

ONVIF™ Media 2 Service Specification

Version 16.12

December, 2016



© 2008-2016 by ONVIF: Open Network Video Interface Forum Inc.. All rights reserved.

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

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

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

CONTENTS

1	Scope	6
2	Normative references	6
3	Terms and Definitions	6
3.1	Definitions.....	6
3.2	Abbreviations	6
4	Overview	6
4.1	Media profiles.....	7
4.2	Video source mode	9
5	Service	9
5.1	Media Profile Methods	10
5.1.1	Create media profile	10
5.1.2	Get media profiles.....	11
5.1.3	Add one or more configurations to a profile.....	11
5.1.4	Remove one or more configurations from a profile	12
5.1.5	Delete media profile.....	13
5.2	Media Configurations	13
5.2.1	General	13
5.2.2	Video source configuration	14
5.2.3	Video encoder configuration.....	14
5.2.4	Audio source configuration	14
5.2.5	Audio encoder configuration.....	14
5.2.6	PTZ Configuration.....	15
5.2.7	Analytics Configuration	15
5.2.8	Metadata Configuration	15
5.2.9	Audio output configuration.....	15
5.2.10	Audio decoder configuration.....	16
5.3	Media Configuration Methods	16
5.3.1	General	16
5.3.2	Get configurations.....	16
5.3.3	Modify a configuration.....	18
5.3.4	Get configuration options.....	19
5.3.5	Get video encoder instance information	20
5.4	Stream URI.....	21
5.4.1	Request stream URI	21
5.5	Snapshot	22
5.5.1	Request snapshot URI.....	22
5.6	Multicast	22
5.6.1	Start multicast streaming	22
5.6.2	Stop multicast streaming	23
5.7	Synchronization Points.....	23
5.7.1	Set synchronization point.....	23
5.8	Video source mode	24
5.8.1	GetVideoSourceModes.....	24
5.8.2	SetVideoSourceMode.....	25
5.9	OSD (On-Screen Display).....	26
5.9.1	General	26
5.9.2	CreateOSD	26
5.9.3	DeleteOSD.....	27
5.9.4	GetOSDs.....	27
5.9.5	SetOSD.....	28
5.9.6	GetOSDOptions	29

5.10	Privacy Masks	30
5.10.1	General	30
5.10.2	Create Mask	30
5.10.3	Delete Mask	31
5.10.4	GetMasks	31
5.10.5	SetMask	32
5.10.6	GetMaskOptions	32
5.11	Capabilities	34
5.12	Events	35
5.12.1	Profile Change	35
5.12.2	Configuration Change	35
5.12.3	Active Connections	35
5.13	Service specific data types	36
5.13.1	MediaProfile	36
5.13.2	ConfigurationSet	36
5.13.3	ConfigurationEntity	37
5.13.4	VideoSourceConfiguration	37
5.13.5	VideoSourceConfigurationExtension	37
5.13.6	VideoSourceConfigurationExtension2	37
5.13.7	LensDescription	38
5.13.8	LensOffset	38
5.13.9	LensProjection	38
5.13.10	Rotate	38
5.13.11	VideoSourceConfigurationOptions	39
5.13.12	VideoSourceConfigurationOptionsExtension	39
5.13.13	RotateOptions	39
5.13.14	SceneOrientation	39
5.13.15	VideoSourceConfigurationOptionsExtension2	40
5.13.16	VideoEncoder2Configuration	40
5.13.17	VideoResolution2	40
5.13.18	VideoRateControl2	41
5.13.19	VideoEncoder2ConfigurationOptions	41
5.13.20	EncoderInstanceInfo	41
5.13.21	EncoderInstance	42
5.13.22	AudioSourceConfiguration	42
5.13.23	AudioSourceConfigurationOptions	42
5.13.24	AudioEncoder2Configuration	42
5.13.25	AudioEncoder2ConfigurationOptions	42
5.13.26	MetadataConfiguration	43
5.13.27	PTZFilter	43
5.13.28	EventSubscription	43
5.13.29	MetadataConfigurationOptions	44
5.13.30	MetadataConfigurationOptionsExtension	44
5.13.31	PTZStatusFilterOptions	44
5.13.32	AudioOutputConfiguration	44
5.13.33	AudioOutputConfigurationOptions	45
5.13.34	AudioDecoderConfiguration	45
5.13.35	MulticastConfiguration	45
5.13.36	StreamSetup	45
5.13.37	Transport	46
5.13.38	Video Source Mode	46
5.13.39	OSDPosConfiguration	46
5.13.40	OSDTextConfiguration	47
5.13.41	OSDImgConfiguration	47
5.13.42	OSDTextOptions	47
5.13.43	OSDImgOptions	48
5.13.44	OSDColorOptions	48
5.13.45	OSDConfiguration	49
5.13.46	OSDConfigurationOptions	49
5.13.47	Mask	50

5.13.48	MaskOptions.....	50
5.14	Service specific fault codes.....	51
Annex A. Efficient XML Interchange (EXI) (Normative)		53
Annex B. Lens Description (Normative)		54
Annex C. Revision History		56

1 Scope

This document defines the second generation 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

IANA Media Type Reference

< <http://www.iana.org/assignments/media-types/media-types.xhtml>>

ONVIF Core Specification

<<http://www.onvif.org/onvif/specs/core/ONVIF-Core-Specification-v211.pdf>>

ONVIF PTZ Service Specification

<<http://www.onvif.org/onvif/specs/srv/ptz/ONVIF-PTZ-Service-Spec-v211.pdf>>

ONVIF Streaming Specification

<<http://www.onvif.org/onvif/specs/stream/ONVIF-Streaming-Spec-v211.pdf>>

3 Terms and Definitions

3.1 Definitions

Configuration Entity	A network video device media abstract component that produces or consumes a media stream on the network, i.e. video and/or audio stream.
Digital PTZ	Function that diminishes or crops an image to adjust the image position and ratio.
GZIP	GNU data format for lossless compression.
Media Profile	Maps a video and audio sources and outputs encoders as well as 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).
Reference Token	Token provided by the device to uniquely reference an instance of a physical IO, configuration or profile.
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

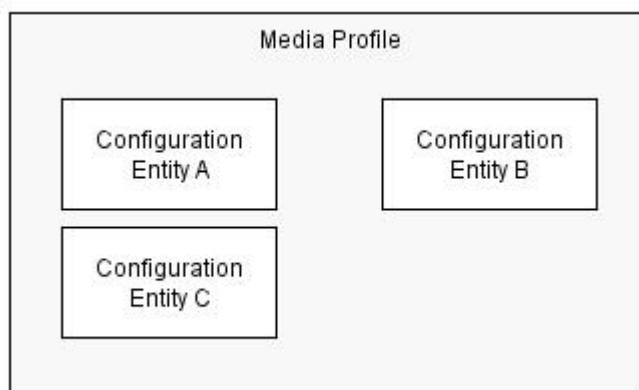
Media configurations are handled through the media service. Media configurations are used to determine the streaming properties of requested media streams as defined in this specification. The device provides media configuration through the media service. WSDL for this service is specified in <http://www.onvif.org/ver20/media/wsdll/media.wsdll>.

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
tr2	http://www.onvif.org/ver20/media/wsdl
tns1	http://www.onvif.org/ver10/topics

4.1 Media profiles

Real-time video and audio streaming configurations are controlled using media profiles. A media profile maps a video and/or audio source to a video and/or an audio encoder, PTZ and analytics configurations. An ONVIF compliant device supporting the media service presents different available profiles depending on its capabilities (the set of available profiles might change dynamically though).

**Figure 1: A media profile**

A device 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.

