ONVIF™
Media2 Service Specification

Version 21.06

June 2021

CONTENTS

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.pdf>

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

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

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/wsdl/media.wsdlI.

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).

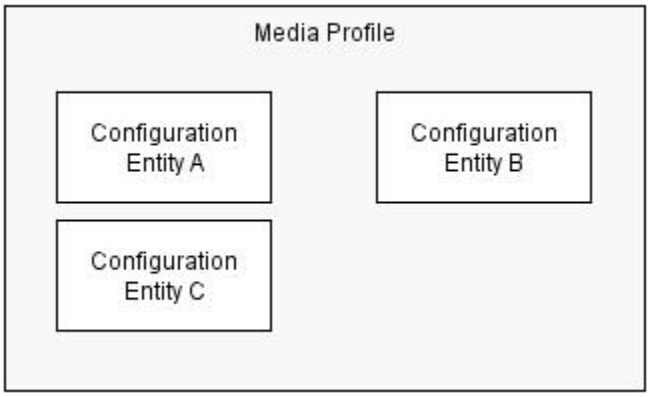

Figure 1: A media profile

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.

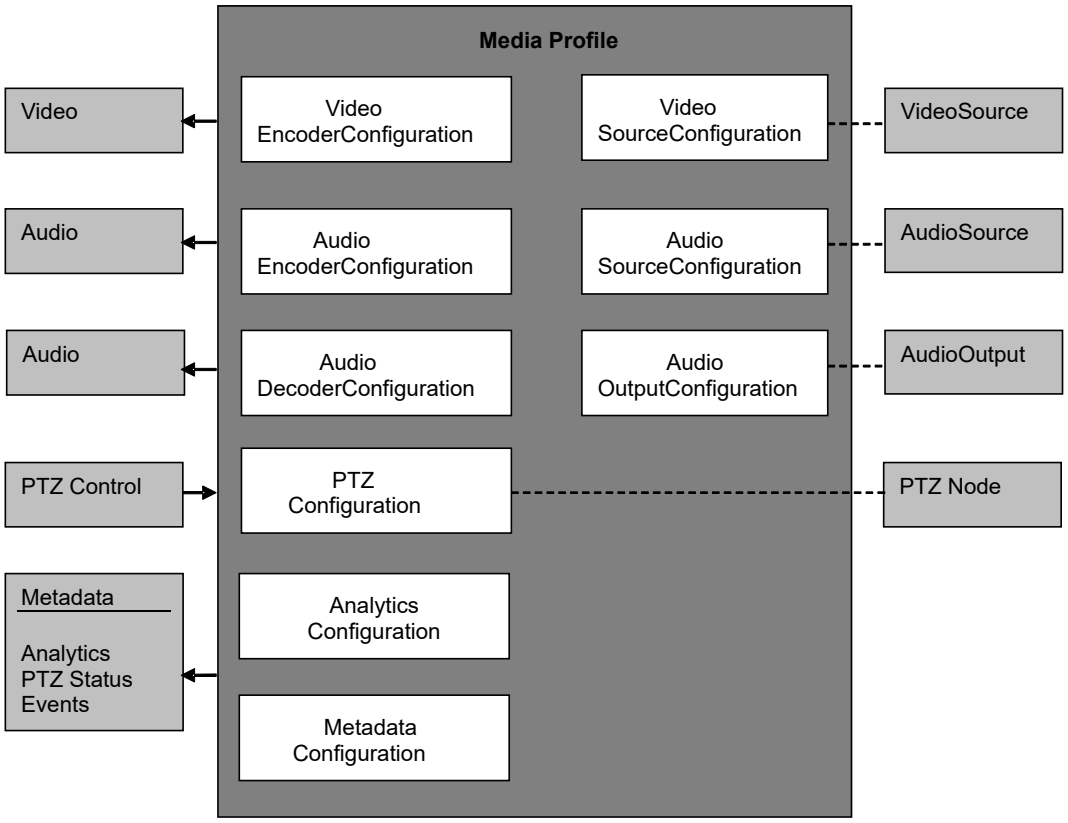

Figure 2: Figure 2: Complete profile configuration

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 CreateProfile

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

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

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

REQUEST:

- **Name [tt:Name]**
  Name of the new profile.
- **Configuration - optional, unbounded [tr2:ConfigurationRef]**
  Optional list of configurations to be added to the new profile. List entries with tr2:ConfigurationEnumeration value "All" shall be ignored.

RESPONSE:

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

FAULTS:

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

ACCESS CLASS:

ACTUATE

5.1.2 GetProfiles

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

The token parameter controls which profiles are returned:

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

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

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

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

REQUEST:

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

RESPONSE:

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

FAULTS:

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

ACCESS CLASS:

READ_MEDIA

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

5.1.3 AddConfiguration

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.

