ONVIF™
Media Service Specification

Version 2.1
June, 2011

CONTENTS

1 Scope

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

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

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

2 Normative references

ONVIF Core Specification

<http://www.onvif.org/onvif/specs/core/ONVIF-Core-Spec-v210.pdf>

ONVIF Media Service Specification

<http://www.onvif.org/onvif/specs/srv/media/ONVIF-Media-Service-Spec-v210.pdf>

ONVIF Imaging Service Specification

<http://www.onvif.org/onvif/specs/srv/img/ONVIF-Imaging-Service-Spec-v210.pdf>

ONVIF PTZ Service Specification

<http://www.onvif.org/onvif/specs/srv/ptz/ONVIF-PTZ-Service-Spec-v210.pdf>

ONVIF Streaming Specification

<http://www.onvif.org/onvif/specs/stream/ONVIF-Streaming-Spec-v210.pdf>

ONVIF Video Analytics Specification

<http://www.onvif.org/onvif/specs/srv/analytics/ONVIF-VideoAnalytics-Service-Spec-v210.pdf>

3 Terms and Definitions

3.1 Definitions

Configuration Entity	A network video device media abstract component that is used to produce a media stream on the network, i.e. video and/or audio stream.
Control Plane	Consists of Media control functions, such as device control, media configuration and PTZ commands.
Digital PTZ	Function that diminishes or crops an image to adjust the image position and ratio.
Media Entity	Media configuration entity such as video source, encoder, audio source, PTZ, and analytics, for example.
Media Plane	Consists of media stream, such as video, audio and metadata.
Media Profile	Maps a video or an audio source or an audio output to a video or an audio encoder, a audio decoder configuration and PTZ and analytics configurations.
Metadata	All streaming data except video and audio, including video analytics results, PTZ position data and other metadata (such as textual data from POS applications).
NVT	Network Video Transmitter
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

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 the media service is provided in the Media WSDL file.

4.1.1 Media profiles

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

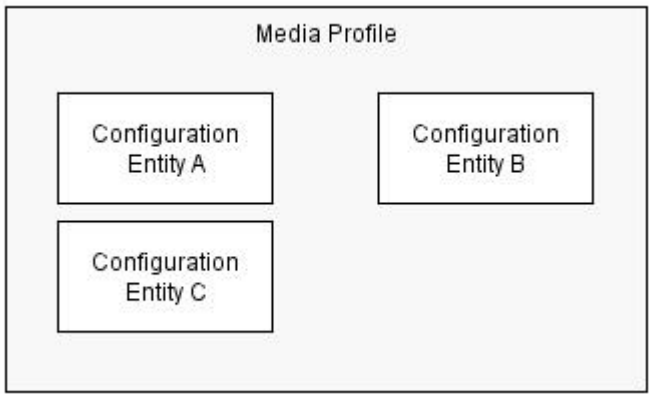

Figure 1: A media profile

A device having Media configuration service provides at least one media profile at boot. A device may provide “ready to use” profiles for the most common media configurations that the device offers.

The Profile contains a “fixed” attribute that indicates if a profile can be deleted or not. The fixed attribute does not signal that a profile is immutable. Hence it shall be possible to add or remove configurations to or from a fixed profile. Whether a profile is fixed or not is defined by the NVT.

A profile consists of a set of interconnected *configuration entities.* Configurations are provided by the NVT and can be either static or created dynamically by the NVT. For example, the dynamic configurations can be created by the NVT 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 NVT, a particular configuration entity can be part of a profile or not. For example, a profile with an audio source and an audio encoder configuration can exist only in a device with audio support.

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

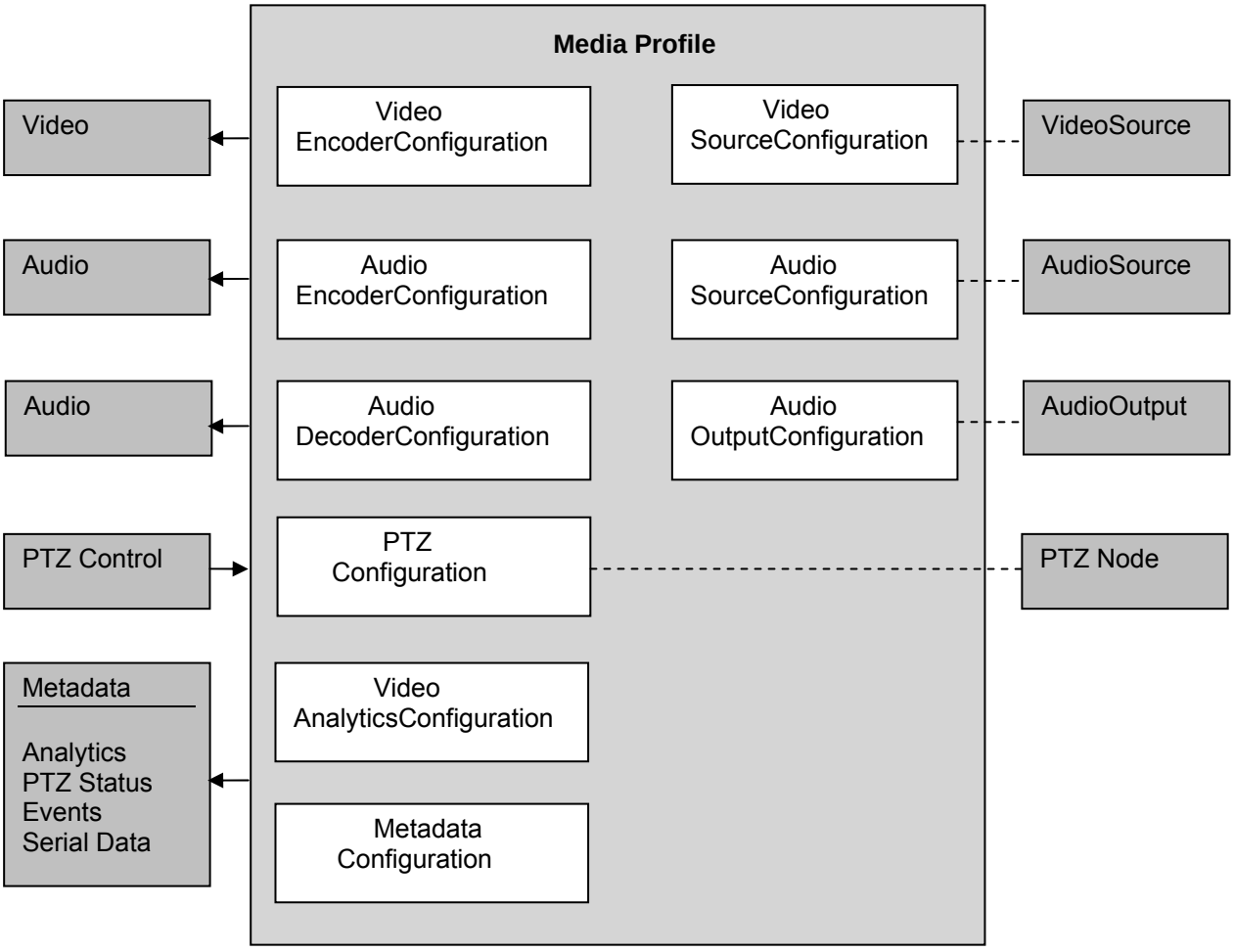

Figure 2: Complete profile configuration

A media profile describes how and what to present to the client in a media stream as well as how to handle PTZ input and Analytics.

The following commands list existing sources:

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

The following commands manage Media Profiles:

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

The following commands manage Configuration Entities:

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

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

GetVideoEncoderConfiguration

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

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

5 Service

The media service is used to configure the NVT media streaming properties. The NVT shall support the media service as specified in [ONVIF Media WSDL].

The media service allows a client to configure media and other real time streaming configurations. Media configurations are handled through media profiles. An overview of the ONVIF media configuration model is given in Section 1.

The media service commands are divided into two major categories:

- Media configuration:
    - Media profile commands
    - Video source commands
    - Vide encoder commands
    - Audio source commands
    - Audio encoder commands
    - Video analytics commands
    - Metadata commands
    - Audio output commands
    - Audio decoder commands

- Media streaming:
    - Request stream URI
    - Get snapshot URI
    - Multicast control commands
    - Media synchronization point

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

5.1 Audio and video codecs

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

- The NVT shall support JPEG QVGA.

- The NVT shall support G.711μ Law (Simplex-Camera Microphone Only, 1ch) [ITU-T G.711] if the NVT supports audio.

5.2 Media Profile

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

An NVT shall provide at least one media profile at boot. An NVT 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 NVT and can be either static or created dynamically by the NVT. For example, the dynamic configurations can be created by the NVT 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 NVT, 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.

5.2.1 Create media profile

This operation creates a new empty media profile. The media profile shall be created in the NVT and shall be persistent (remain after reboot). The NVT shall support the creation of media profiles as defined in this standard through the CreateProfile command.

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

Table 1: CreateProfile command

CreateProfile	Access Class: ACTUATE
Message name	**Description**
CreateProfileRequest	*Contains the friendly **Name** of the Profile to create as well as an optional **Token** parameter, specifying the unique identifier of the new media profile* tt:Name **Name** [1][1] tt:ReferenceToken **Token** [0][1]
CreateProfileResponse	*Returns an empty Profile structure with no configuration entities.* tt:Profile **Profile** [1][1]
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:ProfileExists	*A profile with the token **ProfileToken** already exists.*
env:Receiver ter:Action ter:MaxNVTProfiles	*The maximum number of supported profiles has been reached.*

5.2.2 Get media profiles

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

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

Table 2: GetProfiles command

GetProfiles	Access Class: READ_MEDIA
Message name	**Description**
GetProfilesRequest	*This is an empty message.*
GetProfilesResponse	*The response contains a list of profiles. Each profile contains a set of configuration entities defining a specific configuration that can be used for media streaming, analytics, metadata streaming etc.* tt:Profile **Profiles** [0][unbounded]
Fault codes	**Description**
	No command specific faults!