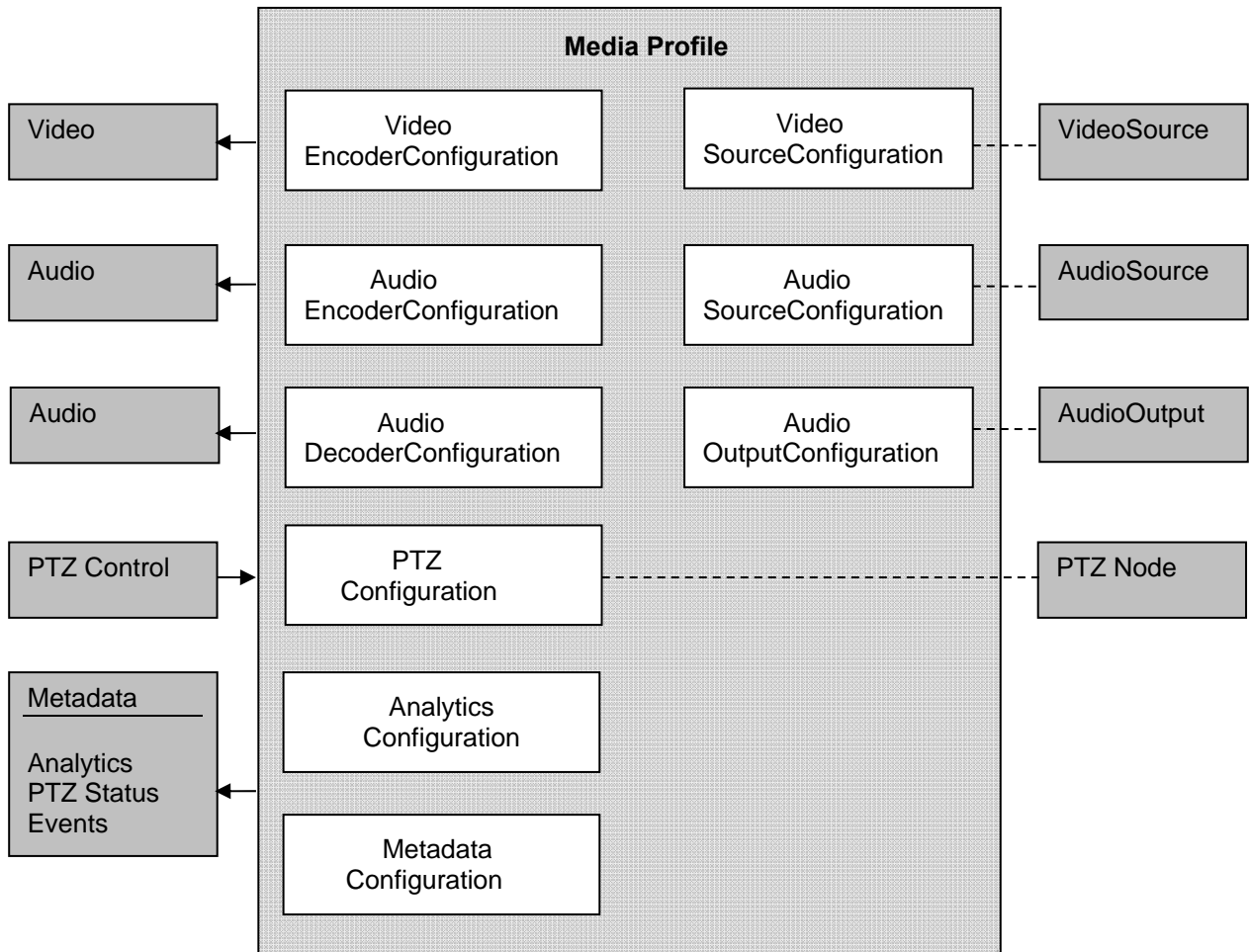


Figure 2: Complete profile configuration

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 manage Media Profiles:

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

- *RemoveConfiguration* – Removes one or more configuration entities from a media profile.

The following commands manage Configuration Entities:

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

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

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

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

4.2 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

- Video encoder commands
- Audio source commands
- Audio encoder commands
- Video analytics commands
- Metadata commands
- Audio output commands
- Audio decoder commands
- Media streaming:
 - Request stream URI
 - Get snapshot URI
 - Multicast control commands
 - Media synchronization point

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

5.1 Media Profile Methods

5.1.1 Create media profile

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

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

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

Table 2: CreateProfile command

CreateProfile		Access Class: ACTUATE
Message name	Description	
CreateProfileRequest	<p>Contains the user defined Name of the Profile to be created as well as an optional list of Configuration's.</p> <p>tt:Name Name [1][1] tr2:ConfigurationRef Configuration[0][unbounded]</p>	
CreateProfileResponse	<p>Returns the reference token of the newly created profile.</p> <p>tt:ReferenceToken Token [1][1]</p>	
Fault codes	Description	
env:Receiver ter:Action ter:MaxNVTPProfiles	<p>The maximum number of supported profiles supported by the device has been reached.</p>	

5.1.2 Get media profiles

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

The token parameter controls which profiles are returned:

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

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

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

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

Table 3: GetProfiles command

GetProfiles		Access Class: READ_MEDIA
Message name	Description	
GetProfilesRequest	<p><i>This message may contain the token to the requested profile. If one or more types are passed only the corresponding configurations will be returned.</i></p> <p>tt:ReferenceToken Token [0][1] xs:string Type[0][unbounded]</p>	
GetProfilesResponse	<p><i>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.</i></p> <p>tr2:MediaProfile Profiles [0][unbounded]</p>	
Fault codes	Description	
env:Sender ter:InvalidArgVal ter:NoProfile	<p><i>The requested profile token Token does not exist.</i></p>	

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

5.1.3 Add one or more configurations to a profile

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

Table 4: AddConfiguration command

AddConfiguration		Access Class: ACTUATE
Message name	Description	
AddConfiguration Request	<p>Contains a reference to the Configuration to be added and the Profile where it shall be added. Optionally the Name of the profile can be updated via the Name parameter.</p> <p>tt:ReferenceToken ProfileToken [1][1] tt:Name Name[0][1] tr2:ConfigurationRef Configuration [0][unbounded]</p>	
AddConfiguration Response	<p>This is an empty message.</p>	
Fault codes	Description	
env:Sender ter:InvalidArgVal ter:NoProfile	<p>The requested profile token passed via the parameter ProfileToken does not exist.</p>	
env:Sender ter:InvalidArgVal ter:NoConfig	<p>One of the provided Configurations indicated by their ConfigurationToken does not exist.</p>	
env:Receiver ter:Action ter:ConfigurationConflict	<p>Other configurations of the media profile conflicts with the one to add and adding it would cause a conflicting media profile.</p>	

5.1.4 Remove one or more configurations from a profile

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

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

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

Table 5: RemoveConfiguration command

RemoveConfiguration		Access Class: ACTUATE
Message name	Description	
RemoveConfigurationRequest	<p>Contains a reference to the media profile from which the <Entity>Configuration shall be removed.</p> <p>tt:ReferenceToken ProfileToken [1][1] tr2:ConfigurationRef Configuration [1][unbounded]</p>	
RemoveConfigurationResponse	<p>This is an empty message.</p>	
Fault codes	Description	

env:Sender ter:InvalidArgVal ter:NoProfile	<i>The requested profile passed via the parameter ProfileToken does not exist.</i>
env:Receiver ter:Action ter:ConfigurationConflict	<i>Other configurations of the media profile are dependant on this configuration and removing it would cause a conflicting state of the media profile.</i>

5.1.5 Delete media profile

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

Table 6: DeleteProfile command

DeleteProfile		Access Class: ACTUATE
Message name	Description	
DeleteProfileRequest	<i>The Token references the media profile to delete.</i> tt:ReferenceToken Token [1][1]	
DeleteProfileResponse	<i>This is an empty message.</i>	
Fault codes	Description	
env:Sender ter:InvalidArgVal ter:NoProfile	<i>The requested profile token does not exist.</i>	
env:Sender ter:Action ter:DeletionOfFixedProfile	<i>A fixed Profile cannot be deleted.</i>	

5.2 Media Configurations

5.2.1 General

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

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

A profile consists of a set of interconnected *configuration entities*. Configurations are provided by the device and can either be static or created dynamically by the device. For example, the dynamic configurations can be created by the device depending on current available encoding resources. See the following subsections for configuration entity specific constraints.

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

All configurations have the following base parameters

- Token – The identifier of the configuration. This parameter is readonly and cannot be modified by clients.
- Name – A configurable name of up to 64 characters.

- UseCount – Readonly property indicating the number of referenced profiles. Deprecated in Media2 – devices may support this value.

5.2.2 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.

The Rotate option specifies an optional rotation of the Video.

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

The Lens Description option allows to describe the geometric distortion of the Video Source. For details see Annex B.