REQUEST:

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

RESPONSE:

This is an empty message.

FAULTS:

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

ACCESS CLASS:

ACTUATE

5.1.4 RemoveConfiguration

This operation removes one or more configurations from an existing media profile. Tokens appearing in the configuration list shall be ignored. Presence of the "All" type shall result in an empty 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.

REQUEST:

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

RESPONSE:

This is an empty message.

FAULTS:

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

ACCESS CLASS:

ACTUATE

5.1.5 DeleteProfile

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

A device signalling support for MultiTrackStreaming shall support deleting of virtual profiles via the command. Note that deleting a profile of a virtual profile set may invalidate the virtual profile.

REQUEST:

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

RESPONSE:

This is an empty message.

FAULTS:

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

ACCESS CLASS:

ACTUATE

5.1.6 CreateMultitrackConfiguration

This operation bundles a set of existing profiles to a so called virtual profile allowing multitrack streaming in a single RTSP session.

A device shall except a set of profiles where all VideoSourceConfiguration elements point to different video sources and the device is able to stream the set of video sources as a multitrack stream. If the method succeeds the device shall assign the returned virtual profile token to all profiles associated to the passed list of tokens.

Note that a device supporting multitrack streaming is typically shipped with one or more sets of virtual tokens. A client can use the GetProfiles command to retrieve all pre-configured profiles.

A device signalling support for MultiTrackStreaming shall support creating virtual profiles via this method. Virtual profiles can be deleted using DeleteProfile.

REQUEST:

- **ProfileToken -unbounded [tt:ReferenceToken]**
  List of existing profile Tokens.

RESPONSE:

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

FAULTS:

- **env:Sender - ter:InvalidArgVal - ter:NoProfile**
  One or more passed profile tokens do not exist.
- **env:Sender - ter:InvalidArgVal - ter: InvalidProfileToken**
  The requested profiles can not be put together to create a virtual profile.

ACCESS CLASS:

ACTUATE

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.

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

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.

Certain configurations in a media profile are dependent on other configurations. In such cases, a device should not allow adding the dependent configuration if the required configuration isn't present. Similarly, a device should not allow removing the required configuration unless the dependent configuration is removed first. Furthermore a device shall allow adding and removing both configurations in the same AddConfiguration respective RemoveConfiguration command. The dependencies are as follow:

- VideoEncoderConfiguration depends on VideoSourceConfiguration
- AudioEncoderConfiguration depends on AudioSourceConfiguration
- PTZConfiguration depends on VideoSourceConfiguration
- AudioDecoderConfiguration depends on AudioOutputConfiguration
- MetadataConfiguration depends on VideoSourceConfiguration
- AnalyticsConfiguration depends on VideoSourceConfiguration and/or AudioSourceConfiguration

If a dependency is violated in an AddConfiguration or RemoveConfiguration request, a ter:ConfigurationConflict fault should be returned in the response.

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 origin of the bounds is located in the upper left corner of the video source.

The Rotate option specifies an optional rotation of the area defined by the Bounds parameters. Note that in case of e.g. a 90 degree rotation the width parameter corresponds to the height of the Video and vice versa.

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

The View Mode option informs a client what type of view is represented by the video source. The view modes enumeration include

- Fisheye – Undewarped viewmode from a device supporting fisheye lens.
- 360Panrama – 360 degree panoramic view.
- 180Panorama – 180 degree panoramic view.
- Quad – View mode combining four streams in single Quad, eg., applicable for devices supporting four heads.
- Original – Unaltered view from the sensor.
- LeftHalf – Viewmode combining the left side sensors, applicable for devices supporting multiple sensors.
- RightHalf – Viewmode combining the right side sensors, applicable for devices supporting multiple sensors.
- Dewarp – Dewarped view mode for device supporting fisheye lens.

5.2.3 Video encoder configuration

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

- Encoding – The Video Media Subtype used to compress the video. See the [IANA Media Type 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 Audio Media Subtype used to compress the audio. See the [IANA Media Type 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. Additionaly 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.

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.

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

Event streaming can be enabled and controlled using topic filters. A device signalling MaxContentFilterSize shall support content filtering. For topic and content filter configuration refer to section “Event Handling” of the ONVIF Core Specification.

Streaming of scene description can be enabled. Optionally the AnalyticsEngineConfiguration allows to restrict streaming of scene description to the provided list of AnalyticsModules. Note that analytics modules only generate scene description if they are configured in the AnalyticsConfiguration of the profile as defined in section 5.2.7.

