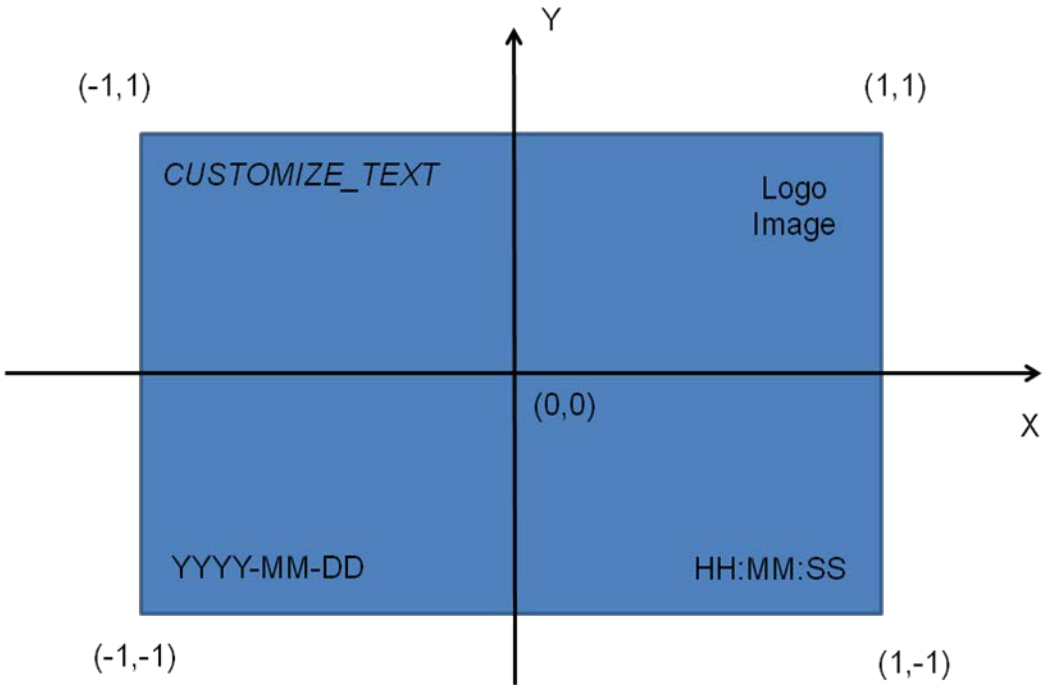

Figure 3: Example of screen which have four OSD configurations and coordinate system

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.

Table 77: CreateOSD command

CreateOSD	Access Class: ACTUATE
Message name	**Description**
CreateOSDRequest	*Contains a new OSD configuration with the specified value. The device is responsible for assigning OSD token. OSD token in CreateOSDRequest can be ignored.* tt:OSDConfiguration **OSD** [1][1]
CreateOSDResponse	*Return the newly created OSD token.* xs:string **Token**[1][1]
Fault codes	**Description**
env:Receiver ter:Action ter:MaxOSDs	*The maximum number of supported OSDs by the specific VideoSourceConfiguration has been reached.*

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.

Table 78: DeleteOSD command

DeleteOSD	Access Class: ACTUATE
Message name	**Description**
DeleteOSDRequest	*The request message contains an OSD token that indicate which OSD shall be deleted* tt:ReferenceToken **OSDToken**[1][1]
DeleteOSDResponse	*This is an empty message.*
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NoConfig	*The requested OSD token **OSDToken** does not exist.*

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.

Table 79: GetOSDs command

GetOSDs	Access Class: READ_MEDIA
Message name	**Description**
GetOSDsRequest	*The request message specifies the VideoSourceConfiguration token for which the OSD should be associated with.* tt:ReferenceToken **VideoSourceConfigurationToken** [0][1]
GetOSDsResponse	*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.* tt:OSDConfiguration **OSD**[0][unbounded]
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NoConfig	*The requested configuration indicated with **VideoSourceConfigurationToken** does not exist.*

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.

Table 80: GetOSD command

GetOSD	Access Class: READ_MEDIA
Message name	**Description**
GetOSDRequest	*This message contains the token of the requested OSD.* tt:ReferenceToken **OSDToken**[1][1]
GetOSDResponse	*The message contains the requested OSD with the matching token.* tt:OSDConfiguration **OSD**[1][1]
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NoConfig	*The requested configuration indicated with* **OSDToken** *does not exist.*

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.

Table 81: SetOSD command

SetOSD	Access Class: ACTUATE
Message name	**Description**
SetOSDRequest	*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* tt:OSDConfiguration **OSD**[1][1]
SetOSDResponse	*This message is empty.*
Fault codes	**Description**
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.*

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.

Table 82: GetOSDOptions command

GetOSDOptions	Access Class: READ_MEDIA
Message name	**Description**
GetOSDOptionsRequest	*The* ***VideoSourceConfigurationToken*** *element specifies the video source configuration of which the sutiable OSD options are requested. The* ***VideoSourceConfigurationToken*** *shall exist in the device* tt:ReferenceToken **VideoSourceConfigurationToken** [1][1]
GetOSDOptionsResponse	*This message contains the OSD options which is suitable for the video source configuration specified in the request* tt:OSDConfigurationOptions **Options**[1][1]
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NoConfig	*The requested video source configuration does not exist*

5.21 Capabilities

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

RTP_RTSP_TCP: Indication if the device supports RTP/RTSP/TCP transport, see Section 5.1.1.3 of the ONVIF Streaming Specificiation.

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: Indiates the support for temporary osd text configuration.

Table 83: GetServiceCapabilities command

GetServiceCapabilities	Access Class: PRE_AUTH
Message name	**Description**
GetServiceCapabilitiesRequest	*This is an empty message.*
GetServiceCapabilitiesResponse	*The capability response message contains the requested service capabilities using a hierarchical XML capability structure.* trt: Capabilities **Capabilities** [1][1]
Fault codes	**Description**
	No command specific faults!

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: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>
```

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-3108a16afaee" />
  </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>
```

5.23 Service specific data types

5.23.1 VideoSource

Representation of a physical video input.

```
<xs:complexType name="VideoSource">
  <xs:extension base= "tt:DeviceEntity/">
  <xs:element name="Framerate" type="xs:float"/>
  <xs:element name="Resolution" type= "tt:VideoResolution/>
  <xs:element name="Imaging" type= "tt:ImagingSettings minOccurs="0"/>
</xs:complexType>
```

- **Framerate**
  Frame rate in frames per second.

