$\mathsf{ONVIF}^{\mathsf{TM}}$

-1-

ONVIF Media Configuration Test Specification

Version 15.06

July, 2015

$\ensuremath{\textcircled{}^{\circ}}$ 2015 by ONVIF, Inc. All rights reserved.

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

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

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

Revision History

Ver.	Date	Description
1.02.4	29th/Jul, 2011	First issue of Media Test Specification
11.12	22nd/Dec, 2011	New version numbering scheme has been applied.
		Requirement level terms have been removed.
		Audio outputs configuration test cases have been added.
		Audio & Video Multicast streaming test cases have been added.
		Start & Stop Multicast Streaming test case has been added.
12.06	6 th /Jun, 2012	Media Streaming IPv6 Multicast test cases have been added.
		Media Streaming with line breaks in base64 encoding test case has been added.
		Media Service Capabilities test cases have been added.
		Start & Stop Audio Multicast Streaming test cases have been added.
12.12	20 th /Dec, 2012	Following test cases was added:
		GET GUARANTEED NUMBER OF VIDEO ENCODER INSTANCES AND GET VIDEO ENCODER CONFIGURATION OPTIONS CONSISTENCY
		VIDEO ENCODER CONFIGURATIONS – ALL SUPPORTED VIDEO ENCODINGS
		AUDIO ENCODER CONFIGURATION OPTIONS AND AUDIO ENCODER CONFIGURATIONS CONSISTENCY (BITRATE AND SAMPLERATE)
		SET AUDIO ENCODER CONFIGURATION
		AUDIO ENCODER CONFIGURATION - MULTICAST PORT (IPv4)
		AUDIO ENCODER CONFIGURATION - MULTICAST ADDRESS (IPv4)
		VIDEO ENCODER CONFIGURATION - MULTICAST PORT (IPv4)
		VIDEO ENCODER CONFIGURATION -

www.onvif.org

MULTICAST ADDRESS (IPv4)
VIDEO ENCODER CONFIGURATION - MULTICAST ADDRESS AND PORT IN RTSP Setup (IPv4)
VIDEO ENCODER CONFIGURATION - MULTICAST ADDRESS AND PORT IN RTSP Setup (IPv6)
VIDEO ENCODER CONFIGURATION – JPEG RESOLUTION
VIDEO ENCODER CONFIGURATION – MPEG4 RESOLUTION
VIDEO ENCODER CONFIGURATION – H.264 RESOLUTION
MEDIA STREAMING – GUARANTEED NUMBER OF VIDEO ENCODER INSTANCES (RTP- Unicast/UDP)
MEDIA STREAMING – GUARANTEED NUMBER OF VIDEO ENCODER INSTANCES (RTP- Multicast/UDP)
MEDIA STREAMING – GUARANTEED NUMBER OF VIDEO ENCODER INSTANCES (RTP- Unicast/RTSP/HTTP/TCP)
MEDIA STREAMING – GUARANTEED NUMBER OF VIDEO ENCODER INSTANCES (RTP/RTSP/TCP)
MEDIA STREAMING – GUARANTEED NUMBER OF VIDEO ENCODER INSTANCES (MIX OF TRANSPORT TYPES)
AUDIO STREAMING – G.711 (RTP-Unicast/UDP)
AUDIO STREAMING – G.711 (RTP- Unicast/RTSP/HTTP/TCP)
AUDIO STREAMING – G.711 (RTP/RTSP/TCP)
AUDIO STREAMING – G.726 (RTP-Unicast/UDP)
AUDIO STREAMING – G.726 (RTP- Unicast/RTSP/HTTP/TCP)
AUDIO STREAMING – G.726 (RTP/RTSP/TCP)
AUDIO STREAMING – AAC (RTP-Unicast/UDP)
AUDIO STREAMING – AAC (RTP- Unicast/RTSP/HTTP/TCP)

	51)
MEDIA STREAMING – G.711 Multicast/UDP, IPv4)	(RTP-
MEDIA STREAMING – G.711 Multicast/UDP, IPv6)	(RTP-
MEDIA STREAMING – G.726 Multicast/UDP, IPv4)	(RTP-
MEDIA STREAMING – G.726 Multicast/UDP, IPv6)	(RTP-
MEDIA STREAMING – AAC (RTP-Multica IPv4)	st/UDP,
MEDIA STREAMING – AAC (RTP-Multica IPv6)	st/UDP,
START AND STOP MULTICAST STREAM G.711 (IPv4, ONLY AUDIO PROFILE)	MING –
START AND STOP MULTICAST STREAM G726 (IPv4, ONLY AUDIO PROFILE)	MING –
START AND STOP MULTICAST STREAM AAC (IPv4, ONLY AUDIO PROFILE)	MING –
VIDEO AND AUDIO EN CONFIGURATION – DIFFERENT PORTS	CODER
VIDEO AND AUDIO EN CONFIGURATION – DIFFERENT ADDRES	CODER SS
MEDIA STREAMING – JPEG (VALIDATIN HEADER EXTENSION)	IG RTP
13.06 June, 2013 Real Time Streaming Test Cases section related parts was moved to separate do Real Time Streaming Test Specification.	on and ocument
Current document name was changed from Test Specification to Media Configuration Specification.	n Media on Test
The following test cases were added or u with ID change:	updated
GET GUARANTEED NUMBER OF ENCODER INSTANCES AND GET ENCODER CONFIGURATION OF CONSISTENCY	VIDEO VIDEO PTIONS
VIDEO ENCODER CONFIGURATIONS SUPPORTED VIDEO ENCODINGS	– ALL
DYNAMIC MEDIA PROFILE CONFIGURAT	ION

www.onvif.org

VIDEO SOURCE CONFIGURATION
JPEG VIDEO ENCODER CONFIGURATION
MPEG4 VIDEO ENCODER CONFIGURATION
H.264 VIDEO ENCODER CONFIGURATION
VIDEO SOURCE CONFIGURATION USE COUNT (ADD SAME VIDEO SOURCE CONFIGURATION TO PROFILE TWICE)
VIDEO SOURCE CONFIGURATION USE COUNT (ADD DIFFERENT VIDEO SOURCE CONFIGURATIONS IN PROFILE)
VIDEO SOURCE CONFIGURATION USE COUNT (REMOVE VIDEO SOURCE CONFIGURATION)
VIDEO SOURCE CONFIGURATION USE COUNT (DELETION PROFILE WITH VIDEO SOURCE CONFIGURATION)
VIDEO SOURCE CONFIGURATION USE COUNT (SET VIDEO SOURCE CONFIGURATION)
VIDEO ENCODER CONFIGURATION USE COUNT (ADD SAME VIDEO ENCODER CONFIGURATION TO PROFILE TWICE)
VIDEO ENCODER CONFIGURATION USE COUNT (ADD DIFFERENT VIDEO ENCODER CONFIGURATIONS IN PROFILE)
VIDEO ENCODER CONFIGURATION USE COUNT (REMOVE VIDEO ENCODER CONFIGURATION)
VIDEO ENCODER CONFIGURATION USE COUNT (PROFILE DELETION WITH VIDEO ENCODER CONFIGURATION)
VIDEO ENCODER CONFIGURATION USE COUNT (SET VIDEO ENCODER CONFIGURATION)
VIDEO ENCODER CONFIGURATIONS – ALL SUPPORTED VIDEO ENCODINGS (ALL VIDEO ENCODER CONFIGURATIONS)
AUDIO SOURCE CONFIGURATION
AUDIO ENCODER CONFIGURATION
G.711 AUDIO ENCODER CONFIGURATION
G.726 AUDIO ENCODER CONFIGURATION