5.2.3 Get media profile

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

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

Table 3: GetProfile command

GetProfile	Access Class: READ_MEDIA
Message name	**Description**
GetProfileRequest	*This message contains the token to the requested profile.* tt:ReferenceToken **ProfileToken** [1][1]
GetProfileResponse	*The response contains the **Profile** indicated by the **Token** parameter. A Profile contains a set of configuration entities defining a specific configuration that can be used for media streaming, analytics, metadata streaming etc.* tt:Profile **Profile** [1][1]
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NoProfile	*The requested profile token **ProfileToken** does not exist.*

5.2.4 Add video source configuration to a profile

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

Table 4: AddVideoSourceConfiguration command

AddVideoSourceConfiguration	Access Class: ACTUATE
Message name	**Description**
AddVideoSourceConfiguration Request	*Contains a reference to the **VideoSourceConfiguration** to add and the **Profile** where it shall be added.* tt:ReferenceToken **ProfileToken** [1][1] tt:ReferenceToken **ConfigurationToken** [1][1]
AddVideoSourceConfiguration Response	*This is an empty message.*
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NoProfile	*The requested profile token **ProfileToken** does not exist.*
env:Sender ter:InvalidArgVal ter:NoConfig	*The VideoSourceConfiguration indicated by the **ConfigurationToken** does not exist.*
env:Receiver ter:Action ter:ConfigurationConflict	*Other configurations of the media profile conflicts with the one to add and adding it would cause a conflicting media profile.*

5.2.5 Add video encoder configuration to a profile

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

Adding a VideoEncoderConfiguration to a Profile means that a stream using that Profile will contain video data. Video encoder configurations should be added after adding a video source configuration.

Table 5: AddVideoEncoderConfiguration command

AddVideoEncoderConfiguration	Access Class: ACTUATE
Message name	**Description**
AddVideoEncoderConfiguration Request	*Contains a reference to the **VideoEncoderConfiguration** to add and the **Profile** where it shall be added.* tt:ReferenceToken **ProfileToken** [1][1] tt:ReferenceToken **ConfigurationToken** [1][1]
AddVideoEncoderConfiguration Response	*This is an empty message.*
Fault codes	**Description**
env:Sender ter:InvalidArgs ter:NoProfile	*The requested profile token **ProfileToken** does not exist.*
env:Sender ter:InvalidArgs ter:NoConfig	*The VideoEncoderConfiguration indicated by the **ConfigurationToken** does not exist.*
env:Receiver ter:Action ter:ConfigurationConflict	*Other configurations of the media profile conflicts with the one to add and adding it would cause a conflicting media profile.*

5.2.6 Add audio source configuration to a profile

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

Table 6: AddAudioSourceConfiguration command

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

env:Sender ter:InvalidArgVal ter:NoProfile	*The requested profile token **ProfileToken** does not exist.*
env:Sender ter:InvalidArgVal ter:NoConfig	*The AudioSourceConfiguration indicated by the **ConfigurationToken** does not exist.*
env:Receiver ter:Action ter:ConfigurationConflict	*Other configurations of the media profile conflicts with the one to add and adding it would cause a conflicting media profile.*
env:Receiver ter:ActionNotSupported ter:AudioNotSupported	*Audio is not supported.*

5.2.7 Add audio encoder configuration to a profile

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

Adding an AudioEncoderConfiguration to a media profile means that streams using that media profile will contain audio data. Audio encoder configurations should be added after adding an audio source configuration.

Table 7: AddAudioEncoderConfiguration command

AddAudioEncoderConfiguration	Access Class: ACTUATE
Message name	**Description**
AddAudioEncoderConfiguration Request	*Contains a reference to the **AudioEncoderConfiguration** to add and the **Profile** where it shall be added.* tt:ReferenceToken **ProfileToken** [1][1] tt:ReferenceToken **ConfigurationToken** [1][1]
AddAudioEncoderConfiguration Response	*This is an empty message.*
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NoProfile	*The requested profile token **ProfileToken** does not exist.*
env:Sender ter:InvalidArgVal ter:NoConfig	*The AudioEncoderConfiguration indicated by the **ConfigurationToken** does not exist.*
env:Receiver ter:Action ter:ConfigurationConflict	*Other configurations of the media profile conflicts with the one to add and adding it would cause a conflicting media profile.*
env:Receiver ter:ActionNotSupported ter:AudioNotSupported	*Audio is not supported.*

5.2.8 Add PTZ configuration to a profile

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

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

Table 8: AddPTZConfiguration command

AddPTZConfiguration	Access Class: ACTUATE
Message name	**Description**
AddPTZConfigurationRequest	*Contains a reference to the **PTZConfiguration** to add and the **Profile** where it shall be added.* tt:ReferenceToken **ProfileToken** [1][1] tt:ReferenceToken **ConfigurationToken** [1][1]
AddPTZConfigurationResponse	*This is an empty message.*
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NoProfile	*The requested profile token **ProfileToken** does not exist.*
env:Sender ter:InvalidArgVal ter:NoConfig	*The PTZConfiguration indicated by the **ConfigurationToken** does not exist.*
env:Receiver ter:Action ter:ConfigurationConflict	*Other configurations of the media profile conflicts with the one to add and adding it would cause a conflicting media profile.*
env:Receiver ter:ActionNotSupported ter:PTZNotSupported	*PTZ is not supported.*

5.2.9 Add video analytics configuration to a profile

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

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

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

Table 9: AddVideoAnalytics command

AddVideoAnalytics	Access Class: ACTUATE
Message name	**Description**
AddVideoAnalyticsRequest	*Contains a reference to the **VideoAnalytics** to add and the **Profile** where it shall be added.*

	tt:ReferenceToken **ProfileToken** [1][1] tt:ReferenceToken **ConfigurationToken** [1][1]
AddVideoAnalyticsResponse	*This is an empty message.*
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NoProfile	*The requested profile token **ProfileToken** does not exist.*
env:Sender ter:InvalidArgVal ter:NoConfig	*The VideoAnalytics indicated by the **ConfigurationToken** does not exist.*
env:Receiver ter:Action ter:ConfigurationConflict	*Other configurations of the media profile conflicts with the one to add and adding it would cause a conflicting media profile.*
env:Receiver ter:ActionNotSupported ter:VideoAnalyticsNotSupported	*VideoAnalytics is not supported.*

5.2.10 Add metadata configuration to a profile

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

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

Table 10: AddMetadataConfiguration command

AddMetadataConfiguration	Access Class: ACTUATE
Message name	**Description**
AddMetadataConfiguration Request	*Contains a reference to the **MetadataConfiguration** to add and the **Profile** where it shall be added.* tt:ReferenceToken **ProfileToken** [1][1] tt:ReferenceToken **ConfigurationToken** [1][1]
AddMetadataConfiguration Response	*This is an empty message.*
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NoProfile	*The requested profile token **ProfileToken** does not exist.*
env:Sender ter:InvalidArgVal ter:NoConfig	*The MetadataConfiguration indicated by the **ConfigurationToken** does not exist.*
env:Receiver ter:Action ter:ConfigurationConflict	*Other configurations of the media profile conflicts with the one to add and adding it would cause a conflicting media profile.*

5.2.11 Add audio output configuration

This operation adds an AudioOutputConfiguration to an existing media profile. If a configuration exists in the media profile, it will be replaced. The change shall be persistent. An NVT that has an Audio Output shall support addition of an audio output configuration to a profile through the AddAudioOutputConfiguration command.

Table 11: AddAudioOutputConfiguration

AddAudioOutputConfiguration	Access Class: ACTUATE
Message name	**Description**
AddAudioOutputConfiguration Request	*Contains a reference to the AudioOutputConfiguration to add and the Profile where it shall be added.* tt:ReferenceToken **ProfileToken** [1][1] tt:ReferenceToken **ConfigurationToken** [1][1]
AddAudioOutputConfiguration Response	*This is an empty message.*
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NoProfile	*The requested profile token ProfileToken does not exist.*
env:Sender ter:InvalidArgs ter:NoConfig	*The AudioOutputConfiguration indicated by the ConfigurationToken does not exist.*
env:Receiver ter:Action ter:ConfigurationConflict	*Other configurations of the media profile conflicts with the one to add and adding it would cause a conflicting media profile.*
env:Receiver ter:ActionNotSupported ter:AudioOutputNotSupported	*Audio or Audio Output is not supported*

5.2.12 Add audio decoder configuration

This operation adds an AudioDecoderConfiguration to an existing media profile. If a configuration exists in the media profile, it shall be replaced. The change shall be persistent. An NVT that has audio decoding capabilities shall support addition of an audio decoder configuration to a profile through the AddAudioDecoderConfiguration command.

Table 12: AddAudioDecoderConfiguration

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

env:Sender ter:InvalidArgVal ter:NoProfile	*The requested profile token ProfileToken does not exist.*
env:Sender ter:InvalidArgs ter:NoConfig	*The AudioDecoderConfiguration indicated by the ConfigurationToken does not exist.*
env:Receiver ter:Action ter:ConfigurationConflict	*Other configurations of the media profile conflicts with the one to add and adding it would cause a conflicting media profile.*
env:Receiver ter:ActionNotSupported ter:AudioDecodingNotSupported	*Audio or Audio Decoding is not supported*

5.2.13 Remove video source configuration from a profile

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

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

Table 13: RemoveVideoSourceConfiguration command

RemoveVideoSourceConfiguration	Access Class: ACTUATE
Message name	**Description**
RemoveVideoSourceConfiguration-Request	*Contains a reference to the media profile from which the VideoSourceConfiguration shall be removed.* tt:ReferenceToken **ProfileToken** [1][1]
RemoveVideoSourceConfiguration-Response	*This is an empty message.*
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NoProfile	*The requested profile token **ProfileToken** does not exist.*
env:Sender ter:InvalidArgVal ter:NoConfig	*There exists no video source configuration in the media profile.*
env:Receiver ter:Action ter:ConfigurationConflict	*Other configurations of the media profile are dependant on the VideoSourceConfiguration and removing it would cause a conflicting media profile.*