- **Resolution**
  Horizontal and vertical resolution

- **Imaging**
  Optional configuration of the image sensor.

5.23.2 AudioSource

Representation of a physical audio input.

```
<xs:complexType name="AudioSource">
  <xs:extension base= "tt:DeviceEntity"/>
  <xs:element name="Channels" type="xs:int"/>
</xs:complexType>
```

- **Channels**
  number of available audio channels. (1: mono, 2: stereo)

5.23.3 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.

```
<xs:complexType name="Profile">
  <xs:attribute name="token" type= "tt:ReferenceToken" use="required"/>
  <xs:attribute name="fixed" type="xs:boolean"/>
  <xs:element name="Name" type= "tt:Name"/>
  <xs:element name="VideoSourceConfiguration" type=
            "tt:VideoSourceConfiguration" minOccurs="0"/>
  <xs:element name="AudioSourceConfiguration" type=
            "tt:AudioSourceConfiguration" minOccurs="0"/>
  <xs:element name="VideoEncoderConfiguration" type=
            "tt:VideoEncoderConfiguration" minOccurs="0"/>
  <xs:element name="AudioEncoderConfiguration" type=
            "tt:AudioEncoderConfiguration" minOccurs="0"/>
  <xs:element name="VideoAnalyticsConfiguration" type=
            "tt:VideoAnalyticsConfiguration" minOccurs="0"/>
  <xs:element name="PTZConfiguration" type= "tt:PTZConfiguration"
            minOccurs="0"/>
  <xs:element name="MetadataConfiguration" type= "tt:MetadataConfiguration"
            minOccurs="0"/>
  <xs:element name="Extension" type= "tt:ProfileExtension" minOccurs="0"/>
            <xs:attribute name="token" type= "tt:ReferenceToken"
            use="required"/>
  <xs:attribute name="fixed" type="xs:boolean"/>
</xs:complexType>
```

- **token**
  Unique identifier of the profile.
- **fixed**
  A value of true signals that the profile cannot be deleted. Default is false.
- **Name**
  User readable name of the profile.
- **VideoSourceConfiguration**
  Optional configuration of the Video input.
- **AudioSourceConfiguration**
  Optional configuration of the Audio input.
- **VideoEncoderConfiguration**
  Optional configuration of the Video encoder.
- **AudioEncoderConfiguration**
  Optional configuration of the Audio encoder.
- **VideoAnalyticsConfiguration**
  Optional configuration of the video analytics module and rule engine.
- **PTZConfiguration**
  Optional configuration of the pan tilt zoom unit.

- **MetadataConfiguration**
  Optional configuration of the metadata stream.

- **Extension**
  Extensions defined in ONVIF 2.0

5.23.4 ProfileExtension

```
<xs:complexType name="ProfileExtension">
  <xs:element name="AudioOutputConfiguration" type=
            "tt:AudioOutputConfiguration" minOccurs="0"/>
  <xs:element name="AudioDecoderConfiguration" type=
            "tt:AudioDecoderConfiguration" minOccurs="0"/>
</xs:complexType>
```

- **AudioOutputConfiguration**
  Optional configuration of the Audio output.

- **AudioDecoderConfiguration**
  Optional configuration of the Audio decoder.

5.23.5 ConfigurationEntity

Base type defining the common properties of a configuration.

```
<xs:complexType name="ConfigurationEntity">
<xs:attribute name="token" type= "tt:ReferenceToken" use="required"/>
            <xs:element name="Name" type= "tt:Name"/>
            <xs:element name="UseCount" type="xs:int"/>
            <xs:attribute name="token" type= "tt:ReferenceToken"
            use="required"/>
</xs:complexType>
```

- **token**
  Token that uniquely refernces this configuration. Length up to 64 characters.

- **Name**
  User readable name. Length up to 64 characters.

- **UseCount**
  Number of internal references currently using this configuration.

5.23.6 VideoSourceConfiguration

```
<xs:complexType name="VideoSourceConfiguration">
<xs:extension base= "tt:ConfigurationEntity"/>
            <xs:element name="SourceToken" type="tt:ReferenceToken"/>
            <xs:element name="Bounds" type="tt:IntRectangle"/>
            <xs:element name="Extension"
            type="tt:VideoSourceConfigurationExtension" minOccurs="0"/>
</xs:complexType>
```

- **SourceToken**
  Reference to the physical input.

- **Bounds**
  Rectangle specifying the Video capturing area. The capturing area shall not be larger than the whole Video source area.

5.23.7 VideoSourceConfigurationExtension

```
<xs:complexType name="VideoSourceConfigurationExtension">
  <xs:element name="Rotate" type= "tt:Rotate” minOccurs="0"/>
</xs:complexType>
```

- **Rotate**
  Optional element to configure rotation of captured image.

5.23.8 Rotate

```
<xs:complexType name="Rotate">
   <xs:element name="Mode" type="tt:RotateMode"/>
   <xs:element name="Degree" type="xs:Int" minOccurs="0"/>
</xs:complexType>
```

- **Mode**
  Parameter to enable/disable Rotation feature.
    - ON: Enable the Rotate feature. Degree of rotation is specified Degree parameter.
    - OFF: Disable the Rotate feature
    - AUTO: Rotate feature is automatically activated by the device.
- **Degree**
  Optional parameter to configure how much degree of clockwise rotation of image for On mode. Omitting this parameter for On mode means 180 degree rotation.

What resolutions a device supports shall be unaffected by the Rotate parameters.
OSDs shall be unaffected by the Rotate parameters.

If a device is configured with Rotate=AUTO, the device shall take control over the Degree parameter and automatically update it so that a client can query current rotation.

The device shall automatically apply the same rotation to its pan/tilt control direction if the following condition is true.

- if Reverse=AUTO in PTControlDirection
- or if the device doesn't support Reverse in PTControlDirection

5.23.9 VideoSourceConfigurationOptions

```
<xs:complexType name="VideoSourceConfigurationOptions">
  <xs:element name="BoundsRange" type= "tt:IntRectangleRange"/>
  <xs:element name="VideoSourceTokensAvailable" type="tt:ReferenceToken"
            maxOccurs="unbounded"/>
  <xs:element name="Extension"
            type="tt:VideoSourceConfigurationOptionsExtension"
            minOccurs="0"/>
</xs:complexType>
```

- **BoundsRange**
  Supported range for the capturing area.
- **VideoSourceTokensAvailable**
  List of physical inputs.