AAC AUDIO ENCODER CONFIGURATION
GET AUDIO SOURCE CONFIGURATION – INVALID CONFIGURATIONTOKEN
GET AUDIO SOURCE CONFIGURATION OPTIONS – INVALID PROFILETOKEN
GET AUDIO SOURCE CONFIGURATION OPTIONS – INVALID CONFIGURATION TOKEN
SET AUDIO SOURCE CONFIGURATION – INVALID TOKEN
SET AUDIO ENCODER CONFIGURATION
AUDIO SOURCE CONFIGURATION USE COUNT (ADD SAME AUDIO SOURCE CONFIGURATION TO PROFILE TWICE)
AUDIO SOURCE CONFIGURATION USE COUNT (ADD DIFFERENT AUDIO SOURCE CONFIGURATIONS IN PROFILE)
AUDIO SOURCE CONFIGURATION USE COUNT (REMOVE AUDIO SOURCE CONFIGURATION)
AUDIO SOURCE CONFIGURATION USE COUNT (PROFILE DELETION WITH AUDIO SOURCE CONFIGURATION)
AUDIO SOURCE CONFIGURATION USE COUNT (SET AUDIO SOURCE CONFIGURATION)
AUDIO ENCODER CONFIGURATION USE COUNT (ADD SAME AUDIO ENCODER CONFIGURATION TO PROFILE TWICE)
AUDIO ENCODER CONFIGURATION USE COUNT (ADD DIFFERENT AUDIO ENCODER CONFIGURATIONS IN PROFILE)
AUDIO ENCODER CONFIGURATION USE COUNT (REMOVE AUDIO ENCODER CONFIGURATION)
AUDIO ENCODER CONFIGURATION USE COUNT (DELETION PROFILE WITH AUDIO ENCODER CONFIGURATION)
AUDIO ENCODER CONFIGURATION USE COUNT (SET AUDIO ENCODER CONFIGURATION)
SET AUDIO OUTPUT CONFIGURATION
SET AUDIO OUTPUT CONFIGURATION – INVALID CONFIGURATION

-7-

		SET AUDIO OUTPUT CONFIGURATION – INVALID OUTPUTTOKEN
		PTZ CONFIGURATION
		METADATA CONFIGURATION
		SOAP FAULT MESSAGE
		START MULTICAST - INVALID PROFILE TOKEN
		The following Annexes were added or updated:
		Annex A.3 Create Empty Profile
		Annex A.4 Get Guaranteed Number of Video Encoder Instances and Get Video Encoder Configuration Options Consistency Verification Description
13.12	Dec, 2013	Pre-requisites were updated for the following test cases:
		SOAP FAULT MESSAGE
		SOAP FAULT MESSAGE
		START MULTICAST - INVALID PROFILE TOKEN
14.12	Dec, 2014	The following test cases were updated with ID change:
		SET AUDIO OUTPUT CONFIGURATION – INVALID CONFIGURATION
		DYNAMIC MEDIA PROFILE CONFIGURATION
		AUDIO SOURCE CONFIGURATION
		AUDIO ENCODER CONFIGURATION
		PTZ CONFIGURATION
		METADATA CONFIGURATION
15.06	Jul, 2015	The following test cases were added:
		GET AUDIO OUTPUT CONFIGURATION OPTIONS
		AUDIO OUTPUT CONFIGURATION
		AUDIO DECODER CONFIGURATION
		PROFILES AND AUDIO OUTPUT CONFIGURATIONS CONSISTENCY
		AUDIO OUTPUT CONFIGURATIONS AND AUDIO OUTPUT CONFIGURATION CONSISTENCY

-8-

AUDIO OUTPUT CONFIGURATIONS AND AUDIO OUTPUT CONFIGURATION OPTIONS CONSISTENCY
PROFILES AND AUDIO OUTPUT CONFIGURATION OPTIONS CONSISTENCY
AUDIO OUTPUT CONFIGURATIONS AND AUDIO OUTPUTS CONSISTENCY
PROFILES AND AUDIO DECODER CONFIGURATIONS CONSISTENCY
AUDIO DECODER CONFIGURATIONS AND AUDIO DECODER CONFIGURATION CONSISTENCY

-9-

Table of Contents

1	Intro	oduction	14
	1.1	Scope	14
	1.1.1	.1 Media Configuration	14
2	Tern	ms and Definitions	16
	2.1	Definitions	16
	2.2	Abbreviations	16
3	Test	st Overview	17
	3.1	Test Setup	17
	3.1.1	.1 Network Configuration for DUT	17
	3.2	Prerequisites	18
	3.3	Test Policy	18
	3.3.1	.1 Media Configuration	18
4	Med	dia Configuration Test Cases	19
	4.1	Media Profile	19
	4.1.1	.1 MEDIA PROFILE CONFIGURATION	19
	4.1.2	.2 PROFILES CONSISTENCY	20
	4.1.3	.3 DYNAMIC MEDIA PROFILE CONFIGURATION	21
	4.2	Video Configuration	27
	4.2.1	.1 VIDEO ENCODER CONFIGURATION	27
	4.2.2	.2 GUARANTEED NUMBER OF VIDEO ENCODER INSTANCES	28
	4.2.3 ENC	.3 GET GUARANTEED NUMBER OF VIDEO ENCODER INSTANCES AND GET V CODER CONFIGURATION OPTIONS CONSISTENCY	/IDEO 30
	4.2.4	.4 VIDEO SOURCE CONFIGURATION	34
	4.2.5	.5 JPEG VIDEO ENCODER CONFIGURATION	39
	4.2.6	.6 MPEG4 VIDEO ENCODER CONFIGURATION	43
	4.2.7	.7 H.264 VIDEO ENCODER CONFIGURATION	47
	4.3	Video Source Configuration	53
	4.3.1	.1 VIDEO SOURCE CONFIGURATIONS AND PROFILES CONSISTENCY	53
	4.3.2 CON	.2 VIDEO SOURCE CONFIGURATIONS AND VIDEO SOURCE CONFIGURA	ATION 54
	4.3.3 CON	.3 VIDEO SOURCE CONFIGURATIONS AND VIDEO SOURCE CONFIGURATION OPT	TIONS 56
	4.3.4	.4 PROFILES AND VIDEO SOURCE CONFIGURATION OPTIONS CONSISTENCY	58
	4.3.5	.5 VIDEO SOURCE CONFIGURATIONS AND VIDEO SOURCES CONSISTENCY	60
	4.3.6	.6 VIDEO SOURCE CONFIGURATION USE COUNT (CURRENT STATE)	62
	4.3.7 CON	.7 VIDEO SOURCE CONFIGURATION USE COUNT (ADD SAME VIDEO SOUNTION TO PROFILE TWICE)	URCE 64
	4.3.8 CON	.8 VIDEO SOURCE CONFIGURATION USE COUNT (ADD DIFFERENT VIDEO SOUNTIONS IN PROFILE)	URCE 67
	4.3.9	.9 VIDEO SOURCE CONFIGURATION USE COUNT (REMOVE VIDEO SO	URCE
DΝV	/IF	www.onvif.org info@onvif.org	