5.2.14 Remove video encoder configuration from a profile

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

Table 14: RemoveVideoEncoderConfiguration command

RemoveVideoEncoderConfiguration	Access Class: ACTUATE
Message name	**Description**
RemoveVideoEncoderConfiguration-Request	*Contains a reference to the media profile from which the VideoEncoderConfiguration shall be removed.* tt:ReferenceToken **ProfileToken** [1][1]
RemoveVideoEncoderConfiguration-Response	*This is an empty message.*
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NoProfile	*The requested profile token **ProfileToken** does not exist.*
env:Sender ter:InvalidArgVal ter:NoConfig	*There exists no video encoder configuration in the media profile.*
env:Receiver ter:Action ter:ConfigurationConflict	*Other configurations of the media profile are dependant on the VideoEncoderConfiguration and removing it would cause a conflicting media profile.*

5.2.15 Remove audio source configuration from a profile

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

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

Table 15: RemoveAudioSourceConfiguration command

RemoveAudioSourceConfiguration	Access Class: ACTUATE
Message name	**Description**
RemoveAudioSourceConfiguration-Request	*Contains a reference to the media profile from which the AudioSourceConfiguration shall be removed.* tt:ReferenceToken **ProfileToken** [1][1]
RemoveAudioSourceConfiguration-Response	*This is an empty message.*
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NoProfile	*The requested profile token **ProfileToken** does not exist.*
env:Sender ter:InvalidArgVal ter:NoConfig	*There exists no audio source configuration in the media profile.*
env:Receiver ter:Action ter:ConfigurationConflict	*Other configurations of the media profile are dependant on the AudioSourceConfiguration and removing it would cause a conflicting media profile.*

env:Receiver ter:ActionNotSupported ter:AudioNotSupported	*Audio is not supported.*

5.2.16 Remove audio encoder configuration from a profile

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

Table 16: RemoveAudioEncoderConfiguration command

RemoveAudioEncoderConfiguration	Access Class: ACTUATE
Message name	**Description**
RemoveAudioEncoderConfiguration-Request	*Contains a reference to the media profile from which the AudioEncoderConfiguration shall be removed.* tt:ReferenceToken **ProfileToken** [1][1]
RemoveAudioEncoderConfiguration-Response	*This is an empty message.*
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NoProfile	*The requested profile token **ProfileToken** does not exist.*
env:Sender ter:InvalidArgVal ter:NoConfig	*There exists no audio encoder configuration in the media profile.*
env:Receiver ter:Action ter:ConfigurationConflict	*Other configurations of the media profile are dependant on the AudioEncoderConfiguration and removing it would cause a conflicting media profile.*
env:Receiver ter:ActionNotSupported ter:AudioNotSupported	*Audio is not supported.*

5.2.17 Remove PTZ configuration from a profile

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

Table 17: RemovePTZConfiguration command

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

env:Sender ter:InvalidArgVal ter:NoProfile	*The requested profile token **ProfileToken** does not exist.*
env:Sender ter:InvalidArgVal ter:NoConfig	*There exists no PTZ configuration in the media profile.*
env:Receiver ter:Action ter:ConfigurationConflict	*Other configurations of the media profile are dependant on the PTZConfiguration and removing it would cause a conflicting media profile.*
env:Receiver ter:ActionNotSupported ter:PTZNotSupported	*PTZ is not supported.*

5.2.18 Remove video analytics configuration from a profile

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

Table 18: RemoveVideoAnalyticsConfiguration command

RemoveVideoAnalyticsConfiguration	Access Class: ACTUATE
Message name	**Description**
RemoveVideoAnalyticsConfiguration-Request	*Contains a reference to the media profile from which the VideoAnalyticsConfiguration shall be removed.* tt:ReferenceToken **ProfileToken** [1][1]
RemoveVideoAnalyticsConfiguration-Response	*This is an empty message.*
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NoProfile	*The requested profile token **ProfileToken** does not exist.*
env:Sender ter:InvalidArgVal ter:NoConfig	*There exists no video analytics configuration in the media profile.*
env:Receiver ter:Action ter:ConfigurationConflict	*Other configurations of the media profile are dependant on the VideoAnalyticsConfiguration and removing it would cause a conflicting media profile.*
env:Receiver ter:ActionNotSupported ter:VideoAnalyticsNotSupported	*VideoAnalytics is not supported.*

5.2.19 Remove metadata configuration from a profile

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

Table 19: RemoveMetadataConfiguration command

RemoveMetadataConfiguration		Access Class: ACTUATE
Message name	**Description**	

RemoveMetadataConfiguration-Request	*Contains a reference to the media profile from which the MetadataConfiguration shall be removed.* tt:ReferenceToken **ProfileToken** [1][1]
RemoveMetadataConfiguration-Response	*This is an empty message.*
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NoProfile	*The requested profile token **ProfileToken** does not exist.*
env:Sender ter:InvalidArgVal ter:NoConfig	*There exists no metadata configuration in the media profile.*
env:Receiver ter:Action ter:ConfigurationConflict	*Other configurations of the media profile are dependant on the MetadataConfiguration and removing it would cause a conflicting media profile.*

5.2.20 Remove audio output configuration

This operation removes an AudioOutputConfiguration from an existing media profile. If the media profile does not contain an AudioOutputConfiguration, the operation has no effect. The removal shall be persistent. An NVT that has at least one audio output shall support removal of an audio output configuration from a profile through the RemoveAudioOutputConfiguration command.

Table 20: RemoveAudioOutputConfiguration

RemoveAudioOutputConfiguration	Access Class: ACTUATE
Message name	**Description**
RemoveAudioOutputConfiguration-Request	*Contains a reference to the media profile from which the AudioOutputConfiguration shall be removed*. tt:ReferenceToken **ProfileToken** [1][1]
RemoveAudioOutputConfiguration-Response	*This is an empty message.*
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NoProfile	*The requested profile token ProfileToken does not exist.*
env:Sender ter:InvalidArgVal ter:NoConfig	*There exists no audio output configuration in the media profile.*
env:Receiver ter:Action ter:ConfigurationConflict	*Other configurations of the media profile are dependant on the AudioOutputConfiguration and removing it would cause a conflicting media profile.*
env: Receiver ter:ActionNotSupported ter:AudioOutputNotSupported	Audio or Audio output is not supported

5.2.21 Remove audio decoder configuration

This operation removes an AudioDecoderConfiguration from an existing media profile. If the media profile does not contain an AudioDecoderConfiguration, the operation has no effect. The removal shall be persistent. An NVT that supports audio decoding shall support removal

of an audio decoder configuration from a profile through the RemoveAudioDecoderConfiguration command.

Table 21: RemoveAudioDecoderConfiguration

RemoveAudioDecoderConfiguration	Access Class: ACTUATE
Message name	**Description**
RemoveAudioDecoderConfiguration-Request	*Contains a reference to the media profile from which the AudioDecoderConfiguration shall be removed*. tt:ReferenceToken **ProfileToken** [1][1]
RemoveAudioDecoderConfiguration-Response	*This is an empty message.*
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NoProfile	*The requested profile token ProfileToken does not exist.*
env:Sender ter:InvalidArgVal ter:NoConfig	*There exists no audio decoder configuration in the media profile.*
env:Receiver ter:Action ter:ConfigurationConflict	*Other configurations of the media profile are dependant on the AudioDecoder Configuration and removing it would cause a conflicting media profile.*
env: Receiver ter:ActionNotSupported ter::AudioDecodingNotSupported	*Audio or AudioDecoding is not supported*

5.2.22 Delete media profile

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

Table 22: DeleteProfile command

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

5.3 Video source

A VideoSource represents unencoded video input. The structure contains the pixel resolution of the video, framerate and imaging settings. The imaging settings can be manipulated through the ImagingService if supported and contains parameters for focus, exposure and brightness, for example. See the Imagaing Service Specification for more information.

5.3.1 GetVideoSources

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

Table 23: GetVideoSources command

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

5.4 Video source configuration

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

5.4.1 Get video source configurations

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

Table 24: GetVideoSourceConfigurations command

GetVideoSourceConfigurations	Access Class: READ_MEDIA
Message name	**Description**
GetVideoSourceConfigurations-Request	*This is an empty message.*
GetVideoSourceConfigurations-Response	*This message contains a list of* all *existing video source configurations in the NVT. A video source configuration does always*

	point at a real video source with the SourceToken element. tt:VideoSourceConfiguration **Configurations** [0][unbounded]
Fault codes	**Description**
	No command specific faults!

5.4.2 Get video source configuration

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

Table 25: GetVideoSourceConfiguration command

GetVideoSourceConfiguration	Access Class: READ_MEDIA
Message name	**Description**
GetVideoSourceConfiguration-Request	*This message contains the token of the requested video source configuration.* tt:ReferenceToken **ConfigurationToken** [1][1]
GetVideoSourceConfiguration-Response	*This message contains the requested VideoSourceConfiguration with the matching token. A video source configuration does always point at a real video source with the SourceToken element.* tt:VideoSourceConfiguration **Configuration** [1][1]
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NoConfig	*The requested configuration indicated with **ConfigurationToken** does not exist.*

5.4.3 Get compatible video source configurations

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

Table 26: GetCompatibleVideoSourceConfigurations command

GetCompatibleVideoSourceConfigurations	Access Class: READ_MEDIA
Message name	**Description**
GetCompatibleVideoSource-ConfigurationsRequest	*Contains the token of an existing media profile.* tt:ReferenceToken **ProfileToken** [1][1]
GetCompatibleVideoSource-ConfigurationsResponse	*Contains a list of video source configurations that are compatible with the media profile.*

	tt:VideoSourceConfiguration **Configurations** [0][unbounded]
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NoProfile	*The requested profile token **ProfileToken** does not exist.*

5.4.4 Get video source configuration options

This operation returns the available options when the video source parameters are reconfigured If a video source configuration is specified, the options shall concern that particular configuration. If a media profile is specified, the options shall be compatible with that media profile. The NVT shall support the listing of available video source parameter options (for a given profile and configuration) through the GetVideoSourceConfigurationOptions command.