5.23.10 VideoSourceConfigurationOptionsExtension

```
<xs:complexType name="VideoSourceConfigurationOptionsExtension">
  <xs:element name="Rotate" type="tt:RotateOptions" minOccurs="0"/>
</xs:complexType>
```

- **Rotate**
  Options of parameters for Rotation feature.

5.23.11 RotateOptions

```
<xs:complexType name="RotateOptions">
```

```
<xs:element name="Mode" type="tt:RotateMode" maxOccurs="unbounded"/>
            <xs:element name="DegreeList" type="tt:IntList" minOccurs="0"/>
</xs:complexType>
```

- **Mode**
  Supported options of Rotate mode parameter.

- **DegreeList**
  List of supported degree value for rotation.

5.23.12 VideoEncoderConfiguration

- **Encoding**
  Used video codec, either Jpeg, H.264 or Mpeg4

- **Resolution**
  Configured video resolution

- **Quality**
  Relative value for the video quantizers and the quality of the video. A high value within supported quality range means higher quality

- **RateControl**
  Optional element to configure rate control related parameters.

- **MPEG4**
  Optional element to configure Mpeg4 related parameters.

- **H264**
  Optional element to configure H.264 related parameters.

- **Multicast**
  Defines the multicast settings that could be used for video streaming.

- **SessionTimeout**
  The SessionTimeout is provided as a hint for keeping rtsp session by a device. If necessary the device may adapt parameter values for SessionTimeout elements without returning an error.
  For the time between keep alive calls the client shall adhere to the timeout value signaled via RTSP.

  - **GuaranteedFrameRate**
    A value of true indicates that frame rate is a fixed value rather than an upper limit, and that the video encoder shall prioritize frame rate over all other adaptable configuration values such as bitrate. Default is false.

5.23.13 VideoResolution

```
<xs:complexType name="VideoResolution"/>
<xs:element name="Width" type="xs:int"/>
            <xs:element name="Height" type="xs:int"/>
</xs:complexType>
```

- **Width**
  Number of the columns of the Video image.

- **Height**
  Number of the lines of the Video image.

5.23.14 VideoRateControl

```
<xs:complexType name="VideoRateControl"/>
```

```
<xs:element name="FrameRateLimit" type="xs:int"/>
            <xs:element name="EncodingInterval" type="xs:int"/>
            <xs:element name="BitrateLimit" type="xs:int"/>
</xs:complexType>
```

- **FrameRateLimit**
  Maximum output framerate in fps. If an EncodingInterval is provided the resulting encoded framerate will be reduced by the given factor.

- **EncodingInterval**
  Interval at which images are encoded and transmitted. (A value of 1 means that every frame is encoded, a value of 2 means that every 2nd frame is encoded ...)

- **BitrateLimit**
  the maximum output bitrate in kbps

5.23.15 Mpeg4Configuration

```
<xs:complexType name="Mpeg4Configuration">
  <xs:element name="GovLength" type="xs:int"/>
            <xs:element name="Mpeg4Profile" type="tt:Mpeg4Profile"/>
</xs:complexType>
```

- **GovLength**
  Determines the interval in which the I-frames will be coded. An entry of 1 indicates I-frames are continuously generated. An entry of 2 indicates that every 2nd image is an I-frame, and 3 only every 3rd frame, etc. The frames in between are coded as P or B Frames.

- **Mpeg4Profile**
  the Mpeg4 profile, either simple profile (SP) or advanced simple profile (ASP)

5.23.16 H264Configuration

```
<xs:complexType name="H264Configuration">
  <xs:element name="GovLength" type="xs:int"/>
  <xs:element name="H264Profile" type="tt:H264Profile"/>
</xs:complexType>
```

- **GovLength**
  Group of Video frames length. Determines typically the interval in which the I-frames will be coded. An entry of 1 indicates I-frames are continuously generated. An entry of 2 indicates that every 2nd image is an I-frame, and 3 only every 3rd frame, etc. The frames in between are coded as P or B Frames.

- **H264Profile**
  the H.264 profile, either baseline, main, extended or high

5.23.17 VideoEncoderConfigurationOptions

```
<xs:complexType name="VideoEncoderConfigurationOptions">
  <xs:element name="QualityRange" type= "tt:IntRange"/>
  <xs:element name="JPEG" type="tt:JpegOptions" minOccurs="0"/>
  <xs:element name="MPEG4" type="tt:Mpeg4Options" minOccurs="0"/>
  <xs:element name="H264" type="tt:H264Options" minOccurs="0"/>
  <xs:element name="Extension" type="tt:VideoEncoderOptionsExtension"
            minOccurs="0"/>
</xs:complexType>
```

- **QualityRange**
  Range of the quality values. A high value means higher quality.

- **JPEG**
  Optional JPEG encoder settings ranges (See also Extension element).

- **MPEG4**
  Optional MPEG-4 encoder settings ranges (See also Extension element).

- **H264**
  Optional H.264 encoder settings ranges (See also Extension element).

o **GuaranteedFrameRateSupported**
  Indicates the support for the GuaranteedFrameRate attribute on the VideoEncoderConfiguration element.

5.23.18 VideoEncoderOptionsExtension

```
<xs:complexType name="VideoEncoderOptionsExtension">
  <xs:element name="JPEG" type= "tt:JpegOptions2" minOccurs="0"/>
  <xs:element name="MPEG4" type= "tt:Mpeg4Options2" minOccurs="0"/>
  <xs:element name="H264" type= "tt:H264Options2" minOccurs="0"/>
</xs:complexType>
```

- **JPEG**
  Optional JPEG encoder settings ranges.

- **MPEG4**
  Optional MPEG-4 encoder settings ranges.

- **H264**
  Optional H.264 encoder settings ranges.

5.23.19 JpegOptions

```
<xs:complexType name="JpegOptions">
  <xs:element name="ResolutionsAvailable" type= "tt:VideoResolution"
            maxOccurs="unbounded"/>
  <xs:element name="FrameRateRange" type= "tt:IntRange"/>
  <xs:element name="EncodingIntervalRange" type= "tt:IntRange"/>
</xs:complexType>
```

- **ResolutionsAvailable**
  List of supported image sizes.

- **FrameRateRange**
  Supported frame rate in fps (frames per second).

- **EncodingIntervalRange**
  Supported encoding interval range. The encoding interval corresponds to the number of frames devided by the encoded frames. An encoding interval value of "1" means that all frames are encoded.