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

5.2.3 Video encoder configuration

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

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

TheVideoEncoderConfiguration structure also contains multicast parameters.

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

5.2.4 Audio source configuration

An Audio Source Configuration contains a reference to the AudioSource.

5.2.5 Audio encoder configuration

An AudioEncoderConfiguration contains the following parameters for encoding audio data:

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

The AudioEncoderConfiguration structure also contains multicast parameters.

5.2.6 PTZ Configuration

A profile with a PTZConfiguration enables the streaming of PTZ status in the metadata stream. Additionally the media profile can be used for controlling PTZ movement as defined in the PTZ Service Specification.

5.2.7 Analytics Configuration

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

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

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

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

5.2.8 Metadata Configuration

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

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 5.2.7.

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

5.2.9 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 5.2.9.1).

5.2.9.1 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.

5.2.10 Audio decoder configuration

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

5.3 Media Configuration Methods

5.3.1 General

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

5.3.2 Get configurations

The syntax and semantics of the request message are the same for all configuration entities:

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

Table 7: Get<entity>Configurations command

Get<entity>Configurations		Access Class: READ_MEDIA
Message name	Description	
Get<entity>ConfigurationsRequest	<i>This message may contain a profile or configuration token.</i> tt:ReferenceToken ProfileToken [0][1] tt:ReferenceToken ConfigurationToken [0][1]	

Get<entity>Configurations-Response	<i>This message contains a list of the respective configurations.</i> tt:<entity>Configuration Configurations [0][unbounded]
Fault codes	Description
env:Sender ter:InvalidArgVal ter:NoProfile	<i>The requested profile does not exist.</i>
env:Sender ter:InvalidArgVal ter:NoConfig	<i>The requested configuration does not exist in the given context.</i>

5.3.3 Modify a configuration

This operation modifies a configuration. The change may have immediate effect to running streams but the changes are not guaranteed to take effect unless the client restarts any affected stream. Client methods for changing a running stream are out of scope for this specification.

Table 8: Set<entity>Configuration command

Set<entity>Configuration		Access Class: ACTUATE
Message name	Description	
Set<entity>Configuration-Request	<p>The Configuration element contains the modified configuration. The configuration shall exist in the device.</p> <p>tt:<entity>Configuration Configuration [1][1]</p>	
Set<entity>ConfigurationResponse	<p>This message is empty.</p>	
Fault codes	Description	
env:Sender ter:InvalidArgVal ter:NoConfig	<p>The requested configuration does not exist.</p>	
env:Sender ter:InvalidArgVal ter:ConfigModify	<p>The configuration parameters are not possible to set.</p>	
env:Receiver ter:Action ter:ConfigurationConflict	<p>The new settings conflicts with other uses of the configuration.</p>	

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

Note for the AudioSourceConfiguration: If the new settings invalidate any parameters already negotiated using RTSP, for example by changing codec type, the device 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.

5.3.4 Get configuration options

This operation returns the available options for a configuration to inform a client about the available parameters and their valid ranges. Any combination of the parameters obtained using a given profile and configuration shall be a valid input for the corresponding set configuration command.

If a configuration token is provided, the device shall return the compatible options for the 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.

Table 9: Get<entity>ConfigurationOptions command

Get<entity>ConfigurationOptions		Access Class: READ_MEDIA
Message name	Description	
Get<entity>ConfigurationOptionsRequest	<p><i>This message contains optional tokens of a video encoder configuration and a media profile.</i></p> <p>ConfigurationToken specifies an existing configuration that the options are intended for.</p> <p>ProfileToken specifies an existing media profile that the options shall be compatible with.</p> <p>tt:ReferenceToken ConfigurationToken [0][1] tt:ReferenceToken ProfileToken [0][1]</p>	
Get<entity>ConfigurationOptionsResponse	<p><i>This message contains configuration options. Depending on the structure of tr2:Get<entity>ConfigurationOptionsResponse, the options will return one or more elements:</i></p> <p>tt:<entity>ConfigurationOptions Options [1][unbounded]</p> <p><i>Note: entities AudioOutput, AudioSource, Metadata, and VideoSource return just one element.</i></p>	
Fault codes	Description	
env:Sender ter:InvalidArgVal ter:NoProfile	The requested profile does not exist.	
env:Sender ter:InvalidArgVal ter:NoConfig	The requested configuration does not exist in the given context.	

5.3.5 Get video encoder instance information

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

The Info response contains the following information:

- Total Total number of encoder instances independent of the codec,
- Codec Number of encoder instances for each supported codec .

A device shall guarantee to be able to instantiate the signalled amount of instances concurrently. If a device limits the number of instances for respective Video Codecs the response contains shall also contain information per video encoding.

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

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

Table 10: GetVideoEncoderInstances command

GetVideoEncoderInstances		Access Class: READ_MEDIA
Message name	Description	
GetVideoEncoder-InstancesRequest	<i>This request contains a token to the video source configuration.</i> tt: ReferenceToken ConfigurationToken [1][1]	
GetVideoEncoder-InstancesResponse	<i>This message contains the minimum guaranteed total number of encoder instances (applications) per VideoSourceConfiguration.</i> tr2 EncoderInstanceInfo Info [1][1]	
Fault codes	Description	
<i>env:Sender ter:InvalidArgVal ter:NoConfig</i>	<i>The requested configuration indicated with ConfigurationToken does not exist.</i>	

5.4 Stream URI

5.4.1 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 parameters of the profile are changed.

The following stream types are defined

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

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

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

Table 11: GetStreamUri command

GetStreamUri		Access Class: READ_MEDIA
Message name	Description	
GetStreamUriRequest	<p>The Protocol defines how the encoded data is expected to be streamed to the client.</p> <p>The ProfileToken element indicates the media profile to use and will define the configuration of the content of the stream.</p> <p>xs:string Protocol [1][1] tt:ReferenceToken ProfileToken [1][1]</p>	
GetStreamUriResponse	<p>Contains the stable Uri to be used for requesting the media stream.</p> <p>xs:anyURI Uri [1][1]</p>	
Fault codes	Description	
env:Sender ter:InvalidArgVal ter:NoProfile	The media profile does not exist.	
env:Sender ter:InvalidArgVal ter:InvalidStreamSetup	The specified Protocol is not supported.	
env:Sender ter:OperationProhibited ter:StreamConflict	The specified Protocol causes a conflict with another stream.	
env:Receiver ter:Action ter:IncompleteConfiguration	The specified media profile does contain either unused sources or encoder configurations without a corresponding source.	
env:Sender ter:InvalidArgVal ter:InvalidMulticastSettings	Not all configurations are configured for multicast.	

5.5 Snapshot

5.5.1 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 parameters change. The URI can be used for acquiring one or more JPEG images 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 shall support this command when the SnapshotUri capability is set to true.

Table 12: GetSnapshotUri command

GetSnapshotUri		Access Class: READ_MEDIA
Message name	Description	
GetSnapshotUriRequest	<p>The ProfileToken element indicates the media profile to use and will define the source and dimensions of the snapshot.</p> <p>tt:ReferenceToken ProfileToken [1][1]</p>	
GetSnapshotUriResponse	<p>Contains a stable Uri to be used for acquiring a snapshot in JPEG format as well as parameters defining the lifetime of the Uri.</p> <p>xs:anyURI Uri [1][1]</p>	
Fault codes	Description	
env:Sender ter:InvalidArgVal ter:NoProfile	<p>The media profile does not exist.</p>	
env:Receiver ter:Action ter:IncompleteConfiguration	<p>The specified media profile does not contain either a reference to a video encoder configuration or a reference to a video source configuration.</p>	

5.6 Multicast

This specification defines two mechanisms for controlling multicast streams. In addition to setting up multicast sessions via RTSP sessions as defined in section 5.4.1 this chapter defines a multicast streaming mechanism where the actual streaming is controlled via IGMP. Use this method with caution, since an incorrect network configuration may result in flooding the network with Audio and Video packets.

A device that signals support for multicast streaming by the RTPMulticast capability shall support the methods defined in this chapter.

5.6.1 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 be resumed after rebooting. It can be turned off using the StopMulticastStreaming method. The multicast address, port and TTL are configured in the VideoEncoderConfiguration, AudioEncoderConfiguration and MetadataConfiguration respectively.