ONFIGURATION

CONFIGURATION)72
4.3.10 VIDEO SOURCE CONFIGURATION USE COUNT (DELETION PROFILE WITH VIDEO SOURCE CONFIGURATION)
4.3.11 VIDEO SOURCE CONFIGURATION USE COUNT (SET VIDEO SOURCE CONFIGURATION)
4.4 Video Encoder Configuration
4.4.1 VIDEO ENCODER CONFIGURATIONS AND PROFILES CONSISTENCY
4.4.2 VIDEO ENCODER CONFIGURATIONS AND VIDEO ENCODER CONFIGURATION CONSISTENCY
4.4.3 VIDEO ENCODER CONFIGURATIONS AND VIDEO ENCODER CONFIGURATION OPTIONS CONSISTENCY
4.4.4 PROFILES AND VIDEO ENCODER CONFIGURATION OPTIONS CONSISTENCY
4.4.5 VIDEO ENCODER CONFIGURATION USE COUNT (CURRENT STATE)
4.4.6 VIDEO ENCODER CONFIGURATIONS – ALL SUPPORTED VIDEO ENCODINGS
4.4.7 VIDEO ENCODER CONFIGURATION USE COUNT (ADD SAME VIDEO ENCODER CONFIGURATION TO PROFILE TWICE)
4.4.8 VIDEO ENCODER CONFIGURATION USE COUNT (ADD DIFFERENT VIDEO ENCODER CONFIGURATIONS IN PROFILE)
4.4.9 VIDEO ENCODER CONFIGURATION USE COUNT (REMOVE VIDEO ENCODER CONFIGURATION)
4.4.10 VIDEO ENCODER CONFIGURATION USE COUNT (PROFILE DELETION WITH VIDEO ENCODER CONFIGURATION)
4.4.11 VIDEO ENCODER CONFIGURATION USE COUNT (SET VIDEO ENCODER CONFIGURATION)
4.4.12 VIDEO ENCODER CONFIGURATIONS – ALL SUPPORTED VIDEO ENCODINGS (ALL VIDEO ENCODER CONFIGURATIONS)
4.5 Audio Configuration
4.5.1 GET AUDIO SOURCE CONFIGURATION OPTIONS
4.5.2 G.711 AUDIO ENCODER CONFIGURATION
4.5.3 G.726 AUDIO ENCODER CONFIGURATION
4.5.4 AAC AUDIO ENCODER CONFIGURATION
4.5.5 GET AUDIO SOURCE CONFIGURATION – INVALID CONFIGURATIONTOKEN
4.5.6 GET AUDIO SOURCE CONFIGURATION OPTIONS – INVALID PROFILETOKEN
4.5.7 GET AUDIO SOURCE CONFIGURATION OPTIONS – INVALID CONFIGURATION TOKEN 130
4.5.8 SET AUDIO SOURCE CONFIGURATION – INVALID TOKEN
4.5.9 SET AUDIO ENCODER CONFIGURATION
4.5.10 AUDIO SOURCE CONFIGURATION
4.5.11 AUDIO ENCODER CONFIGURATION
4.6 Audio Source Configuration
4.6.1 AUDIO SOURCE CONFIGURATIONS AND PROFILES CONSISTENCY
4.6.2 AUDIO SOURCE CONFIGURATIONS AND AUDIO SOURCE CONFIGURATION



4.6.3 AUDIO SOURCE CONFIGURATIONS AND AUDIO SOURC CONSISTENCY	E CONFIGURATION OPTIONS
4.6.4 PROFILES AND AUDIO SOURCE CONFIGURATION OPTIC	NS CONSISTENCY 152
4.6.5 AUDIO SOURCE CONFIGURATIONS AND AUDIO SOURCE	S CONSISTENCY 153
4.6.6 AUDIO SOURCE CONFIGURATION USE COUNT (CURREN	NT STATE) 155
4.6.7 AUDIO SOURCE CONFIGURATION USE COUNT (AI CONFIGURATION TO PROFILE TWICE)	DD SAME AUDIO SOURCE
4.6.8 AUDIO SOURCE CONFIGURATION USE COUNT (ADD CONFIGURATIONS IN PROFILE)	DIFFERENT AUDIO SOURCE
4.6.9 AUDIO SOURCE CONFIGURATION USE COUNT (CONFIGURATION)	REMOVE AUDIO SOURCE
4.6.10 AUDIO SOURCE CONFIGURATION USE COUNT (PROF SOURCE CONFIGURATION)	ILE DELETION WITH AUDIO
4.6.11 AUDIO SOURCE CONFIGURATION USE COUNT CONFIGURATION)	(SET AUDIO SOURCE
4.7 Audio Encoder Configuration	
4.7.1 AUDIO ENCODER CONFIGURATIONS AND PROFILES CO	NSISTENCY 175
4.7.2 AUDIO ENCODER CONFIGURATIONS AND AUDIO	ENCODER CONFIGURATION
4.7.3 AUDIO ENCODER CONFIGURATIONS AND AUDIO	ENCODER CONFIGURATION
4.7.4 PROFILES AND AUDIO ENCODER CONFIGURATION OPT	ONS CONSISTENCY 180
4.7.5 AUDIO ENCODER CONFIGURATION USE COUNT (CURRE	ENT STATE)
4.7.6 AUDIO ENCODER CONFIGURATION OPTIONS CONFIGURATIONS CONSISTENCY (BITRATE AND SAMPLERATE)	AND AUDIO ENCODER
4.7.7 AUDIO ENCODER CONFIGURATION USE COUNT (AE CONFIGURATION TO PROFILE TWICE)	DD SAME AUDIO ENCODER 185
4.7.8 AUDIO ENCODER CONFIGURATION USE COUNT (ADD E CONFIGURATIONS IN PROFILE)	DIFFERENT AUDIO ENCODER 190
4.7.9 AUDIO ENCODER CONFIGURATION USE COUNT (I CONFIGURATION)	REMOVE AUDIO ENCODER 196
4.7.10 AUDIO ENCODER CONFIGURATION USE COUNT (DELE ENCODER CONFIGURATION)	TION PROFILE WITH AUDIO
4.7.11 AUDIO ENCODER CONFIGURATION USE COUNT CONFIGURATION)	(SET AUDIO ENCODER
4.8 Audio Output Configuration	
4.8.1 SET AUDIO OUTPUT CONFIGURATION	
4.8.2 SET AUDIO OUTPUT CONFIGURATION – INVALID OUTPU	TTOKEN210
4.8.3 SET AUDIO OUTPUT CONFIGURATION – INVALID CONFIG	SURATION212
4.8.4 GET AUDIO OUTPUT CONFIGURATION OPTIONS	
4.8.5 AUDIO OUTPUT CONFIGURATION	
4.8.6 AUDIO DECODER CONFIGURATION	
4.8.7 AUDIO OUTPUT CONFIGURATIONS AND AUDIO OUTPU CONSISTENCY	T CONFIGURATION OPTIONS





	4.8.8		PROFILI	ES AND AUD		CONFIG	URATIO		ONS CONSIS	TENC	Y229
	4.8.9		PROFILI	ES AND AUD		CONFIG	URATIO	ONS CON	ISISTENCY		232
	4.8.1 CON	0 SIST	AUDIO ENCY	OUTPUT	CONFIGUE	RATIONS	AND	AUDIO	OUTPUT	CONF	IGURATION
	4.8.1	1	AUDIO (OUTPUT CO	NFIGURATI	ONS AND	AUDIO	OUTPUT	S CONSIST	ENCY.	235
	4.8.1	2	PROFIL	ES AND AUD	IO DECODI	ER CONFI	GURAT	IONS CO	ONSISTENCY	<i>(</i>	237
	4.8.1 CON	3 SIST	AUDIO ENCY	DECODER	CONFIGU	RATIONS	AND	AUDIO	DECODER	CONF	IGURATION 239
4.9		PTZ	Configura	ation							241
	4.9.1		PTZ CO	NFIGURATIO	ONS AND PF	ROFILES	CONSIS	STENCY.			241
	4.9.2		PTZ CO	NFIGURATIO	ON						243
4.1	0	Meta	data Con	figuration							246
	4.10.	1	METADA	ATA CONFIG	URATION						246
4.1 [′]	1	Medi	a Stream	ing							251
	4.11.	1	SNAPSH	HOT URI							251
4.1	2	Error Handling									
	4.12.1 SOAP FAULT MESSAGE				253						
	4.12.2 SOAP FAULT MESSAGE		254								
	4.12.	2.3 START MULTICAST - INVALID PROFILE TOKEN				255					
4.1	3	Capa	apabilities								
	4.13.	3.1 MEDIA SERVICE CAPABILITIES2				257					
	4.13.2 GET SERVICES AND GET MEDIA SERVICE CAPABILITIES CONSISTENCY			258							
Anne	x A										260
A.1		Inval	id Media	Profile							
A.2		StreamSetup syntax for GetStreamUri									
A.3		Create Empty Profile									
A.4 Opt	tions	Get Cons	Guarante istency V	ed Number /erification De	of Video E escription	ncoder In	stances	s and Ge	et Video End	coder (Configuration
A.5		Name and Token Parameters									
A.6		Crea	te or con	figure Empty	Profile						
A.7		Rest	ore media	a profile							
A.8		Delete media profile									

1 Introduction

The goal of the ONVIF test specification set is to make it possible to realize fully interoperable IP physical security implementation from different vendors. The set of ONVIF test specification describes the test cases need to verify the [ONVIF Network Interface Specs] and [ONVIF Conformance] requirements. And also the test cases are to be basic inputs for some Profile specification requirements. It also describes the test framework, test setup, pre-requisites, test policies needed for the execution of the described test cases.