5.23.20 JpegOptions2

```
<xs:complexType name="JpegOptions2">
  <xs:extension base= "tt:JpegOptions"/>
  <xs:element name="BitrateRange" type= "tt:IntRange"/>
</xs:complexType>
```

- **BitrateRange**
  Supported range of encoded bitrate in kbps.

5.23.21 Mpeg4Options

```
<xs:complexType name="Mpeg4Options">
  <xs:element name="ResolutionsAvailable" type= "tt:VideoResolution"
            maxOccurs="unbounded"/>
  <xs:element name="GovLengthRange" type= "tt:IntRange"/>
  <xs:element name="FrameRateRange" type= "tt:IntRange"/>
```

```
  <xs:element name="EncodingIntervalRange" type= "tt:IntRange"/>
  <xs:element name="Mpeg4ProfilesSupported" type="tt:Mpeg4Profile"
            maxOccurs="unbounded"/>
</xs:complexType>
```

- **ResolutionsAvailable**
  List of supported image sizes.

- **GovLengthRange**
  Supported group of Video frames length. This value typically corresponds to the I-frame distance.

- **FrameRateRange**
  Supported frame rate in fps (frames per second).

- **EncodingIntervalRange**
  Supported encoding interval range. The encoding interval corresponds to the number of frames devided by the encoded frames. An encoding interval value of "1" means that all frames are encoded.

- **Mpeg4ProfilesSupported**
  List of supported MPEG-4 profiles.

5.23.22 Mpeg4Options2

```
<xs:complexType name="Mpeg4Options2">
  <xs:extension base= "tt:Mpeg4Options"/>
  <xs:element name="BitrateRange" type= "tt:IntRange"/>
</xs:complexType>
```

- **BitrateRange**
  Supported range of encoded bitrate in kbps.

5.23.23 H264Options

```
<xs:complexType name="H264Options">
  <xs:element name="ResolutionsAvailable" type= "tt:VideoResolution"
            maxOccurs="unbounded"/>
  <xs:element name="GovLengthRange" type= "tt:IntRange"/>
  <xs:element name="FrameRateRange" type= "tt:IntRange"/>
  <xs:element name="EncodingIntervalRange" type= "tt:IntRange"/>
  <xs:element name="H264ProfilesSupported" type="tt:H264Profile"
            maxOccurs="unbounded"/>
</xs:complexType>
```

- **ResolutionsAvailable**
  List of supported image sizes.

- **GovLengthRange**
  Supported group of Video frames length. This value typically corresponds to the I-frame distance.

- **FrameRateRange**
  Supported frame rate in fps (frames per second).

- **EncodingIntervalRange**
  Supported encoding interval range. The encoding interval corresponds to the number of frames devided by the encoded frames. An encoding interval value of "1" means that all frames are encoded.

- **H264ProfilesSupported**
  List of supported H.264 profiles.

5.23.24 H264Options2

```
<xs:complexType name="H264Options2">
  <xs:extension base= "tt:H264Options"/>
  <xs:element name="BitrateRange" type= "tt:IntRange"/>
</xs:complexType>
```

- **BitrateRange**
  Supported range of encoded bitrate in kbps.

5.23.25 AudioSourceConfiguration

```
<xs:complexType name="AudioSourceConfiguration">
  <xs:extension base= "tt:ConfigurationEntity"/>
  <xs:element name="SourceToken" type= "tt:ReferenceToken"/>
</xs:complexType>
```

- **SourceToken**
  Token of the Audio Source the configuration applies to

5.23.26 AudioSourceConfigurationOptions

```
<xs:complexType name="AudioSourceConfigurationOptions">
  <xs:element name="InputTokensAvailable" type= "tt:ReferenceToken"
            maxOccurs="unbounded"/>
</xs:complexType>
```

- **InputTokensAvailable**
  Tokens of the audio source the configuration can be used for.

5.23.27 AudioEncoderConfiguration

```
<xs:complexType name="AudioEncoderConfiguration">
  <xs:extension base= "tt:ConfigurationEntity"/>
  <xs:element name="Encoding" type="tt:AudioEncoding"/>
  <xs:element name="Bitrate" type="xs:int"/>
  <xs:element name="SampleRate" type="xs:int"/>
  <xs:element name="Multicast" type="tt:MulticastConfiguration"/>
  <xs:element name="SessionTimeout" type="xs:duration"/>
</xs:complexType>
```

- **Encoding**
  Audio codec used for encoding the audio input (either G.711, G.726 or AAC)

- **Bitrate**
  The output bitrate in kbps.

- **SampleRate**
  The output sample rate in kHz.

- **Multicast**
  Defines the multicast settings that could be used for video streaming.

- **SessionTimeout**
  The rtsp session timeout for the related audio stream

5.23.28 AudioEncoderConfigurationOptions

```
<xs:complexType name="AudioEncoderConfigurationOptions">
  <xs:element name="Options" type= "tt:AudioEncoderConfigurationOption"
            minOccurs="0" maxOccurs="unbounded"/>
</xs:complexType>
```

- **Options**
  list of supported AudioEncoderConfigurations

5.23.29 AudioEncoderConfigurationOption

```
<xs:complexType name="AudioEncoderConfigurationOption">
  <xs:element name="Encoding" type="tt:AudioEncoding"/>
  <xs:element name="BitrateList" type= "tt:IntList"/>
  <xs:element name="SampleRateList" type= "tt:IntList"/>
</xs:complexType>
```

- **Encoding**
  The enoding used for audio data (either G.711, G.726 or AAC)

- **BitrateList**
  List of supported bitrates in kbps for the specified Encoding

- **SampleRateList**
  List of supported Sample Rates in kHz for the specified Encoding

5.23.30 VideoAnalyticsConfiguration

```
<xs:complexType name="VideoAnalyticsConfiguration">
  <xs:extension base= "tt:ConfigurationEntity"/>
  <xs:element name="AnalyticsEngineConfiguration"
            type="tt:AnalyticsEngineConfiguration"/>
  <xs:element name="RuleEngineConfiguration"
            type="tt:RuleEngineConfiguration"/>
</xs:complexType>
```

- **AnalyticsEngineConfiguration**

- **RuleEngineConfiguration**

5.23.31 MetadataConfiguration

```
<xs:complexType name="MetadataConfiguration">
  <xs:extension base= "tt:ConfigurationEntity"/>
  <xs:element name="PTZStatus" type= "tt:PTZFilter" minOccurs="0"/>
  <xs:element name="Events" type= "tt:EventSubscription" minOccurs="0"/>
  <xs:element name="Analytics" type="xs:boolean" minOccurs="0"/>
  <xs:element name="Multicast" type="tt:MulticastConfiguration"/>
  <xs:element name="SessionTimeout" type="xs:duration"/>
  <xs:element name="AnalyticsEngineConfiguration"
            type="tt:AnalyticsEngineConfiguration" minOccurs="0"/>
  <xs:attribute name="CompressionType" type="xs:string"/>
</xs:complexType>
```

- **PTZStatus**
  optional element to configure which PTZ related data is to include in the metadata stream

- **Events**
  Optional element to configure the streaming of events. A client might be interested in receiving all, none or some of the events produced by the device:

  - To get all events: Include the Events element but do not include a filter element.