Table 27: GetVideoSourceConfigurationOptions command

GetVideoSourceConfigurationOptions	Access Class: READ_MEDIA
Message name	**Description**
GetVideoSourceConfiguration-OptionsRequest	*This message contains optional tokens of a video source configuration and a media profile.* ***ConfigurationToken** specifies an existing configuration that the options are intended for.* ***ProfileToken** specifies an existing media profile that the options shall be compatible with.* tt:ReferenceToken **ConfigurationToken** [0][1] tt:ReferenceToken **ProfileToken** [0][1]
GetVideoSourceConfiguration-OptionsResponse	*This message contains the video configuration options. If a video source configuration is specified, the options shall concern that particular configuration. If a media profile is specified, the options shall be compatible with that media profile. If no tokens are specified, the options shall be considered generic for the device.* tt:VideoSourceConfigurationOptions **Options** [1][1]
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NoProfile	*The requested profile token **ProfileToken** does not exist.*
env:Sender ter:InvalidArgVal ter:NoConfig	*The requested configuration does not exist.*

5.4.5 Modify a video source configuration

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

shall support the modification of video source parameters through the SetVideoSourceConfiguration command.

Table 28: SetVideoSourceConfiguration command

SetVideoSourceConfiguration	Access Class: ACTUATE
Message name	**Description**
SetVideoSourceConfiguration-Request	*The **Configuration** element contains the modified video source configuration. The configuration shall exist in the NVT.* *The **ForcePersistence** element determines if the configuration changes shall be stored and remain after reboot. If true, changes shall be persistent. If false, changes MAY revert to previous values after reboot.* tt:VideoSourceConfiguration **Configuration** [1][1] xs:boolean **ForcePersistence** [1][1]
SetVideoSourceConfiguration-Response	*This message is empty.*
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NoConfig	*The configuration does not exist.*
env:Sender ter:InvalidArgVal ter:ConfigModify	*The configuration parameters are not possible to set.*
env:Receiver ter:Action ter:ConfigurationConflict	*The new settings conflicts with other uses of the configuration.*

5.5 Video encoder configuration

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

- Encoder – The encoding used for the video data.
- Resolution – The pixel resolution of the encoded video data.
- Quality – Determines the quality of the video. A high value within supported quality range means higher quality.
- RateControl – Defines parameters to configure the bitrate [kbps] as well as an EncodingInterval parameter (Interval at which images are encoded and transmitted) and a FrameRateLimit [fps] parameter to configure the output framerate.
- MPEG4/H264 specifics – Defines the encoding profile and GOV length [frame].

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

5.5.1 Get video encoder configurations

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

Table 29: GetVideoEncoderConfigurations command

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

5.5.2 Get video encoder configuration

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

Table 30: GetVideoEncoderConfiguration command

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

5.5.3 Get compatible video encoder configurations

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

support the listing of compatible (with a specific profile) video encoder configurations through the GetCompatibleVideoEncoderConfigurations command.

Table 31: GetCompatibleVideoEncoderConfigurations command

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

5.5.4 Get video encoder configuration options

This operation returns the available options when the video encoder parameters are reconfigured. The NVT shall support the listing of available video parameter options (for a given profile and configuration) through the GetVideoEncoderConfigurationOptions command.

Table 32: GetVideoEncoderConfigurationOptions command

GetVideoEncoderConfigurationOptions	Access Class: READ_MEDIA
Message name	**Description**
GetVideoEncoderConfiguration-OptionsRequest	*This message contains optional tokens of a video encoder configuration and a media profile.* ***ConfigurationToken** specifies an existing configuration that the options are intended for.* ***ProfileToken** specifies an existing media profile that the options shall be compatible with.* tt:ReferenceToken **ConfigurationToken** [0][1] tt:ReferenceToken **ProfileToken** [0][1]
GetVideoEncoderConfiguration-OptionsResponse	*This message contains the video configuration options. If a video encoder configuration is specified, the options shall concern that particular configuration. If a media profile is specified, the options shall be compatible with that media profile. If no tokens are specified, the options shall be considered generic for the device.* tt:VideoEncoderConfigurationOptions **Options** [1][1]
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NoProfile	*The requested profile token **ProfileToken** does not exist.*

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

5.5.5 Modify a video encoder configuration

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

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

Table 33: SetVideoEncoderConfiguration command

SetVideoEncoderConfiguration	Access Class: ACTUATE
Message name	**Description**
SetVideoEncoderConfiguration-Request	*The **Configuration** element contains the modified video encoder configuration. The configuration shall exist in the NVT.* *The **ForcePersistence** element determines if the configuration changes shall be stored and remain after reboot. If true, changes shall be persistent. If false, changes MAY revert to previous values after reboot.* tt:VideoEncoderConfiguration **Configuration** [1][1] xs:boolean **ForcePersistence** [1][1]
SetVideoEncoderConfiguration-Response	*This message is empty.*
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NoConfig	*The configuration does not exist.*
env:Sender ter:InvalidArgVal ter:ConfigModify	*The configuration parameters are not possible to set.*
env:Receiver ter:Action ter:ConfigurationConflict	*The new settings conflicts with other uses of the configuration.*

5.5.6 Get guaranteed number of video encoder instances

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

Table 34: GetGuaranteedNumberOfVideoEncoderInstances command

GetGuaranteedNumberOfVideoEncoderInstances	Access Class: READ_MEDIA
Message name	**Description**
GetGuaranteedNumberOf-EncoderInstancesRequest	*This request contains a token to the video source configuration.* tt: ReferenceToken **ConfigurationToken** [1][1]
GetGuaranteedNumberOf-EncoderInstancesResponse	*This message contains the minimum guaranteed **TotalNumber** of encoder instances (applications) per VideoSourceConfiguration. If a device limits the number of instances for respective Video Codecs the response contains the information how many **Jpeg**, **H264** and **Mpeg4** can be set up at the same time. In all other cases the device is able to deliver the **TotalNumber** of streams independend from the configured VideoCodec at the same time.* xs:int **TotalNumber** [1][1] xs:int **JPEG** [0][1] xs:int **H264** [0][1] xs:int **MPEG4** [0][1]
Fault codes	**Description**
env:Sender *ter:InvalidArgVal* *ter:NoConfig*	*The requested configuration indicated with **ConfigurationToken** does not exist.*

5.6 Audio source

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

5.6.1 Get audio sources

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

Table 35: GetAudioSources command

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

5.7 Audio source configuration

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

5.7.1 Get audio source configurations

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

Table 36: GetAudioSourceConfigurations command

GetAudioSourceConfigurations	Access Class: READ_MEDIA
Message name	**Description**
GetAudioSourceConfigurations-Request	*This is an empty message.*
GetAudioSourceConfigurations-Response	*This message contains a list of* all *existing audio source configurations in the NVT. An audio source configuration does always point at a real audio source with the SourceToken element.* tt:AudioSourceConfiguration **Configurations** [0][unbounded]
Fault codes	**Description**
env:Receiver ter:ActionNotSupported ter:AudioNotSupported	*NVT does not support audio.*

5.7.2 Get audio source configuration

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

Table 37: GetAudioSourceConfiguration command

GetAudioSourceConfiguration	Access Class: READ_MEDIA
Message name	**Description**
GetAudioSourceConfiguration-Request	*This message contains the token of the requested audio source configuration. An audio source configuration does always point at a real audio source with the SourceToken element.* tt:ReferenceToken **ConfigurationToken** [1][1]
GetAudioSourceConfiguration-Response	*This message contains the requested AudioSourceConfiguration with the matching token.* tt:AudioSourceConfiguration **Configuration** [1][1]
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NoConfig	*The requested configuration indicated with **ConfigurationToken** does not exist.*
env:Receiver ter:ActionNotSupported ter:AudioNotSupported	*NVT does not support audio.*

5.7.3 Get compatible audio source configurations

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

Table 38: GetCompatibleAudioSourceConfigurations command

GetCompatibleAudioSourceConfigurations	Access Class: READ_MEDIA
Message name	**Description**
GetCompatibleAudioSource-ConfigurationsRequest	*Contains the token of an existing media profile.* tt:ReferenceToken **ProfileToken** [1][1]
GetCompatibleAudioSource-ConfigurationsResponse	*Contains a list of audio source configurations that are compatible with the media profile.* tt:AudioSourceConfiguration **Configurations** [0][unbounded]
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NoProfile	*The requested profile token **ProfileToken** does not exist.*
env:Receiver ter:ActionNotSupported ter:AudioNotSupported	*NVT does not support audio.*

5.7.4 Get audio source configuration options

This operation returns the available options when the audio source parameters are reconfigured. If an audio source configuration is specified, the options shall concern that particular configuration. If a media profile is specified, the options shall be compatible with that media profile. An NVT that supports audio streaming from NVT to client shall support the listing of available audio parameter options (for a given profile and configuration) through the GetAudioSourceConfigurationOptions command.

Table 39: GetAudioSourceConfigurationOptions command

GetAudioSourceConfigurationOptions	Access Class: READ_MEDIA
Message name	**Description**
GetAudioSourceConfiguration-OptionsRequest	*This message contains optional tokens of an audio source configuration and a media profile.* ***ConfigurationToken*** *specifies an existing configuration that the options are intended for.* ***ProfileToken*** *specifies an existing media profile that the options shall be compatible with.* tt:ReferenceToken **ConfigurationToken** [0][1] tt:ReferenceToken **ProfileToken** [0][1]
GetAudioSourceConfiguration-OptionsResponse	*This message contains the audio configuration options. If an audio source configuration is specified, the options shall concern that particular configuration. If a media profile is specified, the options shall be compatible with that media profile. If no tokens are specified, the options shall be considered generic for the device.* tt:AudioSourceConfigurationOptions **Options** [1][1]
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NoProfile	*The requested profile token* ***ProfileToken*** *does not exist.*
env:Sender ter:InvalidArgVal ter:NoConfig	*The requested configuration does not exist.*
env:Receiver ter:ActionNotSupported ter:AudioNotSupported	*NVT does not support audio.*

5.7.5 Modify an audio source configuration

This operation modifies an audio source configuration. The ForcePersistence flag indicates if the changes shall remain after reboot of the NVT. Running streams using this configuration may be immediately updated according to the new settings, but the changes are not guaranteed to take effect unless the client requests a new stream URI and restarts any affected stream. If the new settings invalidate any parameters already negotiated using RTSP, for example by changing codec type, the NVT 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.