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

The structure also contains multicast parameters used to configure and control multicast of the metadata stream. 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<entity>Configurations

The Get<entity>Configurations operation allows to retrieve the actual settings of one ore more 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.

REQUEST:

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

RESPONSE:

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

FAULTS:

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

ACCESS CLASS:

READ_MEDIA

5.3.3 Set<entity>Configuration

The Set<entity>Configuration 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.

REQUEST:

- **Configuration [tt:<entity>Configuration]**
  The Configuration element contains the modified configuration. The configuration shall exist in the device.

RESPONSE:

This message is empty.

FAULTS:

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

ACCESS CLASS:

ACTUATE

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<entity>ConfigurationOptions

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

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

REQUEST:

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

RESPONSE:

- **ConfigurationOptions - unbounded [tt:<entity>Configuration]**
  Depending on the structure of tr2:Get<entity>ConfigurationOptionsResponse, the options will return one or more elements. Note: entities AudioOutput, AudioSource, Metadata, and VideoSource return just one element.

FAULTS:

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

ACCESS CLASS:

READ_MEDIA

5.3.5 GetVideoEncoderInstances

This command provides information on how many video encoders a device can instantiate concurrently for a VideoSourceConfiguration. A device signaling support for VideoEncoder via the ConfigurationsSupported capability shall support this command.

The Info response contains the following information:

Total Total number of encoder instances independent of the codec,

Codec Number of encoder instances for each supported codec .

A device shall guarantee to instantiate the indicated number of instances concurrently. If a device limits the number of instances of each particular video encoding type, the response shall contain information per video codec. For each video source, there shall be at least one video source configuration for which the GetVideoEncoderInstances shall return a Total greater than 0.

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

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

REQUEST:

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

RESPONSE:

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

FAULTS:

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

ACCESS CLASS:

READ_MEDIA

5.4 Stream URI

5.4.1 GetStreamUri

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

A device supporting MultiTrackStreaming shall support the retrieval of a multitrack RTSP session URI by passing a virtual profile token.

REQUEST:

- **Protocol [xs:string]**
  The Protocol defines how the encoded data is expected to be streamed to the client.
- **ProfileToken [tt:ReferenceToken]**
  The ProfileToken element indicates the media profile to use and will define the configuration of the content of the stream.

RESPONSE:

- **Uri [xs:anyUri]**
  The stable Uri to be used for requesting the media stream.

FAULTS:

- **env:Sender - ter:InvalidArgVal - ter:NoProfile**
  The requested profile token token does not exist.
- **env:Sender - ter:InvalidArgVal - ter:InvalidStreamSetup**
  The specified Protocol is not supported.
- **env:Sender - ter:OperationProhibited - ter:StreamConflict**
  The specified Protocol causes a conflict with another stream.
- **env:Sender - ter:InvalidArgVal - ter:InvalidMulticastSettings**
  No configuration is configured for multicast.
- **env:Sender - ter:InvalidArgVal - ter: InvalidProfileToken**
  Virtual profile token corresponding source could not be spliced together.
- **env:Receiver - ter:Action - ter:IncompleteConfiguration**
  The specified media profile does not have the minimum amount of configurations to have streams. Please add at least one source configuration and one matching encoder configuration.

ACCESS CLASS:

READ_MEDIA

5.5 Snapshot

5.5.1 GetSnapshotUri

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.

REQUEST:

- **ProfileToken [tt:ReferenceToken]**
  The ProfileToken element indicates the media profile to use and will define the configuration of the content of the stream.

RESPONSE:

- **Uri [xs:anyUri]**
  Stable Uri to be used for acquiring a snapshot in JPEG format as well as parameters defining the lifetime of the Uri.

FAULTS:

- **env:Sender - ter:InvalidArgVal - ter:NoProfile**
  The requested profile token token does not exist.
- **env:Receiver - ter:Action - ter:IncompleteConfiguration**
  The specified media profile does contain either unused sources or encoder configurations without a corresponding source.

ACCESS CLASS:

READ_MEDIA

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 were 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 non-RTSP controlled multicast streaming by the AutoStartMulticast capability shall support the methods defined in this chapter.

5.6.1 StartMulticastStreaming

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

REQUEST:

- **ProfileToken [tt:ReferenceToken]**
  The ProfileToken element indicates the media profile to use and will define the configuration of the content of the stream.

RESPONSE:

This message is empty.

FAULTS:

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

ACCESS CLASS:

ACTUATE

5.6.2 StopMulticastStreaming

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

REQUEST:

- **ProfileToken [tt:ReferenceToken]**
  The ProfileToken element indicates the media profile to use.

RESPONSE:

This message is empty.