-14-

This ONVIF Media Test Specification acts as a supplementary document to the [ONVIF Network Interface Specs], illustrating test cases need to be executed and passed. And also this specification acts as an input document to the development of test tool which will be used to test the ONVIF device implementation conformance towards ONVIF standard. This test tool is referred as ONVIF Client hereafter.

1.1 Scope

This ONVIF Media Test Specification defines and regulates the conformance testing procedure for the ONVIF conformant devices. Conformance testing is meant to be functional black-box testing. The objective of this specification is to provide test cases to test individual requirements of ONVIF devices according to ONVIF Media Service which is defined in [ONVIF Network Interface Specs].

The principal intended purposes are:

- 1. Provide self-assessment tool for implementations.
- 2. Provide comprehensive test suite coverage for [ONVIF Network Interface Specs].

This specification does not address the following.

- 1. Product use cases and non-functional (performance and regression) testing.
- 2. SOAP Implementation Interoperability test i.e. Web Service Interoperability Basic Profile version 2.0 (WS-I BP 2.0).
- 3. Network protocol implementation Conformance test for HTTP, HTTPS, RTP and RTSP protocol.
- 4. Poor streaming performance test (audio/video distortions, missing audio/video frames, incorrect lib synchronization etc.).
- 5. Wi-Fi Conformance test

The set of ONVIF Test Specification will not cover the complete set of requirements as defined in [ONVIF Network Interface Specs]; instead it would cover subset of it. The scope of this specification is to derive all the normative requirements of [ONVIF Network Interface Specs] which are related to ONVIF Media Service and some of the optional requirements.

This ONVIF Media Test Specification covers ONVIF Media service which is a functional block of [ONVIF Network Interface Specs]. The following sections describe the brief overview of and scope of each functional block.

1.1.1 Media Configuration

Media Configuration section covers the test cases needed for the verification of media service features as mentioned in [ONVIF Network Interface Specs]. Media service is used to configure the

media and other real time streaming configurations.

Briefly it covers the following things

- 1. Manage media profiles.
- 2. Manage configuration entities.
- 3. Initiate and manipulate Audio/Video streams for different media formats.

The scope of this specification is to cover following configuration entities and Audio/Video media formats.

-15-

- 1. Configuration Entities:
 - a. Video source configuration
 - b. Audio source configuration
 - c. Video encoder configuration
 - d. Audio encoder configuration
 - e. PTZ configuration
 - f. Metadata configuration
- 2. Video Codec:
 - a. JPEG QVGA
 - b. MPEG-4, Simple Profile
 - c. H.264
- 3. Audio Codec:
 - a. G.711
 - b. G.726
 - c. AAC



2 Terms and Definitions

2.1 Definitions

This section defines terms that are specific to the ONVIF Media Service and tests. For a list of applicable general terms and definitions, please see [ONVIF Base Test].

-16-

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.
Media Profile	A media profile maps a video and/or audio source to a video and/or an audio encoder, PTZ and analytics configurations.

2.2 Abbreviations

This section describes abbreviations used in this document.

AAC	Advanced Audio Coding		
JPEG	Joint Photographic Experts Group		
MPEG-4	Moving Pictures Experts Group-4		
QVGA	Quarter Video Graphics Array		
TTL	Time To Live		

3 Test Overview

This section describes about the test setup and prerequisites needed, and the test policies that should be followed for test case execution.

-17-

3.1 Test Setup

3.1.1 Network Configuration for DUT

The generic test configuration for the execution of test cases defined in this document is as shown below (Figure 1)

Based on the individual test case requirements, some of the entities in the below setup may not be needed for the execution of those corresponding test cases.



Figure 1: Test Configuration for DUT

DUT: ONVIF device to be tested. Hereafter, this is referred to as DUT (Device Under Test).

ONVIF Client (Test Tool): Tests are executed by this system and it controls the behavior of the DUT. It handles both expected and unexpected behavior.

HTTP Proxy: provides facilitation in case of RTP and RTSP tunneling over HTTP.

Wireless Access Point: provides wireless connectivity to the devices that support wireless connection.

DNS Server: provides DNS related information to the connected devices.

DHCP Server: provides IPv4 Address to the connected devices.

NTP Server: provides time synchronization between ONVIF Client and DUT.

Switching Hub: provides network connectivity among all the test equipments in the test environment. All devices should be connected to the Switching Hub.

Router: provides router advertisements for IPv6 configuration.

3.2 Prerequisites

The pre-requisites for executing the test cases described in this Test Specification are

- 1. The DUT shall be configured with an IPv4 address.
- 2. The DUT shall be IP reachable [in the test configuration].
- 3. The DUT shall be able to be discovered by the Test Tool.
- 4. The DUT shall be configured with the time i.e. manual configuration of UTC time and if NTP is supported by DUT then NTP time shall be synchronized with NTP Server.
- 5. The DUT time and Test tool time shall be synchronized with each other either manually or by common NTP server.

3.3 Test Policy

This section describes the test policies specific to the test case execution of each functional block.

The DUT shall adhere to the test policies defined in this section.

3.3.1 Media Configuration

Prior to the execution of Media Configuration test cases, DUT shall be discovered by ONVIF Client and it shall demonstrate media capabilities to ONVIF Client using device management service.

DUT shall support at least one media profile with Video Configuration. Video Configuration shall include video source and video encoder media entities.

DUT shall support JPEG QVGA encoding.

ONVIF Client shall explicitly specify the optional media formats supported by DUT.

ONVIF Client shall explicitly specify if the DUT supports Audio and PTZ.

DUT shall allow for creation of at least one media profile by ONVIF Client. In certain test cases, ONVIF Client may create new media configuration (i.e. media profile and media entities). In such cases, the test procedure will delete those modified configuration at the end of the test procedure.

DUT should respond with proper response message for all SOAP actions. Sending fault messages such as "ter:ConfigurationConflict" will be treated as FAILURE of the test cases.

Please refer to Section 4 for Media Configuration Test Cases.

4 Media Configuration Test Cases

4.1 Media Profile

4.1.1 MEDIA PROFILE CONFIGURATION

Test Label: Media Configuration Media Profiles.

Test Case ID: MEDIA-1-1-1

ONVIF Core Specification Coverage: Get media profiles

Command Under Test: GetProfiles

WSDL Reference: media.wsdl

Test Purpose: To retrieve existing media profiles of DUT and the corresponding media entities (video source and video encoder).

-19-

Pre-Requisite: None

Test Configuration: ONVIF Client and DUT

Test Sequence:



Test Procedure:

- 1. Start an ONVIF Client.
- 2. Start the DUT.
- 3. ONVIF Client will invoke GetProfilesRequest message to retrieve existing media profiles of the DUT.
- 4. Verify that the DUT returns at least one media profile with video configuration (video source and video encoder) in the GetProfilesResponse message.
- 5. Verify that all media profile elements have 'fixed' attribute.

Test Result:

PASS –

DUT passes all assertions.

FAIL -

DUT did not send GetProfilesResponse message.

DUT has no default media profile configuration.

DUT did not provide at least one media profile with video source and video encoder configurations.

One or more media profiles don't have 'fixed' attribute.

4.1.2 PROFILES CONSISTENCY

Test Label: Media Service DUT GetProfiles Command and GetProfile Command Consistency Validation.

Test Case ID: MEDIA-1-1-3

ONVIF Core Specification Coverage: Get media profiles, Get media profile

Command Under Test: GetProfiles, GetProfile

WSDL Reference: media.wsdl

Test Propose: To check that GetProfile command and GetProfiles command are consistent.

Pre-Requisite: ONVIF Client gets the Media Service entry point by GetCapabilities command.