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

Table 40: SetAudioSourceConfiguration command

SetAudioSourceConfiguration	Access Class: ACTUATE
Message name	**Description**
SetAudioSourceConfiguration-Request	*The **Configuration** element contains the modified audio source configuration. The configuration shall exist in the NVT.* *The **ForcePersistence** element determines if the configuration changes shall be stored and remain after reboot. If true, changes shall be persistent. If false, changes MAY revert to previous values after reboot.* tt:AudioSourceConfiguration **Configuration** [1][1] xs:boolean **ForcePersistence** [1][1]
SetAudioSourceConfiguration-Response	*This message is empty.*
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NoConfig	*The configuration does not exist.*
env:Sender ter:InvalidArgVal ter:ConfigModify	*The configuration parameters are not possible to set.*
env:Receiver ter:Action ter:ConfigurationConflict	*The new settings conflicts with other uses of the configuration.*
env:Receiver ter:ActionNotSupported ter:AudioNotSupported	*NVT does not support audio.*

5.8 Audio encoder configuration

An AudioEncoderConfiguration contains the following parameters for encoding audio data:

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

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

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

5.8.1 Get audio encoder configurations

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

Table 41: GetAudioEncoderConfigurations command

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

5.8.2 Get audio encoder configuration

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

Table 42: GetAudioEncoderConfiguration command

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

5.8.3 Get compatible audio encoder configurations

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

Table 43: GetCompatibleAudioEncoderConfigurations command

GetCompatibleAudioEncoderConfigurations	Access Class: READ_MEDIA
Message name	**Description**
GetCompatibleAudioEncoder-ConfigurationsRequest	*Contains the token of an existing media profile.* tt:ReferenceToken **ProfileToken** [1][1]
GetCompatibleAudioEncoder-ConfigurationsResponse	*Contains a list of audio encoder configurations that are compatible with the given media profile.* tt:AudioEncoderConfiguration **Configurations** [0][unbounded]
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NoProfile	*The requested profile token **ProfileToken** does not exist.*
env:Receiver ter:ActionNotSupported ter:AudioNotSupported	*NVT does not support audio.*

5.8.4 Get audio encoder configuration options

This operation returns the available options when the audio encoder parameters are reconfigured. An NVT that supports audio streaming from NVT to client shall support the listing of available audio encoder parameter options (for a given profile and configuration) through the GetAudioEncoderConfigurationOptions command.

Table 44: GetAudioEncoderConfigurationOptions command

GetAudioEncoderConfigurationOptions	Access Class: READ_MEDIA
Message name	**Description**
GetAudioEncoderConfiguration-OptionsRequest	*This message contains optional tokens of an audio encoder configuration and a media profile.* ***ConfigurationToken** specifies an existing configuration that the options are intended for.* ***ProfileToken** specifies an existing media profile that the options shall be compatible with.* tt:ReferenceToken **ConfigurationToken** [0][1] tt:ReferenceToken **ProfileToken** [0][1]

GetAudioEncoderConfiguration-OptionsResponse	*This message contains the audio configuration options. If a audio encoder configuration is specified, the options shall concern that particular configuration. If a media profile is specified, the options shall be compatible with that media profile. If no tokens are specified, the options shall be considered generic for the device.* tt:AudioEncoderConfigurationOptions **Options** [1][1]
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NoProfile	*The requested profile token does not exist.*
env:Sender ter:InvalidArgVal ter:NoConfig	*The requested configuration does not exist.*
env:Receiver ter:ActionNotSupported ter:AudioNotSupported	*NVT does not support audio.*

5.8.5 Modify audio encoder configurations

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

Table 45: SetAudioEncoderConfiguration command

SetAudioEncoderConfiguration	Access Class: ACTUATE
Message name	**Description**
SetAudioEncoderConfiguration-Request	*The **Configuration** element contains the modified audio encoder configuration. The configuration shall exist in the NVT.* *The **ForcePersistence** element determines if the configuration changes shall be stored and remain after reboot. If true, changes shall be persistent. If false, changes MAY revert to previous values after reboot.* tt:AudioEncoderConfiguration **Configuration** [1][1] xs:boolean **ForcePersistence** [1][1]
SetAudioEncoderConfiguration-Response	*This message is empty.*
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NoConfig	*The configuration does not exist.*
env:Sender ter:InvalidArgVal ter:ConfigModify	*The configuration parameters are not possible to set.*
env:Receiver ter:Action ter:ConfigurationConflict	*The new settings conflicts with other uses of the configuration.*

env:Receiver ter:ActionNotSupported ter:AudioNotSupported	*NVT does not support audio.*

5.9 Video analytics configuration

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

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

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

5.9.1 Get video analytics configurations

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

Table 46: GetVideoAnalyticsConfigurations command

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

5.9.2 Get video analytics configuration

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

of a specific video analytics configuration through the GetVideoAnalyticsConfiguration command.

Table 47: GetVideoAnalyticsConfiguration command

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

5.9.3 Get compatible video analytics configurations

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

Table 48: GetCompatibleVideoAnalyticsConfigurations command

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

Supported	

5.9.4 Modify a video analytics configuration

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

Table 49: SetVideoAnalyticsConfiguration command

SetVideoAnalyticsConfiguration	Access Class: ACTUATE
Message name	**Description**
SetVideoAnalyticsConfiguration-Request	*The **Configuration** element contains the modified video analytics configuration. The configuration shall exist in the device.* *The **ForcePersistence** element determines if the configuration changes shall be stored and remain after reboot. If true, changes shall be persistent. If false, changes MAY revert to previous values after reboot.* tt:VideoAnalyticsConfiguration **Configuration** [1][1] xs:boolean **ForcePersistence** [1][1]
SetVideoAnalyticsConfiguration-Response	*This message is empty.*
Fault codes	**Description**
env:Sender ter:InvalidArgs ter:NoConfig	*The configuration does not exist.*
env:Sender ter:InvalidArgVal ter:ConfigModify	*The configuration parameters are not possible to set.*
env:Receiver ter:Action ter:ConfigurationConflict	*The new settings conflicts with other uses of the configuration.*
env:Sender ter:ActionNotSupported ter:VideoAnalyticsNot-Supported	*The device does not support video analytics.*

5.10 Metadata configuration

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

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

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

5.10.1 Get metadata configurations

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

Table 50: GetMetadataConfigurations command

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

5.10.2 Get metadata configuration

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

Table 51: GetMetadataConfiguration command

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

5.10.3 Get compatible metadata configurations

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

Table 52: GetCompatibleMetadataConfigurations command

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

5.10.4 Get metadata configuration options

This operation returns the available options for changing the metadata configuration. A NVT or another device that supports metadata streaming shall support the listing of available metadata parameter options (for a given profile and configuration) through the GetMetadataConfigurationOptions command.

Table 53: GetMetadataConfigurationOptions command

GetMetadataConfigurationOptions	Access Class: READ_MEDIA
Message name	**Description**
GetMetadataConfiguration-OptionsRequest	*This message contains optional tokens of a metadata configuration and a media profile.* ***ConfigurationToken** specifies an existing configuration that the options are intended for.* ***ProfileToken** specifies an existing media profile that the options shall be compatible with.* tt:ReferenceToken **ConfigurationToken** [0][1] tt:ReferenceToken **ProfileToken** [0][1]
GetMetadataConfiguration-OptionsResponse	*This message contains the metadata configuration options. If a metadata configuration is specified, the options shall concern that particular configuration. If a media profile is specified, the options shall be compatible with that media profile. If no tokens are specified, the options shall be considered generic for the device.*

	tt:MetadataConfigurationOptions **Options** [1][1]
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NoProfile	*The requested profile token does not exist.*
env:Sender ter:InvalidArgVal ter:NoConfig	*The requested configuration does not exist.*

5.10.5 Modify a metadata configuration

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

Table 54: SetMetadataConfiguration command

SetMetadataConfiguration	Access Class: ACTUATE
Message name	**Description**
SetMetadataConfiguration-Request	*The **Configuration** element contains multicast settings as well as a set of filters determining what data to include in the metadata stream.* *The **ForcePersistence** element determines if the configuration changes shall be stored and remain after reboot. If true, changes shall be persistent. If false, changes MAY revert to previous values after reboot.* tt:MetadataConfiguration **Configuration** [1][1] xs:boolean **ForcePersistence** [1][1]
SetMetadataConfiguration-Response	*This message is empty.*
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NoConfig	*The configuration does not exist.*
env:Sender ter:InvalidArgVal ter:ConfigModify	*The configuration parameters are not possible to set.*
env:Receiver ter:Action ter:ConfigurationConflict	*The new settings conflicts with other uses of the configuration.*

5.11 Audio outputs

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

5.11.1 Get audio outputs

This command lists all available audio outputs of a device. An NVT that has one or more physical audio outputs shall support listing of available audio outputs through the GetAudioOutputs command.

Table 55: GetAudioOutputs

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

5.12 Audio output configuration

The audio output configuration contains the following parameters:

- SourceToken: a reference to an existing audio output.
- OutputLevel: a parameter to configure the output volume
- SendPrimacy: a parameter that can be used for NVTs with a half duplex audio in/output to configure the active transmission direction (see Section 5.14).

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

5.12.1 Get audio output configurations

This command lists all existing AudioOutputConfigurations of a device. The NVC need not know anything apriori about the audio configurations to use this command. An NVT that is able to output audio shall support the listing of AudioOutputConfigurations through this command.

Table 56: GetAudioOutputConfigurations

GetAudioOutputConfigurations	Access Class: READ_MEDIA
Message name	**Description**
GetAudioOutputConfigurationsRequest	*This is an empty message.*
GetAudioOutputConfigurationsResponse	*Contains a list of AudioOutputConfigurations that are available on the device*

	tt:AudioOutputConfiguration **Configurations** [0][unbounded]
Fault codes	**Description**
env: Receiver ter:ActionNotSupported ter:AudioOutputNotSupported	*Audio or Audio Outputs are not supported by the device*

5.12.2 Get audio output configuration

If the audio output configuration token is already known, the output configuration can be fetched through the GetAudioOutputConfiguration command. An NVT that has one or more audio outputs shall support the retrieval of a specific audio output configuration through the GetAudioOutputConfiguration command.