  - To get no events: Do not include the Events element.

  - To get only some events: Include the Events element and include a filter in the element.

- **Analytics**
  Defines if data to include from the analytics engine part shall be included in the stream

- **Multicast**
  Defines the multicast settings that could be used for video streaming.

- **SessionTimeout**
  The rtsp session timeout for the related audio stream

- **AnalyticsEngineConfiguration**
  Optional parameter to configure analytics engine.

- **CompressionType**
  Optional parameter to configure compression type of Metadata payload

5.23.32 PTZFilter

```
<xs:complexType name="PTZFilter">
  <xs:element name="Status" type="xs:boolean"/>
  <xs:element name="Position" type="xs:boolean"/>
</xs:complexType>
```

- **Status**
  True if the metadata stream shall contain the PTZ status (IDLE, MOVING or UNKNOWN)

- **Position**
  True if the metadata stream shall contain the PTZ position

5.23.33 EventSubscription

Subcription handling in the same way as base notification subscription.

```
<xs:complexType name="EventSubscription">
  <xs:element name="Filter" type="wsnt:FilterType" minOccurs="0"/>
  <xs:element name="SubscriptionPolicy" minOccurs="0"/>
</xs:complexType>
```

- **Filter**

- **SubscriptionPolicy**

5.23.34 MetadataConfigurationOptions

```
<xs:complexType name="MetadataConfigurationOptions">
  <xs:element name="PTZStatusFilterOptions"
            type="tt:PTZStatusFilterOptions"/>
  <xs:element name="Extension"
            type="tt:MetadataConfigurationOptionsExtension"/>
</xs:complexType>
```

- **PTZStatusFilterOptions**

5.23.35 MetadataConfigurationOptionsExtension

```
<xs:complexType name="MetadataConfigurationOptions">
  <xs:element name="CompressionType" type="xs:string"
            minOccurs="0" maxOccurs="unbounded"/>
</xs:complexType>
```

- **CompressionType**
  List of supported metadata compression type. Its options shall be chosen from tt:MetadataCompressionType.

5.23.36 PTZStatusFilterOptions

```
<xs:complexType name="PTZStatusFilterOptions">
  <xs:element name="PanTiltStatusSupported" type="xs:boolean"/>
  <xs:element name="ZoomStatusSupported" type="xs:boolean"/>
  <xs:element name="PanTiltPositionSupported" type="xs:boolean"
            minOccurs="0"/>
```

```
  <xs:element name="ZoomPositionSupported" type="xs:boolean"
            minOccurs="0"/>
</xs:complexType>
```

- **PanTiltStatusSupported**
  True if the device is able to stream pan or tilt status information.

- **ZoomStatusSupported**
  True if the device is able to stream zoom status inforamtion.

- **PanTiltPositionSupported**
  True if the device is able to stream the pan or tilt position.

- **ZoomPositionSupported**
  True if the device is able to stream zoom position information.

5.23.37 VideoOutput

Representation of a physical video outputs.

```
<xs:complexType name="VideoOutput">
  <xs:extension base= "tt:DeviceEntity"/>
  <xs:element name="Layout" type="tt:Layout"/>
</xs:complexType>
```

- **Layout**

5.23.38 VideoOutputConfiguration

```
<xs:complexType name="VideoOutputConfiguration">
  <xs:extension base= "tt:ConfigurationEntity"/>
</xs:complexType>
```

5.23.39 VideoDecoderConfigurationOptions

```
<xs:complexType name="VideoDecoderConfigurationOptions">
  <xs:element name="JpegDecOptions" type= "tt:JpegDecOptions"
            minOccurs="0"/>
  <xs:element name="H264DecOptions" type= "tt:H264DecOptions"
            minOccurs="0"/>
  <xs:element name="Mpeg4DecOptions" type= "tt:Mpeg4DecOptions"
            minOccurs="0"/>
</xs:complexType>
```

- **JpegDecOptions**
  If the device is able to decode Jpeg streams this element describes the supported codecs and configurations

- **H264DecOptions**
  If the device is able to decode H.264 streams this element describes the supported codecs and configurations

- **Mpeg4DecOptions**
  If the device is able to decode Mpeg4 streams this element describes the supported codecs and configurations

5.23.40 H264DecOptions

```
<xs:complexType name="H264DecOptions">
  <xs:element name="ResolutionsAvailable" type= "tt:VideoResolution"
            maxOccurs="unbounded"/>
  <xs:element name="SupportedH264Profiles" type="tt:H264Profile"
            maxOccurs="unbounded"/>
  <xs:element name="SupportedInputBitrate" type= "tt:IntRange"/>
  <xs:element name="SupportedFrameRate" type= "tt:IntRange"/>
```

```
</xs:complexType>
```

- **ResolutionsAvailable**
  List of supported H.264 Video Resolutions

- **SupportedH264Profiles**
  List of supported H264 Profiles (either baseline, main, extended or high)

- **SupportedInputBitrate**
  Supported H.264 bitrate range in kbps

- **SupportedFrameRate**
  Supported H.264 framerate range in fps

5.23.41 JpegDecOptions

```
<xs:complexType name="JpegDecOptions">
  <xs:element name="ResolutionsAvailable" type= "tt:VideoResolution"
            maxOccurs="unbounded"/>
  <xs:element name="SupportedInputBitrate" type= "tt:IntRange"/>
  <xs:element name="SupportedFrameRate" type= "tt:IntRange"/>
</xs:complexType>
```