Test Configuration: ONVIF Client and DUT

Test Sequence:



Test Procedure:

1. Start an ONVIF Client.

- 2. Start the DUT.
- 3. ONVIF Client invokes GetProfilesRequest message to retrieve list of profiles and their Configurations
- 4. The DUT sends GetProfilesResponse message.
- 5. Check that each Profile from GetProfilesResponse message has unique token.
- 6. ONVIF Client invokes GetProfileRequest (ProfileToken) message to retrieve profile from device.
- 7. The DUT sends GetProfileResponse message.
- 8. Verify that all parameters and their values for the same profile from GetProfilesResponse message and GetProfileResponse message are same.
- 9. Repeat steps 6-8 for other profiles from the GetProfilesResponse message.

Test Result:

PASS –

DUT passes all assertions.

FAIL –

The DUT did not send GetProfilesResponse message.

The DUT did not send valid GetProfilesResponse message.

The DUT return two or more Profiles in GetProfilesResponse message with the same ProfileToken.

The DUT did not send GetProfileResponse message.

The DUT did not send valid GetProfileResponse message.

The DUT did not send equal parameters in the GetProfileResponse message and in the GetProfilesResponse message for the same profile.

4.1.3 DYNAMIC MEDIA PROFILE CONFIGURATION

Test Label: Media Configuration Dynamic Media Profile Configuration.

Test Case ID: DRAFT-MEDIA-1-1-5

ONVIF Core Specification Coverage: Create media profile, Get media profiles, Get media profile, Add video source configuration to a profile, Add video encoder configuration to a profile, Remove video source configuration from a profile, Remove video encoder configuration from a profile, Delete media profile.

Command Under Test: None

WSDL Reference: media.wsdl

Test Purpose: To verify the behavior of the DUT for dynamic media profile configuration.

Pre-Requisite: None

ONVIF

-22-

Test Configuration: ONVIF Client and DUT

Test Sequence:

ONV	IF Client	DUT
	GetProfilesRequest (Empty message)	Start DUT
Receive and validate existing media profiles.	GetProfilesResponse (Existing media profiles)	
	CreateProfileRequest (Name = "testprofileX")	•
Receive and validate create profile response	CreateProfileResponse (Empty profile with token) or SOAP 1.2 fault message (Action/MaxNVTProfiles)	Create a new media profile.
	Annex A.3	
	AddVideoSourceConfigurationRequest (Profile Token , video source cfg)	
Receive add video source configuration response.	AddVideoSourceConfigurationResponse (empty message)	Add video source configuration to a given profile token
	AddVideoEncoderConfigurationRequest (Profile Token, video encoder cfg)	
Receive add video encoder configuration response.	AddVideoEncoderConfigurationResponse (empty message)	 Add video encoder configuration to a given profile token
	GetProfileRequest (Profile Token)	
Receive and validate media profile	GetProfileResponse (media profile configuration)	Return media profile configuration
configuration.		

ONVIF Client		DUT
	GetProfilesRequest	
Receive and validate media profile list.	GetProfilesResponse (media profile list)	► Return media profile list.
	RemoveVideoEncoderConfiguration Request (Profile Token)	Remove Video Encoder
	RemoveVideoEncoderConfiguration Response (empty message)	Cfg from media profile.
Receive the message	 RemoveVideoSourceConfiguration Request (Profile Token) 	_
Receive the message	RemoveVideoSourceConfiguration Response (empty message)	Remove Video Source Cfg from media profile.
	DeleteProfileRequest (Profile Token)	-
	DeleteProfileResponse (empty message)	Delete media profile.
Receive delete response.	GetProfileRequest (Profile Token)	_
Receive and validate SOAP 1.2 fault message	SOAP 1.2 fault message (InvalidArgVal/NoProfile)	Generate SOAP 1.2 fault message

Test Procedure:

- 1. Start an ONVIF Client.
- 2. Start the DUT.
- 3. ONVIF Client will invoke GetProfilesRequest message to retrieve existing media profiles configurations of the DUT.
- 4. Verify that the DUT returns at least one media profile with video configuration (video source and video encoder) in GetProfilesResponse message.
- 5. ONVIF Client will invoke CreateProfileRequest message to create a new empty media profile.
- 6. The DUT returns an empty profile with no profile entities in the CreateProfileResponse message or SOAP 1.2 fault message (Action/MaxNVTProfiles) from the DUT. If fault was received, execute Annex A.3.
- 7. ONVIF Client will invoke AddVideoSourceConfigurationRequest message (Profile Token, Reference to Video Source Configuration of existing media profile) to add video source configuration to new profile.
- 8. The DUT sends AddVideoSourceConfigurationResponse message indicating successful addition of video source configuration.
- 9. ONVIF Client will invoke AddVideoEncoderConfigurationRequest message (Profile Token, Reference to Video Encoder Configuration of existing media profile) to add video encoder configuration to new profile.
- 10. The DUT sends AddVideoEncoderConfigurationResponse message indicating successful addition of video encoder configuration.
- 11. ONVIF Client will invoke GetProfileRequest (Profile Token) message to verify video source and encoder configurations in a new profile.
- 12. The DUT will return media profile configuration for requested media profile in the GetProfileResponse message.
- 13. ONVIF Client will invoke GetProfilesRequest message to verify video source and encoder configurations in a new profile.
- 14. The DUT will return media profile list with newly created profile in the GetProfilesResponse message.
- 15. ONVIF Client will invoke RemoveVideoEncoderConfigurationRequest message (Profile Token) to remove video encoder configuration from a media profile.
- 16. The DUT sends RemoveVideoEncoderConfigurationResponse message indicating successfully removal of video encoder configuration.
- 17. ONVIF Client will invoke RemoveVideoSourceConfigurationRequest message (Profile Token) to remove video source configuration from a media profile.
- 18. The DUT sends RemoveVideoSourceConfigurationResponse message indicating successfully removal of video source configuration.
- 19. If profile with ProfileToken = Profile Token was created during test execution, then ONVIF Client will invoke DeleteProfileRequest (Profile Token) message to delete the newly created media profile. Otherwise, ONVIF client skip rest steps and restore profile settings.

20. The DUT will delete the media profile and sends DeleteProfileResponse message.

-26-

- 21. ONVIF Client will invoke GetProfileRequest (deleted Profile Token) message to check the existence of deleted media profile.
- 22. The DUT will generate SOAP 1.2 fault message (InvalidArgVal/NoProfile).

Test Result:

PASS –

The DUT passed all assertions.

FAIL –

The DUT did not send GetProfilesResponse message.

The DUT did not send CreateProfileResponse message.

The DUT did not send AddVideoSourceConfigurationResponse message.

The DUT did not send AddVideoEncoderConfigurationResponse message.

The DUT did not send GetProfileResponse message.

The DUT did not set 'fixed' attribute of created media profile to 'false'

The DUT did not send RemoveVideoEncoderConfigurationResponse message.

The DUT did not send RemoveVideoSourceConfigurationResponse message.

The DUT did not send DeleteProfileResponse message.

The DUT did not send SOAP 1.2 fault message (InvalidArgVal/NoProfile).

Note: See Annex in [ONVIF Base Test] for Invalid SOAP 1.2 fault message definition.

Note: See Annex A.5 for Name and Token Parameters Length limitations.

Note: If profile was deleted during Annex A.3 execution, ONVIF Client restores the deleted profile and profile settings.