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

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

Table 13: StartMulticastStreaming command

StartMulticastStreaming		Access Class: ACTUATE
Message name	Description	
StartMulticastStreaming-Request	<i>Contains the token of the Profile that is used to define the multicast stream.</i> tt:ReferenceToken ProfileToken [1][1]	
StartMulticastStreaming-Response	<i>This message is empty.</i>	
Fault codes	Description	
env:Sender ter:InvalidArgVal ter:NoProfile	<i>The profile does not exist.</i>	
env:Receiver ter:Action ter:IncompleteConfiguration	<i>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</i>	

5.6.2 Stop multicast streaming

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

Table 14: StopMulticastStreaming command

StopMulticastStreaming		Access Class: ACTUATE
Message name	Description	
StopMulticastStreaming-Request	<i>Contains the token of the Profile that is used to define the multicast stream.</i> tt:ReferenceToken ProfileToken [1][1]	
StopMulticastStreaming-Response	<i>This message is empty.</i>	
Fault codes	Description	
env:Sender ter:InvalidArgVal ter:NoProfile	<i>The profile does not exist.</i>	
env:Receiver ter:Action ter:IncompleteConfiguration	<i>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</i>	

5.7 Synchronization Points

5.7.1 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 forces the device to add an I-frame as soon as possible. Clients can request Synchronization Points for profiles. The device shall add synchronization points for all streams associated with this profile.

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

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

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

Table 15: 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.	

5.8 Video source mode

A device may have the ability to change its video source mode. Different video source modes may affect which options are available in GetVideoSourceConfigurationOptions and GetVideoEncoderConfigurationOptions, such as only allowing 16x9 aspect ratios in one mode and only allowing 4x3 aspect ratios in another, or only allowing 30fps-derived frame rates in one mode and only allowing 25fps-derived frame rates in another. The GetVideoSourceModes command provides summary information about the different modes supported by the device.

5.8.1 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 16: GetVideoSourceModes command

GetVideoSourceModes		Access Class: READ_SYSTEM
Message name	Description	
GetVideoSourceModesRequest	<i>The request message specifies video source</i> tt:ReferenceToken VideoSourceToken[1][1]	
GetVideoSourceModesResponse	<i>The response contains list of mode information for seeing capabilities of video source.</i>	

	tr2:VideoSourceMode VideoSourceMode[1][unbounded]
Fault codes	Description
env:Sender ter:InvalidArgVal ter:NoVideoSource	<i>The requested video source does not exist.</i>

5.8.2 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 17: SetVideoSourceMode command

SetVideoSourceMode	Access Class: WRITE_SYSTEM
Message name	Description
SetVideoSourceModeRequest	<i>The request message specifies video source.</i> tt:ReferenceToken VideoSourceToken[1][1] tt:ReferenceToken VideoSourceModeToken[1][1]
SetVideoSourceModeResponse	<i>The response contains information about rebooting after returning response. When Reboot is set “true”, a device will reboot automatically after setting mode.</i> xs:boolean Reboot[1][1]
Fault codes	Description
env:Sender ter:InvalidArgVal ter:NoVideoSource	<i>The requested video source does not exist.</i>
env:Sender ter:InvalidArgVal ter:NoVideoSourceMode	<i>The requested video source mode does not exist.</i>

5.9 OSD (On-Screen Display)

5.9.1 General

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.

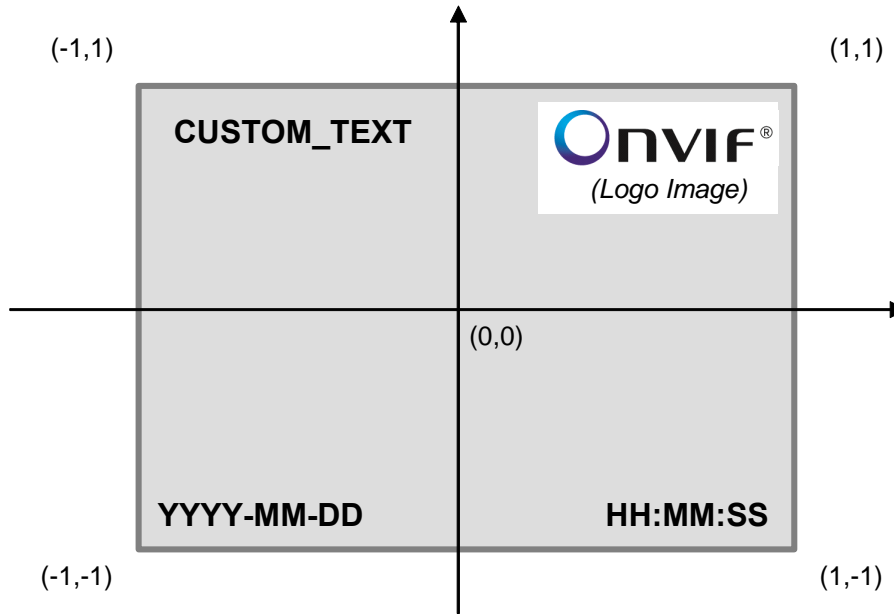


Figure 3: Example of screen which have four OSD configurations and coordinate system

This chapter defines methods to create and delete OSD configurations as well as getting, setting and querying the options. A device that signals support for OSD via the OSD capability shall support all OSD methods defined in section 5.9.

5.9.2 CreateOSD

This operation creates a new OSD configuration with specified values and also makes the association between the new OSD and an existing VideoSourceConfiguration identified by 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. A device that indicates OSD capability shall support the creation of OSD as long as the number of existing OSDs does not exceed the value of MaximumNumberOfOSDs in GetOSDOptions.

A created OSD configuration shall be deletable.

Table 18: CreateOSD command

CreateOSD		Access Class: ACTUATE
Message name	Description	
CreateOSDRequest	<p>Contains a new OSD configuration with the specified value. The device is responsible for assigning OSD token. OSD token in CreateOSDRequest can be ignored.</p> <p>tt:OSDConfiguration OSD [1][1]</p>	
CreateOSDResponse	<p>Return the newly created OSD token.</p> <p>xs:string OSDToken[1][1]</p>	
Fault codes	Description	
env:Receiver ter:Action ter:MaxOSDs	<p>The maximum number of supported OSDs by the specific VideoSourceConfiguration has been reached.</p>	
env:Sender ter:InvalidArgVal ter:NoConfig	<p>The requested the video source configuration does not exist</p>	
env:Sender ter:InvalidArgVal ter:ConfigModify	<p>The configuration parameters are not possible to set.</p>	

5.9.3 DeleteOSD

This operation deletes an OSD configuration.

Table 19: DeleteOSD command

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

5.9.4 GetOSDs

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

- If an OSD token is provided the device shall respond with the requested configuration or provide an error if it does not exist.

- In case only a video source configuration token is provided the device shall respond with all configurations that exist for the video source configuration.
- If no tokens are provided the device shall respond with all available OSD configurations.

Table 20: GetOSDs command

GetOSDs		Access Class: READ_MEDIA
Message name	Description	
GetOSDsRequest	<i>The request message may specify the VideoSourceConfiguration or OSD token for which matching OSD should be retrieved.</i> tt:ReferenceToken OSDToken [0][1] tt:ReferenceToken ConfigurationToken [0][1]	
GetOSDsResponse	<i>The message contains the requested OSD with the matching token.</i> tt:OSDConfiguration OSDs [0][unbounded]	
Fault codes	Description	
env:Sender ter:InvalidArgVal ter:NoConfig	<i>The requested video source configuration or OSD indicated with OSDToken does not exist.</i>	

5.9.5 SetOSD

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

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

Table 21: SetOSD command

SetOSD		Access Class: ACTUATE
Message name	Description	
SetOSDRequest	<i>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</i> tt:OSDConfiguration OSD [1][1]	
SetOSDResponse	<i>This message is empty.</i>	
Fault codes	Description	
env:Sender ter:InvalidArgVal ter:NoConfig	<i>The requested OSD does not exist</i>	
env:Sender ter:InvalidArgVal ter:ConfigModify	<i>The configuration parameters are not possible to set.</i>	

--	--

5.9.6 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 22: GetOSDOptions command

GetOSDOptions		Access Class: READ_MEDIA
Message name	Description	
GetOSDOptionsRequest	<p>The ConfigurationToken element specifies the video source configuration of which the suitable OSD options are requested. The Video Source Configuration Token shall exist in the device</p> <p>tt:ReferenceToken ConfigurationToken [1][1]</p>	
GetOSDOptionsResponse	<p>This message contains the OSD options which are suitable for the video source configuration specified in the request</p> <p>tt:OSDConfigurationOptions OSDOptions[1][1]</p>	
Fault codes	Description	
env:Sender ter:InvalidArgVal ter:NoConfig	<p>The requested video source configuration does not exist</p>	

5.10 Privacy Masks

5.10.1 General

Privacy Masks allow to cover regions of the image. The device signals via the options how many masks and how many edges each mask may support. Additionally devices may restrict to support rectangle masks only. Figure 3 shows how the mask is defined by a polygon using normalized coordinates relative to the VideoSourceConfiguration window.