- **ResolutionsAvailable**
  List of supported Jpeg Video Resolutions

- **SupportedInputBitrate**
  Supported Jpeg bitrate range in kbps

- **SupportedFrameRate**
  Supported Jpeg framerate range in fps

5.23.42 Mpeg4DecOptions

```
<xs:complexType name="Mpeg4DecOptions">
  <xs:element name="ResolutionsAvailable" type= "tt:VideoResolution"
            maxOccurs="unbounded"/>
  <xs:element name="SupportedMpeg4Profiles" type="tt:Mpeg4Profile"
            maxOccurs="unbounded"/>
  <xs:element name="SupportedInputBitrate" type= "tt:IntRange"/>
  <xs:element name="SupportedFrameRate" type= "tt:IntRange"/>
</xs:complexType>
```

- **ResolutionsAvailable**
  List of supported Mpeg4 Video Resolutions

- **SupportedMpeg4Profiles**
  List of supported Mpeg4 Profiles (either SP or ASP)

- **SupportedInputBitrate**
  Supported Mpeg4 bitrate range in kbps

- **SupportedFrameRate**
  Supported Mpeg4 framerate range in fps

5.23.43 AudioOutput

Representation of a physical audio outputs.

```
<xs:complexType name="AudioOutput">
  <xs:extension base= "tt:DeviceEntity"/>
</xs:complexType>
```

5.23.44 AudioOutputConfiguration

```
<xs:complexType name="AudioOutputConfiguration">
  <xs:extension base= "tt:ConfigurationEntity"/>
  <xs:element name="OutputToken" type= "tt:ReferenceToken"/>
  <xs:element name="SendPrimacy" type="xs:anyURI" minOccurs="0"/>
  <xs:element name="OutputLevel" type="xs:int"/>
</xs:complexType>
```

- **OutputToken**
  Token of the phsycial Audio output.

- **SendPrimacy**
  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. The optional SendPrimacy 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.

- **OutputLevel**
  Volume setting of the output. The applicable range is defined via the option AudioOutputOptions.OutputLevelRange.

5.23.45 AudioOutputConfigurationOptions

```
<xs:complexType name="AudioOutputConfigurationOptions">
  <xs:element name="OutputTokensAvailable" type= "tt:ReferenceToken"
            maxOccurs="unbounded"/>
  <xs:element name="SendPrimacyOptions" type="xs:anyURI" minOccurs="0"
            maxOccurs="unbounded"/>
  <xs:element name="OutputLevelRange" type= "tt:IntRange"/>
</xs:complexType>
```

- **OutputTokensAvailable**
  Tokens of the physical Audio outputs (typically one).

- **SendPrimacyOptions**
  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. The optional SendPrimacy 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.

- **OutputLevelRange**
  Minimum and maximum level range supported for this Output.

5.23.46 AudioDecoderConfiguration

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

```
<xs:complexType name="AudioDecoderConfiguration">
  <xs:extension base= "tt:ConfigurationEntity"/>
</xs:complexType>
```

5.23.47 AudioDecoderConfigurationOptions

```
<xs:complexType name="AudioDecoderConfigurationOptions">
  <xs:element name="AACDecOptions" type= "tt:AACDecOptions" minOccurs="0"/>
            <xs:element name="G711DecOptions" type= "tt:G711DecOptions"
            minOccurs="0"/>
  <xs:element name="G726DecOptions" type= "tt:G726DecOptions"
            minOccurs="0"/>
</xs:complexType>
```

- **AACDecOptions**
  If the device is able to decode AAC encoded audio this section describes the supported configurations
- **G711DecOptions**
  If the device is able to decode G711 encoded audio this section describes the supported configurations
- **G726DecOptions**
  If the device is able to decode G726 encoded audio this section describes the supported configurations

5.23.48 G711DecOptions

```
<xs:complexType name="G711DecOptions">
  <xs:element name="Bitrate" type= "tt:IntList"/>
  <xs:element name="SampleRateRange" type= "tt:IntList"/>
</xs:complexType>
```

- **Bitrate**
  List of supported bitrates in kbps
- **SampleRateRange**
  List of supported sample rates in kHz

5.23.49 AACDecOptions

```
<xs:complexType name="AACDecOptions">
  <xs:element name="Bitrate" type= "tt:IntList"/>
  <xs:element name="SampleRateRange" type= "tt:IntList"/>
```

```
</xs:complexType>
```

- **Bitrate**
  List of supported bitrates in kbps

- **SampleRateRange**
  List of supported sample rates in kHz

5.23.50 G726DecOptions

```
<xs:complexType name="G726DecOptions">
  <xs:element name="Bitrate" type= "tt:IntList"/>
  <xs:element name="SampleRateRange" type= "tt:IntList"/>
</xs:complexType>
```

- **Bitrate**
  List of supported bitrates in kbps

- **SampleRateRange**
  List of supported sample rates in kHz

5.23.51 MulticastConfiguration

```
<xs:complexType name="MulticastConfiguration">
 <xs:element name="Address" type= "tt:IPAddress"/>
  <xs:element name="Port" type="xs:int"/>
  <xs:element name="TTL" type="xs:int"/>
  <xs:element name="AutoStart" type="xs:boolean"/>
</xs:complexType>
```

- **Address**
  The multicast address (if this address is set to 0 no multicast streaming is enaled)

- **Port**
  The RTP mutlicast destination port. A device may support RTCP. In this case the port value shall be even to allow the corresponding RTCP stream to be mapped to the next higher (odd) destination port number as defined in the RTSP specification.

- **TTL**
  The TTL value that should be used for the multicast stream

- **AutoStart**
  Read only property signalling that streaming is persistant. Use the methods StartMulticastStreaming and StopMulticastStreaming to switch its state.

An ONVIF Device supporting Multicast transport shall support any mix of valid Multicast address and port independent of the address and port configured in the other entities of the unit as long as each address and port configuration is unique. A device may accept the same IP address and port for different multicast configurations. Note that the port should be set to an even number as defined in RFC 3550.