4.2 Video Configuration

4.2.1 VIDEO ENCODER CONFIGURATION

Test Label: Media Configuration Video Encoder Configuration

Test Case ID: MEDIA-2-1-2

ONVIF Core Specification Coverage: Get media profiles, Get video encoder configurations, Get compatible video encoder configurations

-27-

Command Under Test: None

WSDL Reference: media.wsdl

Test Purpose: To verify DUT Video Encoder Configuration Operations

Pre-Requisite: None

Test Configuration: ONVIF Client and DUT

Test Sequence:

ONVIF Client		DU	JT
	GetProfilesRequest (Empty)		Start DUT
Receive and Validate Media Profiles	GetProfilesResponse (Media Profiles)	→	Send configured media profiles
	GetCompatibleVideoEncoder ConfigurationsRequest (Media Profile Token)	-	Send Video Encoder Configurations compatible with the received media profile
	GetCompatibleVideoEncoder ConfigurationsResponse (Video Encoder Configurations)		
Receive and Validate			
Video Encoder Configurations	 GetVideoEncoderConfigurationsRequest (Empty) 		Send all Video Encoder Configurations
Receive and Validate Video Encoder Configurations	GetVideoEncoderConfigurations Response (Video Encoder Configurations)		g

Test Procedure:

- 1. Start an ONVIF Client.
- 2. Start the DUT.
- 3. ONVIF Client invokes GetProfiles Request.
- 4. DUT sends the list of existing media profiles in GetProfiles Response message.
- 5. ONVIF Client invokes GetCompatibleVideoEncoderConfigurations request with one of the received media profile tokens as input argument.
- 6. DUT sends the list of video encoder configurations, compatible with the received media profile token.
- 7. ONVIF Client verifies the list of video encoder configurations sent by DUT.
- 8. ONVIF Client will invoke GetVideoEncoderConfigurations request to retrieve the list of video encoder configurations supported by the DUT.
- 9. DUT sends the list of all video encoder configurations supported by it.
- 10. ONVIF Client verifies the list of video encoder configurations sent by the DUT.

Test Result:

PASS –

DUT passes all assertions.

FAIL –

DUT did not send GetProfilesResponse message.

DUT did not send valid GetCompatibleVideoEncoderConfigurationsResponse message.

DUT did not send valid GetVideoEncoderConfigurationsResponse message.

4.2.2 GUARANTEED NUMBER OF VIDEO ENCODER INSTANCES

Test Label: Media Configuration DUT Video Encoder Instances

Test Case ID: MEDIA-2-1-6

ONVIF Core Specification Coverage: Get video source configurations, Get guaranteed number of video encoder instances

Command Under Test: GetGuaranteedNumberOfVideoEncoderInstances

WSDL Reference: media.wsdl

Test Purpose: To retrieve minimum number of video encoder instances supported by DUT per video source configuration.

Pre-Requisite: None

Test Configuration: ONVIF Client and DUT

Test Sequence:

ONVIF Client		DL	JT
	GetVideoSourceConfigurationsRequest (Empty)		Start DUT
Receive and Validate Video Source Configurations	GetVideoSourceConfigurationsResponse (Video Source Configurations)	→	Send all video source configurations
	GetGuaranteedNumberOfVideoEncoder InstancesRequest (Video Source Config token)		
	GetGuaranteedNumberOfVideoEncoder InstancesResponse (Number of video encoder instances)	→	Send supported number of video encoder instances

-29-

Test Procedure:

- 1. Start an ONVIF Client.
- 2. Start the DUT.
- 3. ONVIF Client invokes GetVideoSourceConfigurations request.
- 4. DUT sends the list of video source configurations supported by it.
- 5. ONVIF Client invokes GetGuaranteedNumberOfVideoEncoderInstancesRequest message for a selected video source configuration.
- 6. DUT sends the minimum guaranteed number of video encoder instances.

Test Result:

PASS -

DUT passes all assertions.

FAIL –

DUT did not send GetVideoSourceConfigurationsResponse message.

DUT did not send GetGuaranteedNumberOfVideoEncoderInstancesResponse message.



DUT did not send 'TotalNumber' value.

4.2.3 GET GUARANTEED NUMBER OF VIDEO ENCODER INSTANCES AND GET VIDEO ENCODER CONFIGURATION OPTIONS CONSISTENCY

-30-

Test Label: Get Guaranteed Number of Video Encoder Instances and Get Video Encoder Configuration Options Consistency Verification.

Test Case ID: MEDIA-2-1-7

ONVIF Core Specification Coverage: GetGuaranteedNumberOfVideoEncoderInstances (ONVIF Media Service Specification)

Command Under Test: GetGuaranteedNumberOfVideoEncoderInstances

WSDL Reference: media.wsdl

Test Purpose: To verify Get Guaranteed Number of Video Encoder Instances and Get Video Encoder Configuration Options Consistency. To verify Get Guaranteed Number of Video Encoder Instances and Get Video Encoder Configurations Consistency. To verify Get Guaranteed Number of Video Encoder Instances and MaximumNumberOfProfiles capability Consistency.

Pre-Requisite: Media is supported by DUT. Media Service was received from the DUT.

Test Configuration: ONVIF Client and DUT

Test Sequence:

ONVIF C	lient DU	Т
	GetCapabilitiesRequest (Category = 'Media')	Start DUT
Receive and Validate GetCapabilities Response message	GetCapabilitiesResponse (MaximumNumberOfProfiles)	Send capabilities of media service of the DUT
	 GetServiceCapabilitiesRequest (empty) 	
Receive and Validate GetServiceCapabilities Response message	GetServiceCapabilitiesResponse (MaximumNumberOfProfiles)	Send capabilities of media service of the DUT
	 GetVideoEncoderConfigurations Request 	
Receive and Validate GetVideoEncoder Configurations Response message	GetVideoEncoderConfigurations Response (list of existing VideoEncoderConfigurations)	Send list of existing VideoEncoder Configurations
	GetVideoSourceConfigurations Request	
Receive and Validate GetVideoSource Configurations Response message	GetVideoSourceConfigurations Response (list of existing VideoSourceConfigurations)	Send list of existing VideoSource Configurations
	GetGuaranteedNumberOfVideo EncoderInstancesRequest (Video source ConfigurationToken)	
Receive and Validate GetGuaranteedNumber OfVideoEncoder InstancesResponse message	GetGuaranteedNumberOfVideo EncoderInstancesResponse (guaranteed number of video encoder instances)	Send supported number of video encoder instances



-32-

Test Procedure:

- 1. Start an ONVIF Client.
- 2. Start the DUT.
- 3. If GetCapabilities supported by the DUT, ONVIF Client will invoke GetCapabilitiesRequest message (Category = 'Media') to get MaximumNumberOfProfiles capability. Otherwise skip steps 3-4 and go to the step 5.
- 4. Verify the GetCapabilitiesResponse message from the DUT.
- 5. If GetServices supported by the DUT, ONVIF Client will invoke GetServiceCapabilitiesRequest message to get MaximumNumberOfProfiles capability. Otherwise skip steps 5-6 and go to the step 7.
- 6. Verify the GetServiceCapabilitiesResponse message from the DUT.
- 7. ONVIF Client will invoke GetVideoEncoderConfigurationsRequest message to retrieve all DUT video encoder configurations.
- 8. Verify the GetVideoSourceConfigurationsResponse message from the DUT.
- 9. ONVIF Client will invoke GetVideoSourceConfigurationsRequest message to retrieve all DUT video source configurations.
- 10. Verify the GetVideoSourceConfigurationsResponse message from the DUT.
- 11. ONVIF Client will invoke GetGuaranteedNumberOfVideoEncoderInstancesRequest message (ConfigurationToken = "Token1", where "Token1" is a first video source configuration token from GetVideoSourceConfigurationsResponse message) to retrieve guaranteed number of video encoder instances per first video source configuration.
- 12. Verify the GetGuaranteedNumberOfVideoEncoderInstancesResponse message from the DUT.
- 13. Verify that GetGuaranteedNumberOfVideoEncoderInstancesResponse.TotalNumber is less or equal to total number of VideoEncoderConfigurations.
- 14. If GetCapabilities is supported by the DUT, verify that GetGuaranteedNumberOfVideoEncoderInstancesResponse.TotalNumber is less or equal to total number of



GetCapabilitiesResponse.Media.Extension.ProfileCapabilities.MaximumNumberOfProfiles if specified.