Table 57: GetAudioOutputConfiguration

GetAudioOutputConfiguration	Access Class: READ_MEDIA
Message name	**Description**
GetAudioOutputConfigurationRequest	*This message contains the token of the requested AudioOutput configuration.* tt:ReferenceToken **ConfigurationToken** [1][1]
GetAudioOutputConfigurationResponse	*This message contains the requested AudioOutputConfiguration with the matching token.* tt:AudioOutputConfiguration **Configuration** [1][1]
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NoConfig	*The requested configuration indicated with **ConfigurationToken** does not exist.*
env: Receiver ter:ActionNotSupported ter::AudioOutputNotSupported	*Audio or Audio Outputs are not supported by the device*

5.12.3 Get compatible audio output configurations

This command lists all audio output configurations of a device that are compatible with a certain media profile. Each returned configuration shall be a valid input for the AddAudioOutputConfiguration command. An NVT that has one or more audio outputs shall support the listing of compatible (with a specific profile) AudioOutputConfigurations through the GetCompatibleAudioOutputConfigurations command.

Table 58: GetCompatibleAudioOutputConfiguration

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

	tt:AudioOutputConfiguration **Configurations** [0][unbounded]
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NoProfile	*The requested profile token **ProfileToken** does not exist.*
env:Receiver ter:ActionNotSupported ter:AudioOutputNotSupported	*Audio or Audio Outputs are not supported by the device*

5.12.4 Get audio output configuration options

This operation returns the available options for configuring an audio output. An NVT that has one or more audio outputs shall support the listing of available audio output configuration options (for a given profile and configuration) through the GetAudioOutputConfigurationOptions command.

Table 59: GetAudioOutputConfigurationOptions

GetAudioOutputConfigurationOptions	Access Class: READ_MEDIA
Message name	**Description**
GetAudioOutputConfiguration-OptionsRequest	*This message contains optional tokens of an audio output configuration and a media profile.* *ConfigurationToken specifies an existing configuration that the options are intended for.* *ProfileToken specifies an existing media profile that the options shall be compatible with.* tt:ReferenceToken **ConfigurationToken** [0][1] tt:ReferenceToken **ProfileToken** [0][1]
GetAudioOutputConfiguration-OptionsResponse	*This message contains the audio output configuration options. If a audio output configuration is specified, the options shall concern that particular configuration. If a media profile is specified, the options shall be compatible with that media profile. If no tokens are specified, the options shall be considered generic for the device.* tt:AudioOutputConfigurationOptions **Options** [1][1]
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NoProfile	*The requested profile token ProfileToken does not exist.*
env:Sender ter:InvalidArgVal ter:NoConfig	*The requested configuration does not exist.*
env:Receiver ter:ActionNotSupported ter:AudioOutputNotSupported	*Audio or Audio Outputs are not supported by the device*

5.12.5 Modify audio output configuration

This operation modifies an audio output configuration. The ForcePersistence flag indicates if the changes shall remain after reboot of the device. An NVT that has one or more audio outputs shall support the modification of audio output parameters through the SetAudioOutputConfiguration command.

Table 60: SetAudioOutputConfiguration

SetAudioOutputConfiguration	Access Class: ACTUATE
Message name	**Description**
SetAudioOutputConfiguration-Request	*The Configuration element contains the modified Audio Output configuration. The configuration must exist in the device.* *The ForcePersistence element determines if the configuration changes shall be stored and remain after reboot. If true, changes shall be persistent. If false, changes MAY revert to previous values after reboot.* tt:AudioOutputConfiguration **Configuration** [1][1] xs:boolean **ForcePersistence** [1][1]
SetAudioOutputConfiguration-Response	*This message is empty.*
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NoConfig	*The configuration does not exist.*
env:Sender ter:InvalidArgVal ter:ConfigModify	*The configuration parameters are not possible to set.*
env:Receiver ter:Action ter:ConfigurationConflict	*The new settings conflicts with other uses of the configuration.*
env: Receiver ter:ActionNotSupported ter:AudioOutputNotSupported	*Audio or Audio Outputs are not supported by the device*

5.13 Audio decoder configuration

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

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

5.13.1 Get audio decoder configurations

This command lists all existing AudioDecoderConfigurations of a device.

The NVC need not know anything apriori about the audio decoder configurations in order to use this command. An NVT that is able to decode audio shall support the listing of AudioOutputConfigurations through this command.

Table 61: GetAudioDecoderConfigurations

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

5.13.2 Get audio decoder configuration

If the audio decoder configuration token is already known, the decoder configuration can be fetched through the GetAudioDecoderConfiguration command. An NVT that is able to decode audio shall support the retrieval of a specific audio decoder configuration through the GetAudioDecoderConfiguration command.

Table 62: GetAudioDecoderConfiguration

GetAudioDecoderConfiguration	Access Class: READ_MEDIA
Message name	**Description**
GetAudioDecoderConfigurationRequest	*This message contains the token of the requested AudioDecoder configuration.* tt:ReferenceToken **ConfigurationToken** [1][1]
GetAudioDecoderConfigurationResponse	*This message contains the requested AudioDecoder Configuration with the matching token.* tt:AudioDecoderConfiguration **Configuration** [1][1]
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NoConfig	*The requested configuration indicated with **ConfigurationToken** does not exist.*
env:Receiver ter:ActionNotSupported ter:AudioDecodingNotSupported	*Audio or Audio decoding is not supported by the device*

5.13.3 Get compatible audio decoder configurations

This operation lists all the audio decoder configurations of the device that are compatible with a certain media profile. Each of the returned configurations shall be a valid input parameter for the AddAudioDecoderConfiguration command on the media profile. An NVT that is able to decode audio shall support the listing of compatible (with a specific profile) audio decoder configurations through the GetCompatibleAudioDecoderConfigurations command.

Table 63: GetCompatibleAudioDecoderConfigurations

GetCompatibleAudioDecoderConfigurations	Access Class: READ_MEDIA
Message name	**Description**
GetCompatibleAudioDecoderConfigurations Request	*Contains the token of an existing media profile.* tt:ReferenceToken **ProfileToken** [1][1]
GetCompatibleAudioDecoderConfigurations Response	*Contains a list of audiodecoder configurations that are compatible with the given media profile.* tt:AudioDecoderConfiguration **Configurations** [0][unbounded]
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NoProfile	*The requested profile token **ProfileToken** does not exist.*
env:Receiver ter:ActionNotSupported ter:AudioDecodingNotSupported	*Audio or Audio decoding is not supported by the device*

5.13.4 Get audio decoder configuration options

This command list the audio decoding capabilities for a given profile and configuration of a device. A device that is able to decode audio shall support the retrieval of AudioDecoderConfigurationOptions through this command.

Table 64: GetAudioDecoderConfigurationOptions

GetAudioDecoderConfigurationOptions	Access Class: READ_MEDIA
Message name	**Description**
GetAudioDecoderConfiguration-OptionsRequest	This message contains optional tokens of a audio decoder configuration and a media profile. ConfigurationToken specifies an existing configuration that the options are intended for. *ProfileToken specifies an existing media profile that the options shall be compatible with.* tt:ReferenceToken **ConfigurationToken** [0][1] tt:ReferenceToken **ProfileToken** [0][1]
GetAudioDecoderConfiguration-OptionsResponse	*This message contains the audio decoder configuration options. If a audio decoder configuration is specified, the options shall concern that particular configuration. If a media profile is specified, the options shall be compatible with that media profile. If no tokens are specified, the options shall be considered generic for the device.* tt:AudioDecoderConfigurationOptions **Options** [1][1]
Fault codes	**Description**
env:Sender ter:InvalidArgVal	*The requested profile token ProfileToken does not exist.*

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

5.13.5 Modify audio decoder configuration

This operation modifies an audio decoder configuration. The ForcePersistence flag indicates if the changes shall remain after reboot of the device. The device that is able to decode audio shall support the modification of audio decoder parameters through the SetAudioDecoderConfiguration command.

Table 65: SetAudioDecoderConfiguration

SetAudioDecoderConfiguration	Access Class: READ_MEDIA
Message name	Description
SetAudioDecoderConfiguration-Request	*The Configuration element contains the modified AudioDecoder configuration. The configuration must exist in the device.* *The ForcePersistence element determines if the configuration changes shall be stored and remain after reboot. If true, changes shall be persistent. If false, changes MAY revert to previous values after reboot.* tt:AudioDecoderConfiguration **Configuration** [1][1] xs:boolean **ForcePersistence** [1][1]
SetAudioDecoderConfiguration-Response	*This message is empty.*
Fault codes	Description
env:Sender ter:InvalidArgVal ter:NoConfig	*The configuration does not exist.*
env:Sender ter:InvalidArgVal ter:ConfigModify	*The configuration parameters are not possible to set.*
env:Receiver ter:Action ter:ConfigurationConflict	*The new settings conflicts with other uses of the configuration.*
env: Receiver ter:ActionNotSupported ter:AudioDecodingNotSupported	*Audio or Audio decoding is not supported by the device*

5.14 Audio channel modes

An audio channel MAY support different types of audio transmission. While for full duplex operation no special handling is required, in half duplex operation the transmission direction needs to be switched.

An optional Send-Primacy Parameter inside the AudioOutputConfiguration indicates which direction is currently active. An NVC 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 NVT in this mode.

- www.onvif.org/ver20/HalfDuplex/Client
  The client is allowed to send audio data via the backchannel to the server. The NVT shall not send audio data to the client in this mode.

- www.onvif.org/ver20/HalfDuplex/Auto
  It is up to the device how to deal with sending and receiving audio data.

Acoustic echo cancellation is out of ONVIF scope.

5.15 Stream URI

5.15.1 Request stream URI

This operation requests a URI that can be used to initiate a live media stream using RTSP as the control protocol. The returned URI shall remain valid indefinitely even if the profile is changed. The InvalidAfterConnect, InvalidAfterReboot and Timeout Parameter shall be set accordingly (InvalidAfterConnect=false, InvalidAfterReboot=false, timeout=PT0S). An NVT shall support the retrieval of a media stream URI for a specific media profile through the GetStreamUri command.

The correct syntax for the StreamSetup element for the media stream setups as defined in 5.1.1 of the ONVIF Streaming Specification are defined in Table 66.