5.23.52 StreamSetup

```
<xs:complexType name="StreamSetup">
  <xs:element name="Stream" type="tt:StreamType"/>
  <xs:element name="Transport" type= "tt:Transport"/>
</xs:complexType>
```

- **Stream**
  Defines if a multicast or unicast stream is requested

- **Transport**

5.23.53 Transport

```
<xs:complexType name="Transport">
  <xs:element name="Protocol" type="tt:TransportProtocol"/>
  <xs:element name="Tunnel" type= "tt:Transport" minOccurs="0"/>
</xs:complexType>
```

- **Protocol**
  Defines the network protocol for streaming, either RTP/UDP, RTP/TCP, RTP/RTSP/TCP or RTP/RTSP/HTTP/TCP

- **Tunnel**
  Optional element to describe further tunnel options. This element is normally not needed

5.23.54 MediaUri

```
<xs:complexType name="MediaUri">
  <xs:element name="Uri" type="xs:anyURI"/>
  <xs:element name="InvalidAfterConnect" type="xs:boolean"/>
  <xs:element name="InvalidAfterReboot" type="xs:boolean"/>
  <xs:element name="Timeout" type="xs:duration"/>
</xs:complexType>
```

- **Uri**
  Stable Uri to be used for requesting the media stream

- **InvalidAfterConnect**
  Indicates if the Uri is only valid until the connection is established. The value shall be set to "false".

- **InvalidAfterReboot**
  Indicates if the Uri is invalid after a reboot of the device. The value shall be set to "false".

- **Timeout**
  Duration how long the Uri is valid. This parameter shall be set to PT0S to indicate that this stream URI is indefinitely valid even if the profile changes

5.23.55 Video Source Mode

```
<xs:complexType name="VideoSourceMode">
    <xs:element name=“MaxFramerate" type="xs:float"/>
    <xs:element name=“MaxResolution" type="tt:VideoResolution”/>
    <xs:element name=“Encodings" type=“trt:EncodingTypes”/>
    <xs:element name=“Reboot" type=“xs:boolean”/>
    <xs:element name=“Description" type="tt:Description” minOccurs="0"/>
    <xs:attribute name="token" type="tt:ReferenceToken" use="required">
    <xs:attribute name=“Enabled" type="xs:boolean"/>
</xs:complexType>
```

- **MaxFramerate**
  Max frame rate in frames per second for this video source mode.

- **MaxResolution**
  Max horizontal and vertical resolution for this video source mode.

- **Encodings**
  Indication which encodings are supported for this video source. The list may contain one or more enumeration values of tt:VideoEncoding.

- **Reboot**
  After setting the mode if a device starts to reboot this value is “true”. If a device change the mode without rebooting this value is “false”. If “true“, configured parameters may not be guaranteed by the device after rebooting.

- **Description**
  Informative description of this video source mode. This field should be described in English.

- **token**
  Indicate token for video source mode.

- **Enabled**
  Indication of whether this mode is active. If active this value is "true". In case of non-indication, it means as "false". The value of "true" shall be had by only one video source mode.

5.23.56 OSDPosConfiguration

```
<xs:complexType name="OSDPosConfiguration">
  <xs:sequence>
    <xs:element name="Type" type="xs:string"/>
    <xs:element name="Pos" type="tt:Vector" minOccurs="0"/>
  </xs:sequence>
<xs:complexType>
```

- **Type**
  The type of the OSD position. Following are the pre-defined: UpperLeft, UpperRight, LowerLeft, LowerRight, or Custom.

- **Pos**
  The value of the OSD position described by x[-1,1] and y[-1,1]. It shall be present when the value of Type is Custom.

5.23.57 OSDTextConfiguration

```
<xs:complexType name="OSDTextConfiguration">
  <xs:element name="Type" type="xs:string" minOccurs="0"/>
  <xs:element name="DateFormat" type="xs:string" minOccurs="0"/>
  <xs:element name="TimeFormat" type="xs:string" minOccurs="0"/>
  <xs:element name="FontSize" type="xs:int" minOccurs="0"/>
  <xs:element name="FontColor" type="tt:OSDColor" minOccurs="0"/>
  <xs:element name="BackgroundColor" type="tt:OSDColor" minOccurs="0"/>
  <xs:element name="PlainText" type="xs:string" minOccurs="0"/>
  <xs:element name="Extension" type="tt:OSDTextConfigurationExtension"
            minOccurs="0"/>
</xs:complexType>
```

- **Type**
  The type of the text show on the screen. The following OSD Text Type are defined:

  - Plain - The Plain type means the OSD is shown as a text string which defined in the "PlainText" item.
  - Date - The Date type means the OSD is shown as a date, format of which should be present in the "DateFormat" item.
  - Time - The Time type means the OSD is shown as a time, format of which should be present in the "TimeFormat" item.
  - DateAndTime - The DateAndTime type means the OSD is shown as date and time, format of which should be present in the "DateFormat" and the "TimeFormat" item.

- **DateFormat**
  The format of the date. It shall be present when the value of Type field is Date or DateAndTime.

- **TimeFormat**
  The format of the time. It shall be present when the value of Type field is Time or DateAndTime.

- **FontSize**
  The text font size in pt.

- **FontColor**
  The color of the text font.

- **BackgroundColor**
  The background color of the text.

- **PlainText**
  The plain text. It shall be present when the value of Type field is Plain.

- **Extension**

5.23.58 OSDImgConfiguration

```
<xs:complexType name="OSDImgConfiguration">
  <xs:element name="ImgPath" type="xs:anyURI"/>
</xs:complexType>
```

- **ImgPath**
  The path of the image show on the screen.

5.23.59 OSDTextOptions

```
<xs:complexType name="OSDTextOptions">
  <xs:element name="Type" type="xs:string" maxOccurs="unbounded"/>
  <xs:element name="FontSizeRange" type="tt:IntRange" minOccurs="0"/>
  <xs:element name="DateFormat" type="xs:string" minOccurs="0"
            maxOccurs="unbounded"/>
  <xs:element name="TimeFormat" type="xs:string" minOccurs="0"
            maxOccurs="unbounded"/>
  <xs:element name="FontColor" type="tt:OSDColorOptions" minOccurs="0"/>
  <xs:element name="BackgroundColor" type="tt:OSDColorOptions"
            minOccurs="0"/>
</xs:complexType>
```

- **Type**
  List of supported OSD text type. When a device indicates the supported number relating to Text type in MaximumNumberOfOSDs, the type shall be presented.

- **FontSizeRange**
  Supported font size in pt.

- **DateFormat**
  List of supported OSD date formats. This element shall be present when the value of Type field has Date or DateAndTime.

- **TimeFormat**
  List of supported OSD time formats. This element shall be present when the value of Type field has Time or DateAndTime.