FAULTS:

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

ACCESS CLASS:

ACTUATE

5.7 Synchronization Points

5.7.1 SetSynchronizationPoint

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

For example, if a video stream is configured with a large I-frame distance and a client loses a single packet, the client does not display video until the next I-frame is transmitted. In such cases, the client can request a Synchronization Point which 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.

REQUEST:

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

RESPONSE:

This message is empty.

FAULTS:

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

ACCESS CLASS:

ACTUATE

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.

REQUEST:

- **VideoSourceToken [tt:ReferenceToken]**
  The video source for which the source modes should be retrieved.

RESPONSE:

- **VideoSourceMode - unbounded [tr2:VideoSourceMode]**
  List of mode information with capabilities of the respective video source.

FAULTS:

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

ACCESS CLASS:

READ_SYSTEM

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.

REQUEST:

- **VideoSourceToken [tt:ReferenceToken]**
  The video source for which the source modes should be retrieved.
- **VideoSourceModeToken [tt:ReferenceToken]**
  The token of the video source mode to be set.

RESPONSE:

- **Reboot [xs:boolean]**
  information about rebooting after returning response. When Reboot is set “true”, a device will reboot automatically after setting mode.

FAULTS:

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

ACCESS CLASS:

WRITE_SYSTEM

5.9 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.

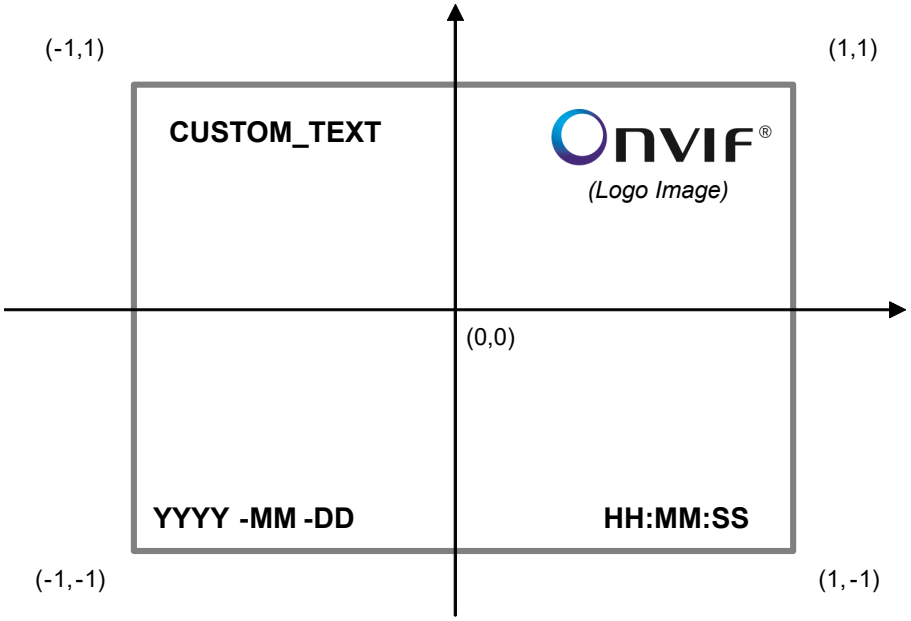

Figure 3: Example of screen which have four OSD configurations and coordinate system

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.

Device supporting temporary OSDTextConfiguration, shall notify TemporaryOSDText capability as defined in section 5.11. Device shall by default make all OSDTextConfigurations as persistent across reboot, but when IsPersistentText attribute in OSDTextConfiguration is set as false, OSD text content shall be cleared after reboot. OSDConfiguration shall still be valid after reboot.

5.9.2 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 Maximum-NumberOfOSDs in GetOSDOptions. A created OSD configuration shall be deletable.

REQUEST:

- **OSD [tt:OSDConfiguration]**
  Contains a new OSD configuration with the specified value. The device is responsible for assigning OSD token. OSD token in CreateOSDRequest can be ignored.

RESPONSE:

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

FAULTS:

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

- **env:Sender - ter:InvalidArgVal - ter:NoConfig**
  One of the provided configurations indicated by their ConfigurationToken does not exist.
- **env:Receiver - ter:InvalidArgVal - ter:ConfigModify**
  The configuration parameters are not possible to set.

ACCESS CLASS:

ACTUATE

Note: an OSDTextConfiguration without the IsPersistentText attribute shall be interpreted by the device as persistent.

5.9.3 DeleteOSD

This operation deletes an OSD configuration.

REQUEST:

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

RESPONSE:

This is an empty message.

FAULTS:

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

ACCESS CLASS:

ACTUATE

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 neither an OSD token nor a video source configuration token is provided the device shall respond with all available OSD configurations.