- 15. If GetServices is supported by the DUT, Verify that GetGuaranteedNumberOfVideoEncoderInstancesResponse.TotalNumber is less or equal to total number of GetServiceCapabilitiesResponse.Capabilities.ProfileCapabilities.MaximumNumberOfProfiles if specified.
- 16. Repeat steps 11-15 to retrieve guaranteed number of video encoder instances for all video source configuration
- 17. ONVIF Client will invoke GetVideoEncoderConfigurationOptionsRequest message (no ConfigurationToken, no ProfileToken) to retrieve general video encoder options for the DUT.
- 18. Verify the GetVideoEncoderConfigurationOptionsResponse message from the DUT.
- 19. Verify that GetVideoEncoderConfigurationOptionsResponse contains Options.JPEG, if there is at least one GetGuaranteedNumberOfVideoEncoderInstancesResponse with JPEG element having value greater than 0.
- 20. Verify that GetVideoEncoderConfigurationOptionsResponse contains Options.MPEG4, if there is at least one GetGuaranteedNumberOfVideoEncoderInstancesResponse with MPEG4 element having value greater than 0.
- 21. Verify that GetVideoEncoderConfigurationOptionsResponse contains Options.H264, if there is at least one GetGuaranteedNumberOfVideoEncoderInstancesResponse with H264 element having value greater than 0.
- 22. Verify that GetVideoEncoderConfigurationOptionsResponse does not contain Options.JPEG, if there are no GetGuaranteedNumberOfVideoEncoderInstancesResponses with skipped JPEG element or with JPEG element having value greater than 0 (i.e. all GetGuaranteedNumberOfVideoEncoderInstancesResponses have JPEG element that is equal to 0).
- 23. Verify that GetVideoEncoderConfigurationOptionsResponse does not contain Options.MPEG4, if there are no GetGuaranteedNumberOfVideoEncoderInstancesResponses with skipped MPEG4 element or with MPEG4 element having value greater than 0 (i.e. all GetGuaranteedNumberOfVideoEncoderInstancesResponses have MPEG4 element that is equal to 0).
- 24. Verify that GetVideoEncoderConfigurationOptionsResponse does not contain Options.H264, if there are no GetGuaranteedNumberOfVideoEncoderInstancesResponses with skipped H264 element or with H264 element having value greater than 0 (i.e. all GetGuaranteedNumberOfVideoEncoderInstancesResponses have H264 element that is equal to 0).

Test Result:

PASS –

The DUT passes all assertions.

FAIL –

The DUT did not send valid GetVideoSourceConfigurationsResponse message.

The DUT did not send valid GetVideoEncoderConfigurationsResponse message.

The DUT did not send valid GetVideoEncoderConfigurationOptionsResponse message.

-34-

The DUT returned GetVideoEncoderConfigurationOptionsResponse with Options.JPEG element, if there are no GetGuaranteedNumberOfVideoEncoderInstancesResponse with skipped JPEG element or with JPEG element having value greater than 0.

The DUT returned GetVideoEncoderConfigurationOptionsResponse with Options.MPEG4 element, if there are no GetGuaranteedNumberOfVideoEncoderInstancesResponse with skipped MPEG4 element or with MPEG4 element having value greater than 0.

The DUT returned GetVideoEncoderConfigurationOptionsResponse with Options.H264 element, if there are no GetGuaranteedNumberOfVideoEncoderInstancesResponse with skipped H264 element or with H264 element having value greater than 0.

The DUT returned GetVideoEncoderConfigurationOptionsResponse without Options.JPEG element, if there is at least one GetGuaranteedNumberOfVideoEncoderInstancesResponse with JPEG element having value greater than 0.

The DUT returned GetVideoEncoderConfigurationOptionsResponse without Options.MPEG4 element, if there is at least one GetGuaranteedNumberOfVideoEncoderInstancesResponse with MPEG4 element having value greater than 0.

The DUT returned GetVideoEncoderConfigurationOptionsResponse without Options.H264 element, if there is at least one GetGuaranteedNumberOfVideoEncoderInstancesResponse with H264 element having value greater than 0.

The DUT returned GetGuaranteedNumberOfVideoEncoderInstancesResponse.TotalNumber greater than total number of VideoEncoderConfigurations.

The DUT returned GetGuaranteedNumberOfVideoEncoderInstancesResponse.TotalNumber greater than GetServiceCapabilitiesResponse.Capabilities.ProfileCapabilities.MaximumNumberOfProfil es.

The DUT returned GetGuaranteedNumberOfVideoEncoderInstancesResponse.TotalNumber greater than GetCapabilitiesResponse.Media.Extension.ProfileCapabilities.MaximumNumberOfProfiles.

Note: See Annex A.4 for more details about test checks.

4.2.4 VIDEO SOURCE CONFIGURATION

Test Label: Media Configuration Video Source Configuration

Test Case ID: MEDIA-2-1-8

ONVIF Core Specification Coverage: Get media profiles, GetVideoSources, Get video source configurations, Get video source configuration, Get compatible video source configurations, Get video source configuration options, Modify a video source configuration

Command Under Test: None

WSDL Reference: media.wsdl

Test Purpose: To verify DUT Video Source Configuration Operations

Pre-Requisite: None

Test Configuration: ONVIF Client and DUT

Test Sequence:

	Driving IP-based physical security through global standardization
--	---

ONVIF Client		DUT
	GetProfilesRequest (Empty)	Start DUT
Receive and Validate Media Profiles	GetProfilesResponse (Media Profiles) ◀	Send configured media profiles
	GetVideoSourcesRequest (Empty)	
Receive and Validate Video Sources	GetVideoSourcesResponse (Video Sources)	Send Video Sources of the device
	GetCompatibleVideoSource ConfigurationsRequest (Media Profile Token)	
Receive and Validate Video Source Configurations	GetCompatibleVideoSource ConfigurationsResponse (Video Source Configurations)	Send Video Source Configurations compatible with the received media profile
	 GetVideoSourceConfigurationsRequest (Empty) 	
Receive and Validate Video Source Configurations	GetVideoSourceConfigurationsResponse (Video Source Configurations)	Send all Video Source Configurations
	GetVideoSourceConfigurationOptions Request (Video Source Cfg Token)	
Receive and Validate Video Source Configuration options	GetVideoSourceConfigurationOptions Response (Video Source Cfg Options)	Send compatible options for the Video Source Configuration
	I	I
ONV	IF Client	DUT
--	---	--
Set Video Source Configuration with invalid values	SetVideoSourceConfigurationRequest (Invalid Video Source Cfg, FALSE)	Send SOAP 1.2 fault
	SOAP Fault Message (InvalidArgVal/ConfigModify)	message
Set Video Source Configuration	SetVideoSourceConfigurationRequest (Video Source Cfg, FALSE)	Modify Video Source
	SetVideoSourceConfigurationResponse (Empty)	
	GetVideoSourceConfigurationRequest (Video Source Cfg Token)	Send the modified Video Source Configuration
Receive and Validate Video Source Configuration	GetVideoSourceConfigurationResponse (Video Source Cfg)	

Test Procedure:

- 1. Start an ONVIF Client.
- 2. Start the DUT.
- 3. ONVIF Client invokes GetProfiles Request.
- 4. DUT sends the list of existing media profiles in GetProfiles Response message.
- 5. ONVIF Client invokes GetVideoSources request to retrieve the existing video sources of DUT.
- 6. ONVIF Client verifies the list of video sources sent by DUT.
- 7. ONVIF Client invokes GetCompatibleVideoSourceConfigurations request with one of the received media profile tokens as input argument.



8. DUT sends the list of video source configurations compatible with the received media profile token.

-38-

- 9. ONVIF Client verifies the list of video source configurations sent by DUT.
- 10. ONVIF Client invokes GetVideoSourceConfigurations request to retrieve the list of video source configurations supported by the DUT.
- 11. ONVIF Client verifies the list of video source configurations sent by the DUT.
- 12. ONVIF Client invokes GetVideoSourceConfigurationOptions request with one of the received video source configuration tokens as input argument.
- 13. ONVIF Client sends the range of configurable values supported by it for the received video source configuration token.
- 14. ONVIF Client invokes SetVideoSourceConfiguration request with video source configuration values outside the range specified in the GetVideoSourceConfigurationOptionsResponse and 'ForcePersistence' flag as 'FALSE'.
- 15. DUT send the SOAP 1.2 fault message (InvalidArgVal/ConfigModify).
- 16. ONVIF Client verifies the SOAP fault message sent by DUT.
- 17. ONVIF Client invokes SetVideoSourceConfiguration request with video source configuration values within the range specified in GetVideoSourceConfigurationOptionsResponse and 'ForcePersistence' flag as 'FALSE'.
- 18. DUT modifies the video source configuration and sends the SetVideoSourceConfiguration response indicating success.
- 19. ONVIF Client invokes GetVideoSourceConfiguration request to verify the modified video source configuration.
- 20. DUT sends the modified video source configuration in GetVideoSourceConfiguration response.