- **FontColor**
  List of supported font color

- **BackgroundColor**
  List of supported background color

5.23.60 OSDImgOptions

```
<xs:complexType name="OSDImgOptions">
  <xs:element name="ImagePath" type="xs:anyURI" maxOccurs="unbounded"/>
</xs:complexType>
```

- **ImagePath**
  List available image path.

5.23.61 OSDColorOptions

```
<xs:complexType name="OSDColorOptions">
  <xs:element name="Color" type="tt:ColorOptions" minOccurs="0">
  <xs:element name="Transparent" type="tt:IntRange" minOccurs="0">
</xs:complexType>
```

- **Color**
  Optional list of supported colors.

- **Transparent**
  The value range of "Transparent" could be defined by vendors and should follow this rule: the minimum value means non-transparent and the maximum value maens fully transparent.

5.23.62 OSDConfiguration

```
<xs:complexType name="OSDConfiguration">
  <xs:extension base="tt:DeviceEntity"/>
  <xs:element name="VideoSourceConfigurationToken" type="tt:OSDReference"/>
  <xs:element name="Type" type="tt:OSDType"/>
  <xs:element name="Position" type="tt:OSDPosConfiguration"/>
   <xs:element name="TextString" type="tt:OSDTextConfiguration"
           minOccurs="0"/>
  <xs:element name="Image" type="tt:OSDImgConfiguration" minOccurs="0"/>
</xs:complexType>
```

- **VideoSourceConfigurationToken**
  The VideoSourceConfiguration which OSD is applied to.

- **Type**
  OSD type, either Text or Image.

- **Position**
  OSD position configuration.

- **TextString**
  Text configuration of OSD. It shall be present when the value of Type field is Text.

- **Image**
  Image configuration of OSD. It shall be present when the value of Type field is Image.

5.23.63 OSDConfigurationOptions

```
<xs:complexType name="OSDConfigurationOptions">
  <xs:element name=”MaximumNumberOfOSDs” type=”xs:int”>
    <xs:attribute name="Total" type="xs:int" use="required"/>
    <xs:attribute name="Image" type="xs:int"/>
    <xs:attribute name="PlainText" type="xs:int"/>
    <xs:attribute name="Date" type="xs:int"/>
    <xs:attribute name="Time" type="xs:int"/>
    <xs:attribute name="DateAndTime" type="xs:int"/>
  </element>
  <xs:element name="Type" type="tt:OSDType" maxOccurs="unbounded"/>
  <xs:element name="PositionOption" type="xs:string"
maxOccurs="unbounded"/>
  <xs:element name="TextOption" type="tt:OSDTextOptions" minOccurs="0"/>
  <xs:element name="ImageOption" type="tt:OSDImgOptions" minOccurs="0"/>
</xs:complexType>
```

- **MaximumNumberOfOSDs**
  The maximum number of OSD configurations supported for the specified video source

configuration. If the configuration does not support OSDs, this value shall be zero and the Type and PositionOption elements are ignored. If a device limits the number of instances by OSDType, it shall indicate the supported number for each type via the related attribute.

- **Type**
  List supported type of OSD configuration. When a device indicates the supported number for each types in MaximumNumberOfOSDs, related type shall be presented. A device shall return Option element relating to listed type.

- **PositionOption**
  List available OSD position type. Following are the pre-defined: UpperLeft, UpperRight, LowerLeft, LowerRight, or Custom.

- **TextOption**
  Option of the OSD text configuration. This element shall be returned if the device is signaling the support for Text.

- **ImageOption**
  Option of the OSD image configuration. This element shall be returned if the device is signaling the support for Image.

5.24 Service specific fault codes

The table below lists the media service specific fault codes. Additionally, each command can also generate a generic fault..

The specific faults are defined as subcode of a generic fault.The parent generic subcode is the *subcode* at the top of each row below and the specific fault *subcode* is at the bottom of the cell.

Table 84: Media service specific fault codes

Fault Code	Parent Subcode Subcode	Fault Reason	Description
env:Receiver	ter:ActionNotSupported ter:AudioNotSupported	No audio capability	The device does not support audio.
env:Receiver	ter:Action ter:MaxNVTProfiles	Maximum number reached	The maximum number of supported profiles has been reached.
env:Receiver	ter:ActionNotSupported ter:AudioOutputNotSupported	No audio output capability	Audio or Audio Outputs are not supported by the device
env:Receiver	ter:ActionNotSupported ter:AudioDecodingNotSupported	No audio decoding capability	Audio or Audio Decoding is not supported by the device
env:Receiver	ter:Action ter:IncompleteConfiguration	Configuration not complete	Entities required by this action are missing in the specified profile.
env:Receiver	ter:Action ter:ConfigurationConflict	Conflict when using new settings	The new settings conflicts with other uses of the configuration.

env:Receiver	ter:Action ter:MaxOSDs	Reach the maximum number of OSD	The maximum number of the OSD supported by the specified VideoSourceConfiguration has been reached.
env:Sender	ter:InvalidArgVal ter:ProfileExists	Profile token already exists	A profile with the token ProfileToken already exists.
env:Sender	ter:InvalidArgVal ter:NoConfig	Configuration token does not exist	The requested configuration indicated by the ConfigurationToken does not exist.
env:Sender	ter:InvalidArgVal ter:NoProfile	Profile token does not exist	The requested profile token ProfileToken does not exist.
env:Sender	ter:Action ter:DeletionOfFixedProfile	Fixed profile can not be deleted	The fixed Profile cannot be deleted.
env:Sender	ter:InvalidArgVal ter:ConfigModify	Parameters can not be set	The configuration parameters are not possible to set.
env:Sender	ter:ActionNotSupported ter:VideoAnalyticsNot-Supported	No video analytics capability	The device does not support video analytics.
env:Sender	ter:InvalidArgVal ter:InvalidStreamSetup	Invalid Stream setup	Specification of StreamType or Transport part in StreamSetup is not supported.
env:Sender	ter:OperationProhibited ter:StreamConflict	Stream conflict	Specification of StreamType or Transport part in StreamSetup causes conflict with other streams.
env:Sender	ter:InvalidArgVal ter:InvalidMulticastSettings	Invalid multicast configuration	Not all configurations are configured for multicast..
env:Sender	ter:InvalidArgVal ter:NoVideoSourceMode	Video source mode token does not exist.	The requested video source mode does not exist

Annex A. Bibliography

[ONVIF Display WSDL] ONVIF Media WSDL, ver 2.0, 2010.
URL:http://www.onvif.org/onvif/ver10/network/wsdl/media.wsdl
[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