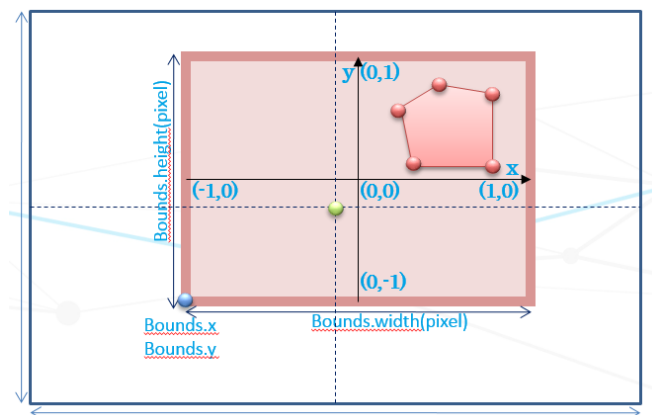


Figure 3: Example of screen which have four OSD configurations and coordinate system

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

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

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

5.10.2 Create Mask

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

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

Table 23: CreateMask command

CreateMask		Access Class: ACTUATE
Message name	Description	
CreateMaskRequest	Contains the new mask configuration. The device is responsible for assigning the mask token. tr2:Mask Mask [1][1]	
CreateMaskResponse	Return the newly created mask token. xs:string Token [1][1]	
Fault codes	Description	
env:Receiver ter:Action ter:MaxMasks	The maximum number of supported masks by the specific VideoSourceConfiguration has been reached.	
env:Sender ter:InvalidArgVal ter:NoConfig	The specified video source configuration does not exist.	

env:Sender ter:InvalidArgVal ter:InvalidPolygon	<i>The provided polygon is not supported.</i>
---	---

5.10.3 Delete Mask

This operation deletes a mask configuration.

Table 24: DeleteMask command

DeleteMask		Access Class: ACTUATE
Message name	Description	
DeleteMaskRequest	<i>The request message contains a mask token that indicate which Mask shall be deleted</i> tt:ReferenceToken Token [1][1]	
DeleteMaskResponse	<i>This is an empty message.</i>	
Fault codes	Description	
env:Sender ter:InvalidArgVal ter:NoConfig	<i>The requested Mask token Token does not exist.</i>	

5.10.4 GetMasks

This operation lists existing Mask configurations for the device. A device signalling support for the Mask capability shall support the listing of existing Mask configurations through this command.

- If a mask token is provided the device shall respond with the requested configuration or provide an error if it does not exist.
- In case only a video source configuration token is provided the device shall respond with all configurations that exist for the video source configuration.
- If no tokens are provided the device shall respond with all available Mask configurations.

Table 25: GetMasks command

GetMasks		Access Class: READ_MEDIA
Message name	Description	
GetMasksRequest	<i>The request message may specify the VideoSourceConfiguration or Mask token for which matching Mask should be retrieved.</i> tt:ReferenceToken MaskToken [0][1] tt:ReferenceToken ConfigurationToken [0][1]	
GetMasksResponse	<i>The message contains the requested Mask with the matching token.</i> tr2:Mask Mask [0][unbounded]	
Fault codes	Description	
env:Sender ter:InvalidArgVal ter:NoConfig	<i>The requested video source configuration or Mask indicated with MaskToken does not exist.</i>	

5.10.5 SetMask

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

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

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

Note: A device signalling RectangleOnly shall accept any polygon with four points. In case the four vertices are not defining an exact rectangle the device may adjust the vertices.

Table 26: SetMask command

SetMask		Access Class: ACTUATE
Message name	Description	
SetMaskRequest	<p>The Mask element contains the modified Mask configuration. The Configuration contains an element that specifies the Mask whose configuration is to be modified.</p> <p>tr2:Mask Mask[1][1]</p>	
SetMaskResponse	<p>This message is empty.</p>	
Fault codes	Description	
env:Sender ter:InvalidArgVal ter:NoConfig	<p>The requested Mask does not exist</p>	
env:Sender ter:InvalidArgVal ter:ConfigModify	<p>The configuration parameters are not possible to set.</p>	
env:Sender ter:InvalidArgVal ter:InvalidPolygon	<p>The provided polygon is not supported.</p>	

5.10.6 GetMaskOptions

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

Table 27: GetMaskOptions command

GetMaskOptions		Access Class: READ_MEDIA
Message name	Description	
GetMaskOptionsRequest	<p>The ConfigurationToken element specifies the video source configuration of which the suitable Mask options are requested. The Video Source Configuration Token shall exist in the device</p> <p>tt:ReferenceToken ConfigurationToken [1][1]</p>	
GetMaskOptionsResponse	<p>This message contains the Mask options which are suitable for the video source configuration specified in the request</p> <p>tr2:MaskOptions Options[1][1]</p>	

Fault codes	Description
env:Sender ter:InvalidArgVal ter:NoConfig	<i>The requested video source configuration does not exist</i>

5.11 Capabilities

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

RTPMulticast: Indication of support of UDP multicasting as described in sections 5.4 and 5.6.

RTP_RTSP_TCP: Indicates the support for RTP/RTSP/TCP transport as defined in section 5.1.1.3 of the ONVIF Streaming Specification.

NonAggregateControl: Indicates support for non aggregate RTSP control as described in section 5.2.1.1 of the ONVIF Streaming Specification.

RTSPStreaming: Indicates the support for live media streaming via RTSP.

MaximumNumberOfProfiles: The sum of fixed and dynamic MediaProfiles supported by the device.

SnapshotUri Indicates the support for GetSnapshotUri.

Rotation Indicates the support for the Rotation feature.

VideoSourceMode: Indicates the support for changing video source mode.

OSD: Indicates support for OSD configuration.

RTSPOverWebSocket: Indicates the support for RTSP/RTP streaming over WebSocket and provides the WebSocket URI, as described in Streaming Specification Section 5.1.1.5.

Mask: Indicates support for mask configuration.

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

Table 28: GetServiceCapabilities command

GetServiceCapabilities		Access Class: PRE_AUTH
Message name	Description	
GetServiceCapabilitiesRequest	<i>This is an empty message.</i>	
GetServiceCapabilitiesResponse	<i>The capability response message contains the requested service capabilities using a hierarchical XML capability structure.</i> tr2:Capabilities2 Capabilities [1][1]	
Fault codes	Description	
	<i>No command specific faults!</i>	

5.12 Events

5.12.1 Profile Change

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

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

5.12.2 Configuration Change

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

```
Topic: tns1:Media/ConfigurationChanged

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

5.12.3 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>
```

5.13 Service specific data types

5.13.1 MediaProfile

```
<xs:complexType name="MediaProfile">
  <xs:element name="Name" type="tt:Name"/>
  <xs:element name="Configurations" type="tr2: ConfigurationSet"
    minOccurs="0"/>
  <xs:attribute name="token" type="tt:ReferenceToken" use="required"/>
  <xs:attribute name="fixed" type="xs:boolean"/>
</xs:complexType>
```

- **Name**
User readable name of the profile.
- **Configurations**
The configurations assigned to the profile.
- **token**
Unique identifier of the profile.
- **fixed**
A value of true signals that the profile cannot be deleted. Default is false.