REQUEST:

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

RESPONSE:

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

FAULTS:

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

ACCESS CLASS:

READ_MEDIA

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.

REQUEST:

- **OSD [tt:OSDConfiguration]**
  The OSD element contains the modified OSD configuration. The Configuration contains an element that specifies the OSD whose configuration is to be modified. The OSD shall exist in the device.

RESPONSE:

This is an empty message.

FAULTS:

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

ACCESS CLASS:

ACTUATE

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.

REQUEST:

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

RESPONSE:

- **OSDOptions [tt:OSDConfigurationOptions]**
  the OSD options which are suitable for the video source configuration specified in the request.

FAULTS:

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

ACCESS CLASS:

READ_MEDIA

5.9.7 OSD Images

The ImageOption section of the GetOSDOptions response contains ImagePath, a list of URIs defining images on the device. If the FormatsSupported attribute of ImageOption is set, any URI in the ImagePath list may be

used to upload (POST) an image. Otherwise, the list defines references to images that may have been installed on the device outside of the ONVIF scope.

A device that returns the FormatsSupported attribute shall support at least the MIME type image/png. The device shall also return a MaxSize attribute of at least 1024 bytes, or MaxWidth and MaxHeight attributes of at least 16 pixels, or all three attributes. If all three attributes are provided, uploaded images that do not satisfy all three requirements shall be rejected.

The number of image paths in ImagePath defines how many images may be stored on the device. When posting an image to one of the URIs, the following HTML status codes may be returned, depending on success or failure:

200 OK or 201 Created	Image file was successfully uploaded
401 Unauthorized	Attempted POST without authentication credentials at the ACTUATE access policy level.
404 Not Found	Attempted POST to a URI not included in ImagePath list
408 Request Timeout	POST took too long to upload.
411 Length Required	POST does not include Content-Length header
413 Request Entity Too Large	File is larger than MaxSize attribute allows, image is larger than MaxWidth or MaxHeight attributes allow, or insufficient device storage space
415 Unsupported Media Type	POST does not include Content-Type header, or file MIME type does not match any

type in the FormatsSupported attribute's list

If an image is successfully uploaded but has a resolution larger than supported by the device, the device shall crop or resize the image and will not return an error.

A device that returns the FormatsSupported attribute shall download any OSD image listed in the ImgPath list with the GET method on the URI. The response shall include the appropriate Content-Type HTML header so the client knows how to interpret the image. When retrieving an image from one of the URIs, the following HTML status codes may be returned, depending on success or failure:

200 OK	Image file was successfully returned in the response.
401 Unauthorized	attempted GET without authentication credentials at the READ_MEDIA access policy.
404 Not Found	attempted GET of a URI not included in ImagePath list, or no uploaded image exists at the requested URI location

The images returned by GetOSDOptions may be utilized by SetOSD. ImgPath in the SetOSD request may specify any image path listed by GetOSDOptions. If SetOSD specifies a valid image path which does not yet have an associated uploaded image, the device may either treat the image as a 0x0 pixel image, or may return the ter:NoConfig fault.

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 4 shows how the mask is defined by a polygon using normalized coordinates relative to the VideoSourceConfiguration window.

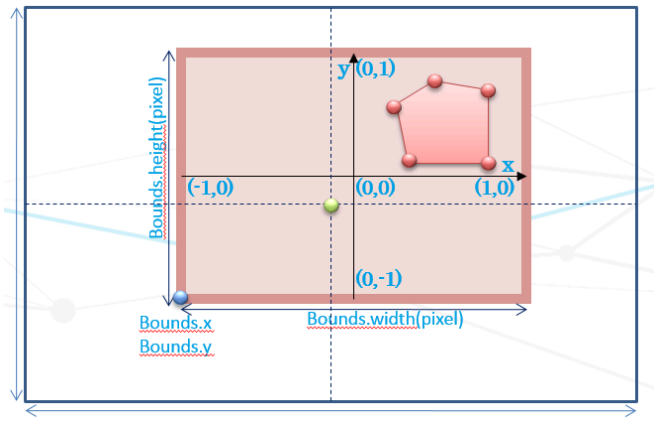

Figure 4: Example of screen with mask and coordinate system

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. Additionaly 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 CreateMask

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

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

REQUEST:

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

RESPONSE:

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

FAULTS:

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

ACCESS CLASS:

ACTUATE

5.10.3 DeleteMask

This operation deletes a mask configuration.

REQUEST:

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

RESPONSE:

This is an empty message.

FAULTS:

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

ACCESS CLASS:

ACTUATE

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. In case neither a mask nor configuration tokens is provided the device shall respond with all available Mask configurations in the device.