Table 66: Valid setup parameter conbinations

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

If a multicast stream is requested the VideoEncoderConfiguration, AudioEncoderConfiguration and MetadataConfiguration element inside the corresponding media profile must be configured with valid multicast settings.

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

Table 67: GetStreamUri command

GetStreamUri	Access Class: READ_MEDIA
Message name	**Description**
GetStreamUriRequest	*The **StreamSetup** element contains two parts. StreamType defines if a unicast or multicast media stream is requested. Transport specifies a chain of transport protocols defining the tunnelling of the media stream over different network protocols.* *The **ProfileToken** element indicates the media profile to use and will define the configuration of the content of the stream.* tt:StreamSetup **StreamSetup** [1][1] tt:ReferenceToken **ProfileToken** [1][1]
GetStreamUriResponse	*Contains the stable **Uri** to be used for requesting the media stream as well as parameters defining the lifetime of the Uri. The **InvalidAfterConnect** and **InvalidAfterReboot** parameter shall be set to false, the **timeout** parameter shall be set to PT0S to indicate that this stream URI is indefinitely valid even if the profile changes.* xs:anyURI **Uri** [1][1] xs:boolean **InvalidAfterConnect** [1][1] xs:boolean **InvalidAfterReboot** [1][1] xs:duration **Timeout** [1][1]
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NoProfile	*The media profile does not exist.*
env:Sender ter:InvalidArgVal ter:InvalidStreamSetup	*Specification of StreamType or Transport part in **StreamSetup** is not supported.*
env:Sender ter:OperationProhibited ter:StreamConflict	*Specification of StreamType or Transport part in **StreamSetup** causes conflict with other streams.*
env:Receiver ter:Action ter:IncompleteConfiguration	*The specified media profile does contain either unused sources or encoder configurations without a corresponding source.*
env:Sender ter:InvalidArgVal ter:InvalidMulticastSettings	*Not all configurations are configured for multicast.*

5.16 Snapshot

5.16.1 Request snapshot URI

A Network client uses the GetSnapshotUri command to obtain a JPEG snhapshot from the NVT. The returned URI shall remain valid indefinitely even if the profile is changed. The ValidUntilConnect, ValidUntilReboot and Timeout Parameter shall be set accordingly (ValidUntilConnect=false, ValidUntilReboot=false, timeout=PT0S). The URI can be used for acquiring a JPEG image through a HTTP GET operation. The image encoding will always be JPEG regardless of the encoding setting in the media profile. A device supporting the media service should support this command.

Table 68: GetSnapshotUri command

GetSnapshotUri	Access Class: READ_MEDIA
Message name	**Description**
GetSnapshotUriRequest	*The **ProfileToken** element indicates the media profile to use and will define the source and dimensions of the snapshot.* tt:ReferenceToken **ProfileToken** [1][1]
GetSnapshotUriResponse	*Contains a stable **Uri** to be used for acquiring a snapshot in JPEG format as well as parameters defining the lifetime of the Uri. The **ValidUntilConnect** and **ValidUntilReboot** parameter shall be set to false, the **timeout** parameter shall be set to PT0S to indicate that this stream URI is indefinitely valid even if the profile changes.* xs:anyURI **Uri** [1][1] xs:boolean **InvalidAfterConnect** [1][1] xs:boolean **InvalidAfterReboot** [1][1] xs:duration **Timeout** [1][1]
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NoProfile	*The media profile does not exist.*
env:Receiver ter:Action ter:IncompleteConfiguration	*The specified media profile does not contain either a reference to a video encoder configuration or a reference to a video source configuration.*

5.17 Multicast

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

5.17.1 Start multicast streaming

This command starts multicast streaming using a specified media profile of an NVT. Streaming continues until StopMulticastStreaming is called for the same Profile. The streaming shall continue after a reboot of the NVT until a StopMulticastStreaming request is received. The multicast address, port and TTL are configured in the VideoEncoderConfiguration, AudioEncoderConfiguration and MetadataConfiguration respectively. An NVT that supports video, audio or metadata multicast streaming shall support the starting of a multicast stream through the StartMulticastStreaming command.

Table 69: StartMulticastStreaming command

StartMulticastStreaming	Access Class: ACTUATE
Message name	**Description**
StartMulticastStreaming-Request	*Contains the token of the Profile that is used to define the multicast stream.* tt:ReferenceToken **ProfileToken** [1][1]
StartMulticastStreaming-Response	*This message is empty.*
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NoProfile	*The profile does not exist.*
env:Receiver ter:Action ter:IncompleteConfiguration	*The specified media profile does not contain either a reference to a video encoder a video source configuration, to a audio source or to audio encoder configuration or a reference to a metadata configuration*

5.17.2 Stop multicast streaming

This command stop multicast streaming using a specified media profile of an NVT An NVT that supports video, audio or metadata multicast streaming shall support the stopping of a multicast stream through the StopMulticastStreaming command.

Table 70: StopMulticastStreaming command

StopMulticastStreaming	Access Class: ACTUATE
Message name	**Description**
StopMulticastStreaming-Request	*Contains the token of the Profile that is used to define the multicast stream.* tt:ReferenceToken **ProfileToken** [1][1]
StopMulticastStreaming-Response	*This message is empty.*
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NoProfile	*The profile does not exist.*
env:Receiver ter:Action ter:IncompleteConfiguration	*The specified media profile does not contain either a reference to a video encoder a video source configuration, to a audio source or to audio encoder configuration or a reference to a metadata configuration*

5.18 Synchronization Points

5.18.1 Set synchronization point

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

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

Frame as soon as possible. Clients can request Synchronization Points for profiles. The NVT shall add synchronization points for all streams associated with this profile.

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

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

An NVT that supports MPEG-4 or H.264 shall support the request for an I-Frame through the SetSynchronizationPoint command.

Table 71: SetSynchronizationPoint command

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

5.19 Capabilities

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

RTPMulticast: Indication of support of UDP multicasting as described in Section 5.1.1.1 of the ONVIF Streaming Specificiation.

RTP_TCP: Indication if the device supports RTP over TCP, see Section 5.1.1.2 of the ONVIF Streaming Specificiation.

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

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

MaximumNumberOfProfiles: The maximum Number of MediaProfiles the device supports.

Table 72: GetServiceCapabilities command

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

5.20 Events (Informative)

The display service dispatches events through the event service. In accordance with previous specifications this section is informative for the time being.

For the following entities of the Media Configuration, the ONVIF TopicNamespace provides the following topics:

```
tns1:MediaConfiguration/Profile
tns1:MediaConfiguration/VideoSourceConfiguration
tns1:MediaConfiguration/AudioSourceConfiguration
tns1:MediaConfiguration/VideoEncoderConfiguration
tns1:MediaConfiguration/AudioEncoderConfiguration
tns1:MediaConfiguration/VideoAnalyticsConfiguration
tns1:MediaConfiguration/PTZConfiguration
tns1:MediaConfiguration/MetaDataConfiguration
```

Each of these topics represents a property. A client subscribing to one of these topics will be notified about changes, creation and deletion of the corresponding entity.

The Message structures of the different topics are specified next using the MessageDescription Language introduced in the ONVIF Core Specification.

5.20.1 Profile

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

5.20.2 VideoSourceConfiguration

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

```
    <tt:ElementItemDescription Name="Config"
                 Type="tt:VideoSourceConfiguration"/>
  </tt:Data>
</tt:MessageDescription>
```

5.20.3 AudioSourceConfiguration

```
<tt:MessageDescription IsProperty="true">
  <tt:Source>
    <tt:SimpleItemDescription Name="AudioSourceConfigurationToken"
                Type="tt:ReferenceToken"/>
  </tt:Source>
  <tt:Data>
    <tt:ElementItemDescription Name="Config"
                 Type="tt:AudioSourceConfiguration"/>
  </tt:Data>
</tt:MessageDescription>
```

5.20.4 VideoEncoderConfiguration

```
<tt:MessageDescription iIsProperty="true">
  <tt:Source>
    <tt:SimpleItemDescription Name="VideoEncoderConfigurationToken"
                Type="tt:ReferenceToken"/>
  </tt:Source>
  <tt:Data>
    <tt:ElementItemDescription Name="Config"
                 Type="tt:VideoEncoderConfiguration"/>
  </tt:Data>
</tt:MessageDescription>
```

5.20.5 AudioEncoderConfiguration

```
<tt:MessageDescription iIsProperty="true">
  <tt:Source>
    <tt:SimpleItemDescription Name="AudioEncoderConfigurationToken"
                Type="tt:ReferenceToken"/>
  </tt:Source>
  <tt:Data>
    <tt:ElementItemDescription Name="Config"
                 Type="tt:AudioEncoderConfiguration"/>
  </tt:Data>
</tt:MessageDescription>
```

5.20.6 VideoAnalyticsConfiguration

```
<tt:MessageDescription IsProperty="true">
  <tt:Source>
    <tt:SimpleItemDescription Name="VideoAnalyticsConfigurationToken"
                Type="tt:ReferenceToken"/>
  </tt:Source>
  <tt:Data>
    <tt:ElementItemDescription Name="Config"
                 Type="tt:VideoAnalyticsConfiguration"/>
  </tt:Data>
</tt:MessageDescription>
```

5.20.7 PTZConfiguration

```
<tt:MessageDescription IsProperty="true">
  <tt:Source>
    <tt:SimpleItemDescription Name="PTZConfigurationToken"
                Type="tt:ReferenceToken"/>
  </tt:Source>
  <tt:Data>
    <tt:ElementItemDescription Name="Config"
                 Type="tt:PTZConfiguration"/>
  </tt:Data>
</tt:MessageDescription>
```

5.20.8 MetaDataConfiguration

```
<tt:MessageDescription IsProperty="true">
  <tt:Source>
    <tt:SimpleItemDescription Name="MetaDataConfigurationToken"
                Type="tt:ReferenceToken"/>
  </tt:Source>
  <tt:Data>
    <tt:ElementItemDescription Name="Config"
                 Type="tt:MetaDataConfiguration"/>
  </tt:Data>
</tt:MessageDescription>
```

5.21 Service specific data types

5.21.1 VideoSource

Representation of a physical video input.

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

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

- **Resolution**
  Horizontal and vertical resolution

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

5.21.2 AudioSource

Representation of a physical audio input.