5.13.2 ConfigurationSet

Contains the configured configuration entities

```
<xs:complexType name=" ConfigurationSet">
  <xs:element name="VideoSource" type="tt:VideoSourceConfiguration"
    minOccurs="0"/>
  <xs:element name="AudioSource" type="tt:AudioSourceConfiguration"
    minOccurs="0"/>
  <xs:element name="VideoEncoder" type="tt:VideoEncoder2Configuration"
    minOccurs="0"/>
  <xs:element name="AudioEncoder" type="tt:AudioEncoder2Configuration"
    minOccurs="0"/>
  <xs:element name="Analytics" type=" tt:VideoAnalyticsConfiguration"
    minOccurs="0"/>
  <xs:element name="PTZ" type="tt:PTZConfiguration" minOccurs="0"/>
  <xs:element name="Metadata" type="tt:MetadataConfiguration"
    minOccurs="0"/>
  <xs:element name="AudioOutput" type="tt:AudioOutputConfiguration"
    minOccurs="0"/>
  <xs:element name="AudioDecoder" type="tt:AudioDecoderConfiguration"
    minOccurs="0"/></xs:complexType>
</xs:complexType>
```

- **VideoSource**
Reference to the Video source.
- **AudioSource**
Reference to the Audio source.
- **VideoEncoder**
Reference to the Video encoder.
- **AudioEncoder**
Reference to the Audio encoder.
- **PTZ**
Reference to the ptz configuration.
- **Analytics**
Reference to the active analytics configuration.
- **Metadata**
Configuration of the Metadata streaming.
- **AudioOutput**
Reference to the Audio output.

- **AudioDecoder**
Reference to the Audio decoder.

5.13.3 ConfigurationEntity

Base type defining the common properties of a configuration.

```
<xs:complexType name="ConfigurationEntity">
  <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**
Informational: number of internal references currently using this configuration.

5.13.4 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.

5.13.5 VideoSourceConfigurationExtension

```
<xs:complexType name="VideoSourceConfigurationExtension">
  <xs:element name="Rotate" type="tt:Rotate" minOccurs="0"/>
  <xs:element name="Extension" type="tt:VideoSourceConfigurationExtension2"
    minOccurs="0"/>
</xs:complexType>
```

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

5.13.6 VideoSourceConfigurationExtension2

```
<xs:complexType name="VideoSourceConfigurationExtension2">
  <xs:element name="LensDescription" type="tt:LensDescription"
    minOccurs="0" maxOccurs="unbounded"/>
  <xs:element name="SceneOrientation" type="tt:SceneOrientation"
    minOccurs="0" maxOccurs="1" />
</xs:complexType>
```

- **LensDescription**
Optional element describing the geometric lens distortion. Multiple instances for future variable lens support.

- **SceneOrientation**
Optional element describing the scene orientation in the camera's field of view.

5.13.7 LensDescription

```
<xs:complexType name="LensDescription">
  <xs:element name="Offset" type="tt:LensOffset" minOccurs="0"/>
  <xs:element name="Projection" type="tt:LensProjection"
    minOccurs="0" maxOccurs="unbounded"/>
  <xs:element name="XFactor" type="xs:float"/>
  <xs:attribute name="FocalLength" type="xs:float"/>
</xs:complexType>
```

- **Offset**
Offset of the lens center to the imager center in normalized coordinates.
- **Projection**
Radial description of the projection characteristics. The resulting curve is defined by the B-Spline interpolation over the given elements. The element for Radius zero shall not be provided. The projection points shall be ordered with ascending Radius. Items outside the last projection Radius shall be assumed to be invisible (black).
- **XFactor**
Compensation of the x coordinate needed for the ONVIF normalized coordinate system.
- **FocalLength**
Optional focal length of the optical system.

5.13.8 LensOffset

```
<xs:complexType name="LensOffset">
  <xs:attribute name="x" type="xs:float"/>
  <xs:attribute name="y" type="xs:float"/>
</xs:complexType>
```

- **x**
Optional horizontal offset of the lens center in normalized coordinates.
- **y**
Optional vertical offset of the lens center in normalized coordinates.

5.13.9 LensProjection

```
<xs:complexType name="LensProjection">
  <xs:element name="Angle" type="xs:float"/>
  <xs:element name="Radius" type="xs:float"/>
  <xs:element name="Transmittance" type="xs:float" minOccurs="0"/>
</xs:complexType>
```

- **Angle**
Angle of incidence.
- **Radius**
Mapping radius as a consequence of the emergent angle.
- **Transmittance**
Optional ray absorption at this point due to vignetting. A value of one means no absorption.

5.13.10 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.

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

5.13.11 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.

5.13.12 VideoSourceConfigurationOptionsExtension

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

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

5.13.13 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.

5.13.14 SceneOrientation

```
<xs:complexType name="SceneOrientation">
  <xs:element name="Mode" type="tt:SceneOrientationMode" />
</xs:complexType>
```

```
<xs:element name="Orientation" type="xs:string" minOccurs="0" />
</xs:complexType>
```

- **Mode**
The mode used to assign the Orientation element.
- **Orientation**
The orientation of the scene in the camera's field of view. The list should contain an enumerated value of tt:SceneOrientationOption.

5.13.15 VideoSourceConfigurationOptionsExtension2

```
<xs:complexType name="VideoSourceConfigurationOptionsExtension2">
  <xs:element name="SceneOrientationMode" type="tt:SceneOrientationMode"
    minOccurs="0" maxOccurs="unbounded" />
</xs:complexType>
```

- **SceneOrientationMode**
Mode options supported for the Scene Orientation feature. The list may contain one or more enumeration values of tt:SceneOrientationMode.

5.13.16 VideoEncoder2Configuration

```
<xs:complexType name="VideoEncoder2Configuration">
  <xs:element name="Encoding" type="xs:string" />
  <xs:element name="Resolution" type="tt:VideoResolution2" />
  <xs:element name="RateControl" type="tt:VideoRateControl2"
    minOccurs="0" />
  <xs:element name="Multicast" type="tt:MulticastConfiguration"
    minOccurs="0" />
  <xs:element name="Quality" type="xs:float" />
  <xs:attribute name="GovLength" type="xs:int" />
  <xs:attribute name="Profile" type="xs:string" />
</xs:complexType>
```

- **Encoding**
Mime name of the encoding as defined in [IANA Media Type Reference]
- **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.
- **Multicast**
Defines the multicast settings that could be used for video streaming.
- **GovLength**
Optional element to configure the group of I-Frames
- **Profile**
The encoder profile as defined in tt:VideoEncodingProfiles.

5.13.17 VideoResolution2

```
<xs:complexType name="VideoResolution2" />
<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.

5.13.18 VideoRateControl2

```
<xs:complexType name="VideoRateControl2"/>
  <xs:element name="FrameRateLimit" type="xs:float"/>
  <xs:element name="BitrateLimit" type="xs:int"/>
  <xs:attribute name="ConstantBitRate" type="xs:boolean"/>
</xs:complexType>
```

- **FrameRateLimit**
Desired frame rate in fps. The actual rate may be lower due to e.g. performance limitations.
- **BitrateLimit**
the maximum output bitrate in kbps
- **ConstantBitRate**
If set the Video Bitrate shall be constant. Default is variable.

5.13.19 VideoEncoder2ConfigurationOptions

```
<xs:complexType name="VideoEncoder2ConfigurationOptions">
  <xs:element name="Encoding" type="xs:string"/>
  <xs:element name="QualityRange" type="tt:IntRange"/>
  <xs:element name="ResolutionsAvailable" type="tt:VideoResolution2"/>
  <xs:element name="BitrateRange" type="tt:IntRange"/>
  <xs:attribute name="GovLengthRange" type="tt:IntAttrList"/>
  <xs:attribute name="FrameRatesSupported" type="tt:FloatAttrList"/>
  <xs:attribute name="ProfilesSupported" type="tt:StringAttrList"/>
  <xs:attribute name="ConstantBitRateSupported" type="xs:boolean"/>
</xs:complexType>
```

- **Encoding**
Mime name of the encoding as defined in [IANA Media Type Reference]
- **QualityRange**
Range of the quality values. A high value means higher quality.
- **ResolutionsAvailable**
List of supported image sizes.
- **GovLengthRange**
Attribute list which shall contain exactly two values, defining the Lower and Upper bounds, respectively, for the supported group of Video frames length. These values typically correspond to the I-Frame distance.
- **FrameRatesSupported**
List of supported target frame rates in fps (frames per second). The list shall be sorted with highest values first.
- **ProfilesSupported**
Encoding specific profiles supported.
- **ConstantBitRateSupported**
Signal whether enforcing constant bitrate is supported.