REQUEST:

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

RESPONSE:

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

FAULTS:

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

ACCESS CLASS:

READ_MEDIA

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.

REQUEST:

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

RESPONSE:

This is an empty message.

FAULTS:

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

ACCESS CLASS:

ACTUATE

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.

REQUEST:

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

RESPONSE:

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

FAULTS:

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

ACCESS CLASS:

READ_MEDIA

5.11 GetServiceCapabilities

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

REQUEST:

This is an empty message.

RESPONSE:

- **Capabilities [tr2:Capabilities2]**
  List of capabilties as defined above.

FAULTS:

None

ACCESS CLASS:

PRE_AUTH

The following profile capabilites are available:

MaximumNumberOfProfiles	The sum of fixed and dynamic MediaProfiles supported by the device. A device implementing this service shall provide this capability.
ConfigurationsSupported	The configurations supported by the device as defined by tr2:ConfigurationEnumeration. The enumeration value "All" shall not be included in this list. A device implementing this service shall provide this capability.
SnapshotUri	Indicates the support for GetSnapshotUri.
Rotation	Indicates the support for the Rotation feature.
VideoSourceMode	Indicates the support for changing video source mode.
OSD	Indicates support for OSD configuration
TemporaryOSDText	Indiates the support for temporary osd text configuration.
Mask	Indicates support for mask configuration.
SourceMask	Indicates that privacy masks are only supported at the video source level and not the video source configuration level. If this is true any addition, deletion or change of a privacy mask done for one video source configuration will automatically be applied by the device to a corresponding privacy mask for all other video source configuration associated with the same video source.

The following streaming capabilites are available:

RTSPStreaming	Indicates the support for live media streaming via RTSP.
RTPMulticast	Indication of support of UDP multicasting as described in sections 5.4.
RTP_RTSP_TCP	Indicates the support for RTP/RTSP/TCP transport as defined in section 5.1.1.3 of the ONVIF Streaming Specificiation.
NonAggregateControl	Indicates support for non aggregate RTSP control as described in section 5.2.1.1 of the ONVIF Streaming Specification.
RTSPWebSocketUri	Indicates the support for RTSP/RTP streaming over WebSocket and provides the WebSocket URI, as described in Streaming Specification Section 5.1.1.5.
AutoStartMulticast	Indicates support for non-RTSP controlled multicast streaming.
MultiTrackStreaming	Indicates suppot for multiple video RTP streams in one RTSP session.

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 should 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 should 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>
```

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

5.12.4 Active Sessions

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

```
Topics: tns1:Monitoring/ActiveSessions/VideoEncoder
        tns1:Monitoring/ActiveSessions/AudioEncoder
```

```
        tns1:Monitoring/ActiveSessions/AudioDecoder
        tns1:Monitoring/ActiveSessions/Metadata

<tt:MessageDescription IsProperty="true">
  <tt:Source>
    <tt:SimpleItemDescription Name="Token" Type="tt:ReferenceToken"/>
  </tt:Source>
  <tt:Data>
    <tt:SimpleItemDescription Name="Sessions" Type="tt:StringAttrList"/>
  </tt:Data>
</tt:MessageDescription>
```

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

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

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

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

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

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

```
<wsnt:Topic Dialect="...">
  tns1:Monitoring/ActiveSessions/Metadata
</wsnt:Topic>
<wsnt:Message>
  <tt:Message UtcTime="..." PropertyOperation="Initialized">
    <tt:Source>
      <tt:SimpleItem Name="Token" Value="af16a847-cd62-4923-9ccd-3108a16afaee" />
    </tt:Source>
    <tt:Data>
      <tt:SimpleItem Name="Sessions" Value="10.220.232.202" />
    </tt:Data>
  </tt:Message>
</wsnt:Message>
```

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

```
<wsnt:Topic Dialect="...">
  tns1:Monitoring/ActiveSessions/AudioEncoder
</wsnt:Topic>
<wsnt:Message>
```