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

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

5.21.3 Profile

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

An NVT shall provide at least one media profile at boot. An NVT 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 NVT and can be either static or created dynamically by the NVT. For example, the dynamic configurations can be created by the NVT depending on current available encoding resources.

```
<xs:complexType name="Profile"/>
<xs:attribute name="token" type= "tt:ReferenceToken use="required"/>
            <xs:attribute name="fixed" type="xs:boolean"/>
            <xs:element name="Name" type= "tt:Name/>
            <xs:element name="VideoSourceConfiguration" type=
```

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

- **token**
  Unique identifier of the profile.
- **fixed**
  A value of true signals that the profile cannot be deleted. Default is false.
- **Name**
  User readable name of the profile.
- **VideoSourceConfiguration**
  Optional configuration of the Video input.
- **AudioSourceConfiguration**
  Optional configuration of the Audio input.
- **VideoEncoderConfiguration**
  Optional configuration of the Video encoder.
- **AudioEncoderConfiguration**
  Optional configuration of the Audio encoder.
- **VideoAnalyticsConfiguration**
  Optional configuration of the video analytics module and rule engine.
- **PTZConfiguration**
  Optional configuration of the pan tilt zoom unit.
- **MetadataConfiguration**
  Optional configuration of the metadata stream.
- **Extension**
  Extensions defined in ONVIF 2.0

5.21.4 ProfileExtension

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

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

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

5.21.5 ConfigurationEntity

Base type defining the common properties of a configuration.

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

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

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

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

5.21.6 VideoSourceConfiguration

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

5.21.7 VideoSourceConfigurationOptions

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

- **BoundsRange**
  Supported range for the capturing area.

- **VideoSourceTokensAvailable**
  List of physical inputs.

5.21.8 VideoEncoderConfiguration

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

- **Resolution**
  Configured video resolution

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

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

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

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

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

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

5.21.9 VideoResolution

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

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

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

5.21.10 VideoRateControl

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

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

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

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

5.21.11 Mpeg4Configuration

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

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

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

5.21.12 H264Configuration

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

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

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

5.21.13 VideoEncoderConfigurationOptions

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

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

- **JPEG**
  Optional JPEG encoder settings ranges (Depricated: use Extension).

- **MPEG4**
  Optional MPEG-4 encoder settings ranges (Depricated: use Extension).

- **H264**
  Optional H.264 encoder settings ranges (Depricated: use Extension).

5.21.14 VideoEncoderOptionsExtension

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

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

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

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

5.21.15 JpegOptions

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

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

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

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

5.21.16 JpegOptions2

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

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

5.21.17 Mpeg4Options

```
<xs:complexType name="Mpeg4Options"/>
<xs:element name="ResolutionsAvailable" type= "tt:VideoResolution
            maxOccurs="unbounded"/>
            <xs:element name="GovLengthRange" type= "tt:IntRange/>
            <xs:element name="FrameRateRange" type= "tt:IntRange/>
            <xs:element name="EncodingIntervalRange" type= "tt:IntRange/>
            <xs:element name="Mpeg4ProfilesSupported"
            type="tt:Mpeg4Profile" maxOccurs="unbounded"/>
</xs:complexType>
```

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

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

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

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

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

5.21.18 Mpeg4Options2

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

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

5.21.19 H264Options

```
<xs:complexType name="H264Options"/>
```

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

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

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

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

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

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

5.21.20 H264Options2

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

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

5.21.21 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

5.21.22 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.

5.21.23 AudioEncoderConfiguration

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

```
            <xs:element name="Multicast" type= "tt:MulticastConfiguration/>
            <xs:element name="SessionTimeout" type="xs:duration"/>
</xs:complexType>
```

- **Encoding**
  Audio codec used for encoding the audio input (either G.711, G.726 or AAC)
- **Bitrate**
  The output bitrate in kbps.
- **SampleRate**
  The output sample rate in kHz.
- **Multicast**
  Defines the multicast settings that could be used for video streaming.
- **SessionTimeout**
  The rtsp session timeout for the related audio stream

5.21.24 AudioEncoderConfigurationOptions

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

- **Options**
  list of supported AudioEncoderConfigurations

5.21.25 AudioEncoderConfigurationOption

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

- **Encoding**
  The enoding used for audio data (either G.711, G.726 or AAC)
- **BitrateList**
  List of supported bitrates in kbps for the specified Encoding
- **SampleRateList**
  List of supported Sample Rates in kHz for the specified Encoding

5.21.26 VideoAnalyticsConfiguration

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

- **AnalyticsEngineConfiguration**
- **RuleEngineConfiguration**

5.21.27 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:complexType>
```

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

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

  - To get all events: Include the Events element but do not include a filter element.
  - To get no events: Do not include the Events element.
  - To get only some events: Include the Events element and include a filter in the element.

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

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

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

5.21.28 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

5.21.29 EventSubscription

Subcription handling in the same way as base notification subscription.

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

- **Filter**

- **SubscriptionPolicy**

5.21.30 MetadataConfigurationOptions

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

- **PTZStatusFilterOptions**

5.21.31 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 inforamtion.

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

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

5.21.32 VideoOutput

Representation of a physical video outputs.

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

- **Layout**

5.21.33 VideoOutputConfiguration

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

5.21.34 VideoDecoderConfigurationOptions

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

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

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

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

5.21.35 H264DecOptions

```
<xs:complexType name="H264DecOptions"/>
```

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

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

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

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

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

5.21.36 JpegDecOptions

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

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

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

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

5.21.37 Mpeg4DecOptions

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

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

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

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

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

5.21.38 AudioOutput

Representation of a physical audio outputs.

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

5.21.39 AudioOutputConfiguration

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

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

- **SendPrimacy**
  An audio channel MAY support different types of audio transmission. While for full duplex operation no special handling is required, in half duplex operation the transmission direction needs to be switched. The optional SendPrimacy parameter inside the AudioOutputConfiguration indicates which direction is currently active. An NVC 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 NVT in this mode.
  - www.onvif.org/ver20/HalfDuplex/Client The client is allowed to send audio data via the backchannel to the server. The NVT shall not send audio data to the client in this mode.
  - www.onvif.org/ver20/HalfDuplex/Auto It is up to the device how to deal with sending and receiving audio data.

  Acoustic echo cancellation is out of ONVIF scope.

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

5.21.40 AudioOutputConfigurationOptions

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

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

- **SendPrimacyOptions**
  An audio channel MAY support different types of audio transmission. While for full duplex operation no special handling is required, in half duplex operation the transmission direction needs to be switched. The optional SendPrimacy parameter inside the AudioOutputConfiguration indicates which direction is currently active. An NVC 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 NVT in this mode.
- www.onvif.org/ver20/HalfDuplex/Client The client is allowed to send audio data via the backchannel to the server. The NVT shall not send audio data to the client in this mode.
- www.onvif.org/ver20/HalfDuplex/Auto It is up to the device how to deal with sending and receiving audio data.

Acoustic echo cancellation is out of ONVIF scope.

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

5.21.41 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>
```

5.21.42 AudioDecoderConfigurationOptions

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

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

5.21.43 G711DecOptions

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

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

5.21.44 AACDecOptions

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

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

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

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

5.21.45 G726DecOptions

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

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

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

5.21.46 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 multicast port

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

- **AutoStart**
  true if the streaming is persistant (shall restart after a reboot)

5.21.47 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**

5.21.48 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

5.21.49 MediaUri

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

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

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

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

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

5.22 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 73: Media service specific fault codes

Fault Code	Parent Subcode Subcode	Fault Reason	Description
env:Receiver	ter:ActionNotSupported ter:AudioNotSupported	No audio capability	NVT does not support audio.
env:Receiver	ter:Action ter:MaxNVTProfiles	Maximum number reached	The maximum number of supported profiles has been reached.
env:Receiver	ter:ActionNotSupported ter:AudioOutputNotSupported	No audio output capability	Audio or Audio Outputs are not supported by the NVT
env:Receiver	ter:ActionNotSupported	No audio decoding	Audio or Audio Decoding is

	ter:AudioDecodingNotSupported	capability	not supported by the NVT
env:Receiver	ter:Action ter:IncompleteConfiguration	Configuration not complete	Entities required by this action are missing in the specified profile.
env:Receiver	ter:Action ter:ConfigurationConflict	Conflict when using new settings	The new settings conflicts with other uses of the configuration.
env:Sender	ter:InvalidArgVal ter:ProfileExists	Profile token already exists	A profile with the token ProfileToken already exists.
env:Sender	ter:InvalidArgVal ter:NoConfig	Configuration token does not exist	The requested configuration indicated by the ConfigurationToken does not exist.
env:Sender	ter:InvalidArgVal ter:NoProfile	Profile token does not exist	The requested profile token ProfileToken does not exist.
env:Sender	ter:Action ter:DeletionOfFixedProfile	Fixed profile can not be deleted	The fixed Profile cannot be deleted.
env:Sender	ter:InvalidArgVal ter:ConfigModify	Parameters can not be set	The configuration parameters are not possible to set.
env:Sender	ter:ActionNotSupported ter:VideoAnalyticsNot-Supported	No video analytics capability	NVT does not support video analytics.
env:Sender	ter:InvalidArgVal ter:InvalidStreamSetup	Invalid Stream setup	Specification of StreamType or Transport part in StreamSetup is not supported.
env:Sender	ter:OperationProhibited ter:StreamConflict	Stream conflict	Specification of StreamType or Transport part in StreamSetup causes conflict with other streams.
env:Sender	ter:InvalidArgVal ter:InvalidMulticastSettings	Invalid multicast configuration.	Not all configurations are configured for multicast..

Bibliography

[ONVIF Display WSDL] ONVIF Media WSDL, ver 2.0, 2010.
URL:http://www.onvif.org/onvif/ver10/network/wsdl/media.wsdl
[ONVIF Schema] ONVIF Schema, ver 2.0, 2010.
URL:http://www.onvif.org/onvif/ver10/schema/onvif.xsd
[ONVIF Topic Namespace] ONVIF Topic Namespace XML, ver 2.0, 2010.
URL:http://www.onvif.org/onvif/ver10/topics/topicns.xml