5.13.20 EncoderInstanceInfo

```
<xs:complexType name="EncoderInstanceInfo">
  <xs:element name="Codec" type="tr2:EncoderInstance" minOccurs="0"
    maxOccurs="unbounded"/>
  <xs:element name="Total" type="xs:int"/>
</xs:complexType>
```

- **Codec**
The minimum guaranteed number of encoder instances per video encoding format.
- **Total**
The total minimum guaranteed number of encoder instances (applications) for the VideoSourceConfiguration.

5.13.21 EncoderInstance

```
<xs:complexType name="EncoderInstance">
  <xs:element name="Encoding" type="xs:string"/>
  <xs:element name="Number" type="xs:int"/>
</xs:complexType>
```

- **Encoding**
Mime name of the video encoding format.
- **Number**
The minimum guaranteed number of encoder instances (applications) for the VideoSourceConfiguration.

5.13.22 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

5.13.23 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.

5.13.24 AudioEncoder2Configuration

```
<xs:complexType name="AudioEncoder2Configuration">
  <xs:extension base="tt:ConfigurationEntity"/>
  <xs:element name="Encoding" type="xs:string"/>
  <xs:element name="Multicast" type="tt:MulticastConfiguration"
    minOccurs="0"/>
  <xs:element name="Bitrate" type="xs:int"/>
  <xs:element name="SampleRate" type="xs:int"/>
</xs:complexType>
```

- **Encoding**
Mime name of the encoding as defined in [IANA Media Type Reference]
- **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.

5.13.25 AudioEncoder2ConfigurationOptions

```
<xs:complexType name="AudioEncoder2ConfigurationOptions">
  <xs:element name="Encoding" type="xs:string"/>
  <xs:element name="BitrateList" type="tt:IntList"/>
  <xs:element name="SampleRateList" type="tt:IntList"/>
</xs:complexType>
```

- **Encoding**
Mime name of the encoding as defined in [IANA Media Type Reference]

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

5.13.26 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**
Deprecated for Media2, and ignored.
- **AnalyticsEngineConfiguration**
Optional parameter to configure analytics engine.
- **CompressionType**
Optional parameter to configure compression type of Metadata payload

5.13.27 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

5.13.28 EventSubscription

Subscription 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**

5.13.29 MetadataConfigurationOptions

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

- **PTZStatusFilterOptions**

5.13.30 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.

5.13.31 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 information.
- **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.

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

5.13.33 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.
- **OutputLevelRange**
Minimum and maximum level range supported for this Output.

5.13.34 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>
```

5.13.35 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 active. 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.

5.13.36 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**

5.13.37 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

5.13.38 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="tr2: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.

5.13.39 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.

5.13.40 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**

5.13.41 OSDImgConfiguration

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

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

5.13.42 OSDTextOptions

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

```
<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

5.13.43 OSDImgOptions

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

- **ImagePath**
List available image path.

5.13.44 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.
Colours are represented by three-dimensional vectors. Additionally, the colourspace of each colour vector can be specified by a colourspace attribute. If the colourspace attribute is missing, the YCbCr colourspace is assumed. It refers to the 'sRGB' gamut with the RGB to YCbCr transformation as of ISO/IEC 10918-1 (Information technology -- Digital compression and coding of continuous-tone still images: Requirements and guidelines), a.k.a. JPEG. The following table lists the acceptable values for Colourspace attribute

Table 29 Colourspace namespace values

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

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

5.13.45 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.

5.13.46 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 specificate video source configuration. If a device limits the number of instances by OSDType, it should indicate the supported number 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.

5.13.47 Mask

```
<xs:complexType name="Mask">
  <xs:element name="ConfigurationToken" type="tt:ReferenceToken"/>
  <xs:element name="Polygon" type="tt:Polygon"/>
  <xs:element name="Type" type="xs:string"/>
  <xs:element name="Color" type="tt:Color" minOccurs="0"/>
  <xs:element name="Enabled" type="xs:boolean"/>
  <xs:attribute name="token" type="tt:ReferenceToken"/>
  <xs:attribute name="VideoSourceConfiguration" type="tt:ReferenceToken"/>
</xs:complexType>
```

- **ConfigurationToken**
Unique identifier of this mask configuration.
- **Polygon**
Geometric representation of the mask area.
- **Type**
Type of masking as defined by **tr2:MaskType**:
 - Color - The masked area is colored with color defined by the Color field.
 - Pixelated - The masked area is filled in mosaic style to hide details.
 - Blurred - The masked area is low pass filtered to hide details.
- **Color**
Color of the masked area.
- **Enabled**
If set the mask will cover the image, otherwise it will be fully transparent.
- **token**
Token of the mask.
- **VideoSourceConfiguration**
Token of the VideoSourceConfiguration the Mask is associated with.

5.13.48 MaskOptions

```
<xs:complexType name="MaskOptions">
  <xs:element name="MaxMasks" type="xs:int"/>
  <xs:element name="MaxPoints" type="xs:int"/>
  <xs:element name="Types" type="xs:string" minOccurs="0" maxOccurs="unbounded"/>
  <xs:element name="Color" type="tt:ColorOptions"/>
  <xs:attribute name="RectangleOnly" type="xs:boolean"/>
  <xs:attribute name="SingleColorOnly" type="xs:boolean"/>
</xs:complexType>
```

- **MaxMasks**
Maximum supported number of masks per VideoSourceConfiguration.
- **MaxPoints**
Maximum supported number of points per mask.
- **Types**
Information which types of **tr2:MaskType** are supported. Valid values are 'Color', 'Pixelated' and 'Blurred'.
- **Color**
Colors supported.
- **RectangleOnly**
Information whether the polygon must have four points and a rectangular shape.
- **SingleColorOnly**
Information that all polygon of a VideoSource have the same color.

5.14 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 30: Media service specific fault codes

Fault Code	Parent Subcode	Fault Reason	Description
	Subcode		
env:Receiver	ter:ActionNotSupported	No audio capability	The device does not support audio.
	ter:AudioNotSupported		
env:Receiver	ter:Action	Maximum number reached	The maximum number of supported profiles has been reached.
	ter:MaxNVTPProfiles		
env:Receiver	ter:ActionNotSupported	No audio output capability	Audio or Audio Outputs are not supported by the device.
	ter:AudioOutputNotSupported		
env:Receiver	ter:ActionNotSupported	No audio decoding capability	Audio or Audio Decoding is not supported by the device.
	ter:AudioDecodingNotSupported		
env:Receiver	ter:Action	Configuration not complete	Entities required by this action are missing in the specified profile.
	ter:IncompleteConfiguration		
env:Receiver	ter:Action	Conflict when using new settings	The new settings conflicts with other uses of the configuration.
	ter:ConfigurationConflict		
env:Receiver	ter:Action	Reach the maximum number of OSD	The maximum number of the OSD supported by the specified VideoSourceConfiguration has been reached.
	ter:MaxOSDs		
env:Sender	ter:InvalidArgVal	Profile token already exists	A profile with the token ProfileToken already exists.
	ter:ProfileExists		
env:Sender	ter:InvalidArgVal	Configuration token does not exist	The requested configuration indicated by the ConfigurationToken does not exist.
	ter:NoConfig		
env:Sender	ter:InvalidArgVal	Profile token does not exist	The requested profile token ProfileToken does not exist.
	ter:NoProfile		

env:Sender	ter:Action	Fixed profile can not be deleted	The fixed Profile cannot be deleted.
	ter:DeletionOfFixedProfile		
env:Sender	ter:InvalidArgVal	Parameters can not be set	The configuration parameters are not possible to set.
	ter:ConfigModify		
env:Sender	ter:ActionNotSupported	No video analytics capability	The device does not support video analytics.
	ter:VideoAnalyticsNot-Supported		
env:Sender	ter:InvalidArgVal	Invalid Stream setup	Specification of StreamType or Transport part in StreamSetup is not supported.
	ter:InvalidStreamSetup		
env:Sender	ter:OperationProhibited	Stream conflict	Specification of StreamType or Transport part in StreamSetup causes conflict with other streams.
	ter:StreamConflict		
env:Sender	ter:InvalidArgVal	Invalid multicast configuration	Not all configurations are configured for multicast..
	ter:InvalidMulticastSettings		
env:Sender	ter:InvalidArgVal	Video source mode token does not exist.	The requested video source mode does not exist.
	ter:NoVideoSourceMode		

Annex A. Efficient XML Interchange (EXI) (Normative)

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 31 and Table 32) shall be supported by all devices signalling support for EXI compression. Schema based EXI encoding shall be used. The required schema can be obtained from any device using the GetWsdlUrl command.