```
  <tt:Message UtcTime="..." PropertyOperation="Deleted">
    <tt:Source>
      <tt:SimpleItem Name="Token" Value="audio" />
    </tt:Source>
    <tt:Data>
      <tt:SimpleItem Name="Sessions" Value="" />
    </tt:Data>
  </tt:Message>
</wsnt:Message>
```

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 A.1 and Table A.2) shall be supported by all devices signalling support for EXI compression. 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 A.1) shall always be used.

Table A.1: ONVIF defined EXI header settings

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

Table A.2: ONVIF defined EXI configuration settings

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

Annex B.
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.

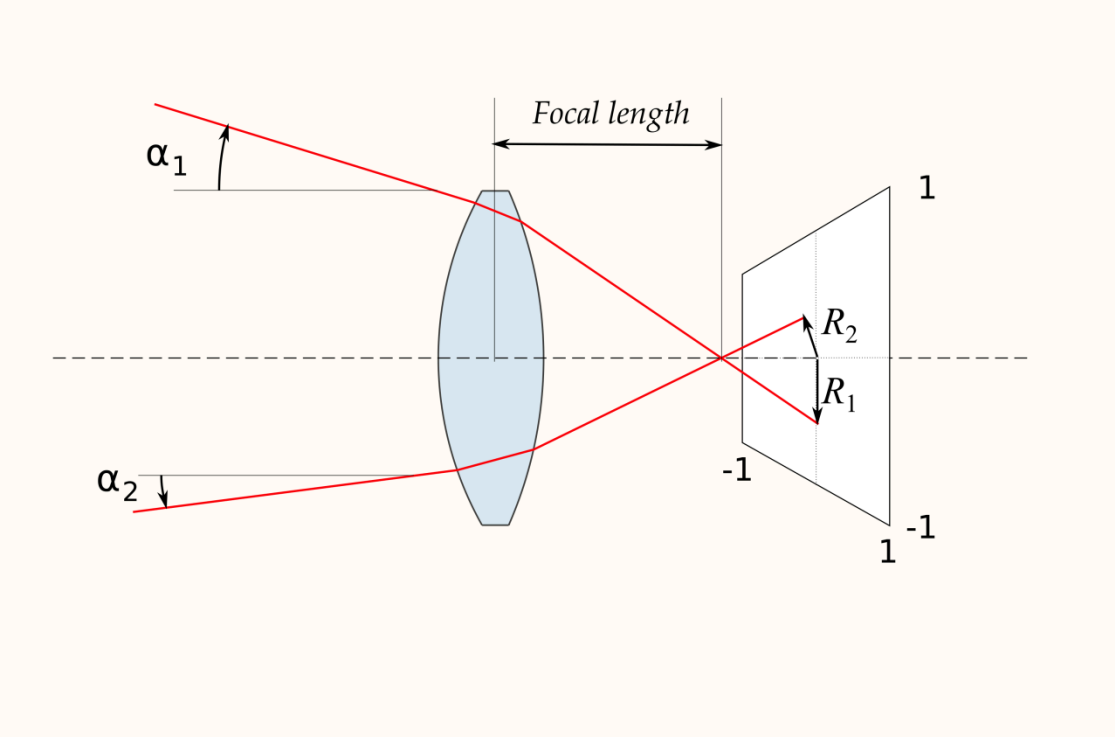

Figure B.1: Optical mapping of angle (α) via radius (R) to normalized x/y coordinates

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

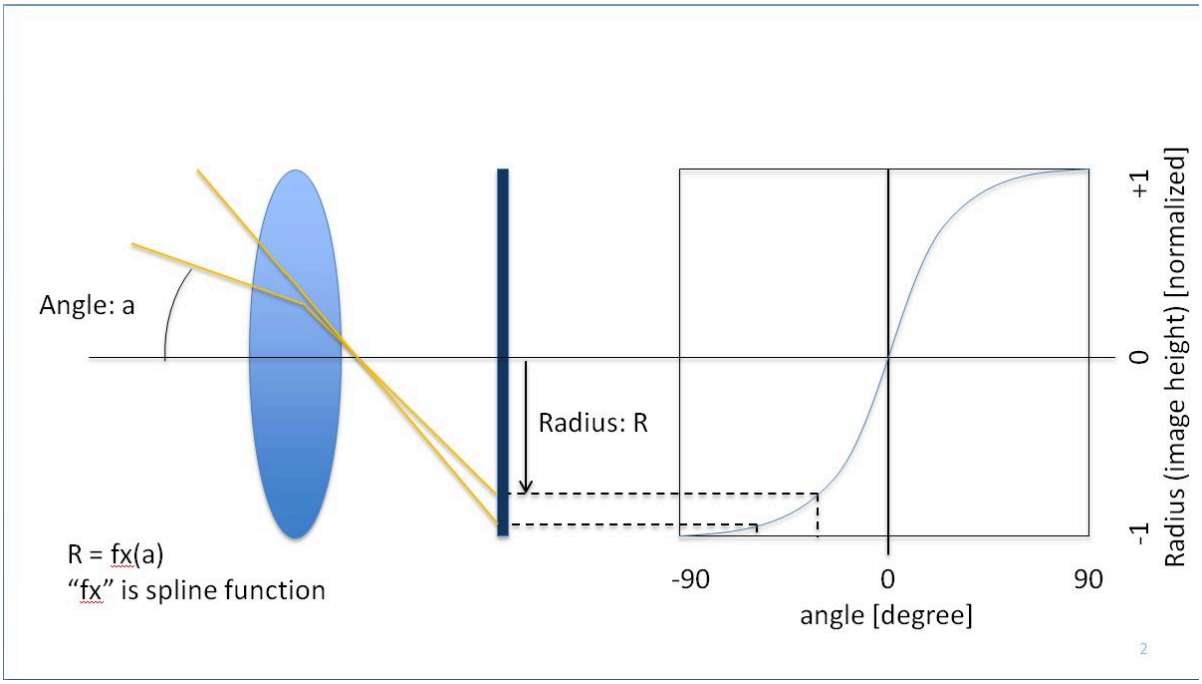