The EXI header shall only be transmitted if a setting different 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 31) shall always be used.

Table 31 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 32
Padding Bits	If present must be "0".

Table 32 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)
schemalD	Insert reference to schema obtained from device here.
datatypeRepresentationMap	none
blockSize	default (1,000,000)
valueMaxLength	default (unbounded)
valuePartitionCapacity	default (unbounded)
user defined meta-data	none

Annex B. Lens Description (Normative)

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

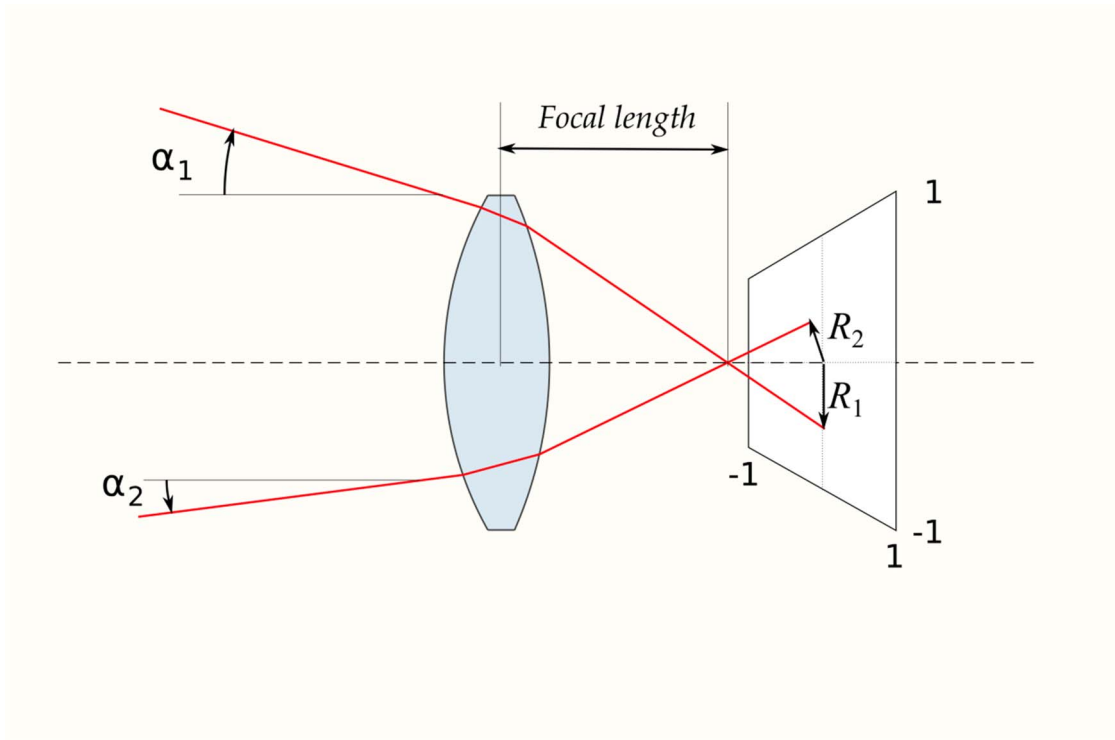


Figure 1: Optical mapping of angle (α) via Radius (R) to normalized x/y coordinates

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

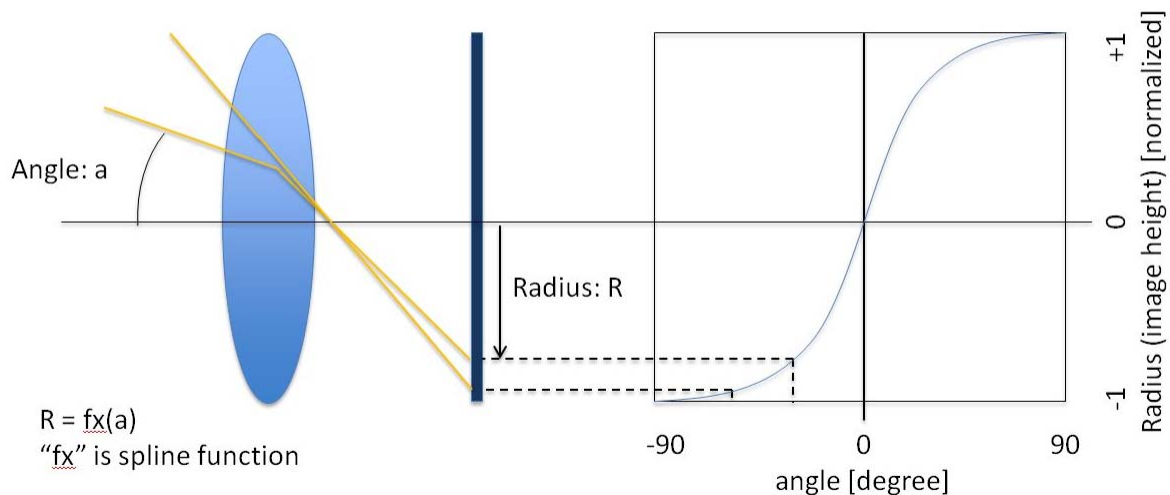
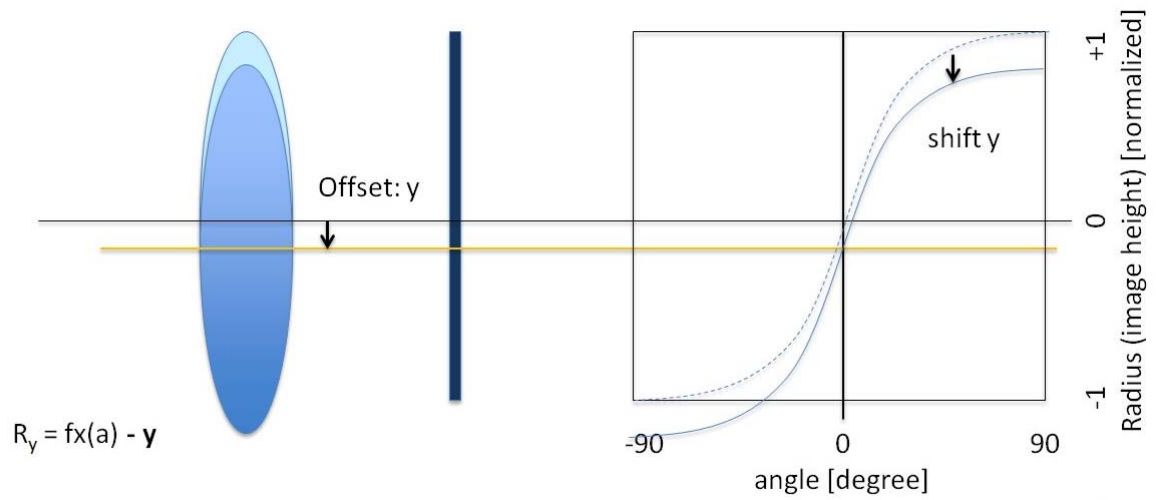


Figure 2: Smooth mapping using B-Splines

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



4

Figure 3: Compensation of vertical axis offset

Annex C. Revision History

Rev.	Date	Editor	Changes
1.0	Dec-2015	Hans Busch	First release
1.01	Mar-2016	Hans Busch	CR 1767, 1819
16.06	Jun-2016	Hiroyuki Sano	CR 1786, 1795, 1799, 1801, 1819, 1861, 1869
16.12	Dec-2016	Sujith Raman Steve Dillingham Hans Busch Hiroyuki Sano	Added RTSP over WebSocket Added Scene Orientation Modes Added Privacy Mask CR 1873, 1875, 1931, 1951, 1983, 1986