Figure B.2: Smooth mapping using B-Splines

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

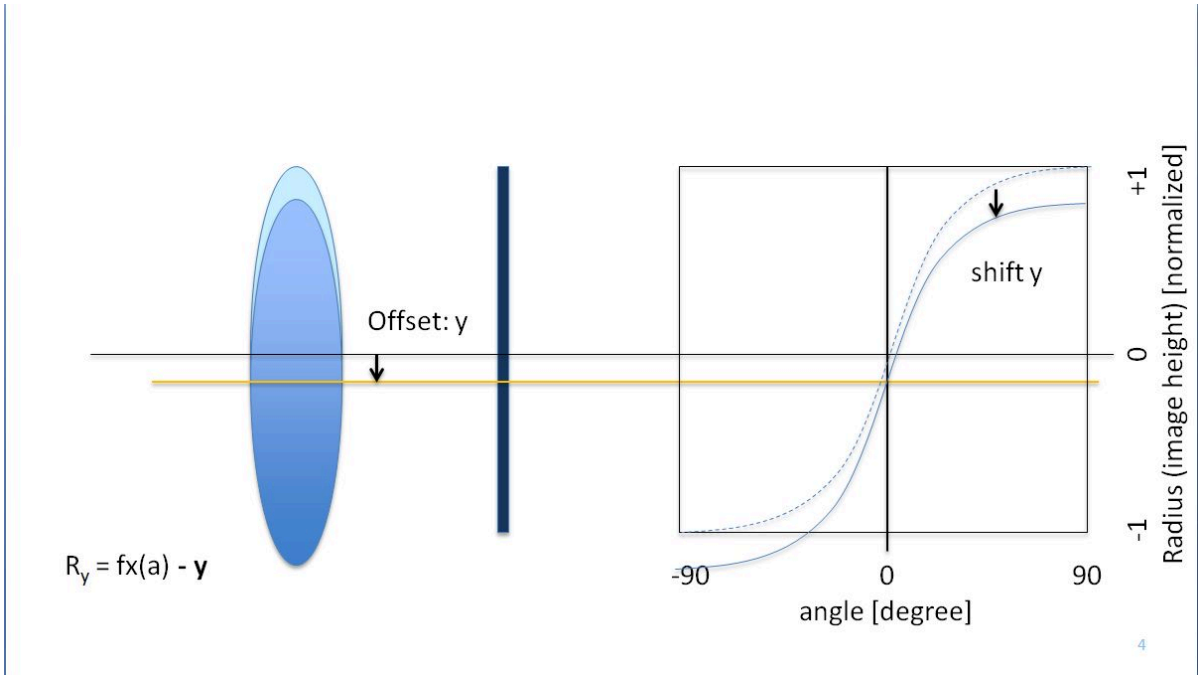

Figure B.3: Compensation of vertical axis offset

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
17.06	Jun-2017	Hans Busch, Hiroyuki Sano	Update method layouts. Change Request 1843, 2018, 2037, 2064, 2082, 2105
17.12	Dec-2017	Sujith Raman, Steve Wolf, Hiroyuki Sano	Added ViewMode Added OSD Image Upload Change Request 2152, 2159, 2176, 2187, 2197, 2203, 2207
18.06	Jun-2018	Hiroyuki Sano	Change Request 2216, 2217, 2282
18.12	Dec-2018	Hiroyuki Sano	Change Request 2295
19.06	Jun-2019	Steve Wolf, Hiroyuki Sano	Added Active Sessions Change Request 2479, 2512
19.12	Dec-2019	Hiroyuki Sano, Dora Han	Change Request 2489, 2557, 2573, 2603, 2636 Added Multitrack Streaming
20.06	Jun-2020	Michio Hirai	Change Request 2571, 2604
20.12	Dec-2020	Hans Busch	Support media service without video encoder.
21.06	Jun-2021	Sujith Raman	Clarify system coordinates.