Test Result:

PASS –

DUT passes all assertions.

FAIL –

DUT did not send GetProfilesResponse message.

DUT did not send valid GetVideoSourcesResponse message.

DUT did not send valid GetCompatibleVideoSourceConfigurationsResponse message.

DUT did not send valid GetVideoSourceConfigurationsResponse message.

DUT did not send GetVideoSourceConfigurationOptionsResponse message.

DUT did not send the SOAP 1.2 fault message (InvalidArgVal/ConfigModify) for invalid SetVideoSourceConfiguration request.

DUT did not send SetVideoSourceConfigurationResponse message.

DUT did not send GetVideoSourceConfigurationResponse message.

DUT did not modify video source configuration correctly.

Note: See Annex A.5 for Name and Token Parameters Length limitations.

4.2.5 JPEG VIDEO ENCODER CONFIGURATION

Test Label: Media Configuration JPEG Video Encoder Configuration

Test Case ID: MEDIA-2-1-9

ONVIF Core Specification Coverage: Get video encoder configurations, Get video encoder configuration, Get video encoder configuration options, Modify a video encoder configuration.

Command Under Test: None

WSDL Reference: media.wsdl

Test Purpose: To verify DUT JPEG Video Encoder Configurations Setting

Pre-Requisite: None

Test Configuration: ONVIF Client and DUT

ONVIF Client		DUT	
Retrieve video encoder	GetVideoEncoderConfigurationsRequest (Empty)	Start DUT	
computations	GetVideoEncoderConfigurationsResponse (Video Encoder Configurations)	Send all video encoder configurations	
	 GetVideoEncoderConfigurationOptions Request (Video Encoder Cfg token) 		
Receive and validate video encoder configuration options	GetVideoEncoderConfigurationOptions Response (Video Encoder Configuration Options)	Send video encoder configuration options	
Set JPEG video encoder configuration with invalid parameters	SetVideoEncoderConfigurationRequest (Invalid JPEG Cfg values, FALSE)		
	SOAP Fault Message (InvalidArgVal/ConfigModify)	Send SOAP 1.2 fault message	
Set JPEG video	 SetVideoEncoderConfigurationRequest (Highest JPEG Configuration, FALSE) 		
	SetVideoEncoderConfigurationResponse	Modify JPEG video	
	 GetVideoEncoderConfigurationRequest (JPEG Encoder Configuration token) 	encoder configuration	
Receive and Validate modified JPEG Encoder Configuration	GetVideoEncoderConfigurationResponse (JPEG Encoder Configuration)	video encoder configuration	

ONVIF Client		DUT
Set JPEG video encoder configuration	SetVideoEncoderConfigurationRequest (Lowest JPEG Configuration, FALSE)	
	SetVideoEncoderConfigurationResponse	Modify JPEG video
	GetVideoEncoderConfigurationRequest (JPEG Encoder Configuration token)	Send modified JPEG
Receive and Validate modified JPEG Encoder Configuration	GetVideoEncoderConfigurationResponse (JPEG Encoder Configuration)	 video encoder configuration
Set JPEG video encoder configuration	 SetVideoEncoderConfigurationRequest (Median JPEG Configuration, FALSE) 	•
	SetVideoEncoderConfigurationResponse	Modify JPEG video
	GetVideoEncoderConfigurationRequest (JPEG Encoder Configuration token)	Send modified JPEG
Receive and Validate modified JPEG Encoder Configuration	GetVideoEncoderConfigurationResponse (JPEG Encoder Configuration)	video encoder configuration

Test Procedure:

- 1. Start an ONVIF Client.
- 2. Start the DUT.
- 3. ONVIF Client invokes GetVideoEncoderConfigurationsRequest to retrieve the list of video encoder configurations supported by DUT.



4. DUT sends video encoder configurations in the GetVideoEncoderConfigurationsResponse message.

-42-

- 5. ONVIF Client invokes GetVideoEncoderConfigurationOptions Request (Video Encoder Configuration token) to retrieve video encoder configuration options for the specified video encoder configuration.
- 6. DUT sends the range of configurable values for the received video encoder configuration in the GetVideoEncoderConfigurationOptionsResponse message.
- 7. Test steps -5 & 6 have to be repeated for all video encoder configurations until ONVIF Client finds a video encoder configuration with JPEG encoding support.
- 8. ONVIF Client invokes SetVideoEncoderConfiguration request with JPEG Resolution value configuration values outside the range defined in the GetVideoEncoderConfigurationOptionsResponse and 'ForcePersistence' flag as 'FALSE'.
- 9. DUT sends the SOAP 1.2 fault message (InvalidArgVal/ConfigModify)
- 10. ONVIF Client verifies the SOAP fault message sent by DUT.
- 11. ONVIF Client invokes SetVideoEncoderConfiguration request (Encoding = "JPEG", Resolution = Highest resolution based on number of pixels, Quality = QualityRange.Max, FramerateLimit = FrameRateRange.Max, BitrateLimit = "64000", EncodingInterval = EncodingIntervalRange.Min, and force persistence = false). These values will be taken from GetVideoEncoderConfigurationOptionsResponse message.
- 12. DUT modifies JPEG video encoder configuration and responds with SetVideoEncoderConfigurationResponse message indicating success.
- 13. ONVIF Client verifies the JPEG Video Encoder Configuration settings on DUT by invoking GetVideoEncoderConfiguration request.
- 14. DUT sends modified JPEG Video Encoder Configuration in the GetVideoEncoderConfigurationResponse message (Encoding = "JPEG", Resolution = Highest resolution based on number of pixels).
- 15. ONVIF Client invokes SetVideoEncoderConfiguration request (Encoding = "JPEG", Resolution = Lowest resolution based on number of pixels, Quality = QualityRange.Min, FramerateLimit = FrameRateRange.Min, BitrateLimit = "64000", EncodingInterval = EncodingIntervalRange.Max, and force persistence = false). These values will be taken from GetVideoEncoderConfigurationOptionsResponse message.
- 16. DUT modifies JPEG video encoder configuration and responds with SetVideoEncoderConfigurationResponse message indicating success.
- 17. ONVIF Client verifies the JPEG Video Encoder Configuration settings on DUT by invoking GetVideoEncoderConfiguration request.
- DUT sends modified JPEG Video Encoder Configuration in the GetVideoEncoderConfigurationResponse message (Encoding = "JPEG", Resolution = Lowest resolution based on number of pixels).
- 19. ONVIF Client invokes SetVideoEncoderConfiguration request (Encoding = "JPEG", Resolution = Median resolution based on number of pixels, Quality = Median value of QualityRange, FramerateLimit = Median value of FrameRateRange, BitrateLimit = "64000", EncodingInterval = Median value of EncodingIntervalRange, and force persistence = false). These values will be taken from GetVideoEncoderConfigurationOptionsResponse message.



- 20. DUT JPEG with modifies video encoder configuration and responds SetVideoEncoderConfigurationResponse message indicating success.
- 21. ONVIF Client verifies the JPEG Video Encoder Configuration settings on DUT by invoking GetVideoEncoderConfiguration request.
- 22. DUT JPEG sends modified Video Encoder Configuration the in GetVideoEncoderConfigurationResponse message (Encoding = "JPEG", Resolution = Median resolution based on number of pixels).

Test Result:

PASS -

DUT passes all assertions.

FAIL -

DUT did not send GetVideoEncoderConfigurationsResponse message.

DUT did not send GetVideoEncoderConfigurationOptionsResponse message.

DUT doesn't support JPEG encoding.

DUT did not send the SOAP 1.2 fault message (InvalidArgVal/ConfigModify) for invalid SetVideoEncoderConfiguration request.

DUT did not send SetVideoEncoderConfigurationResponse message.

DUT did not send GetVideoEncoderConfigurationResponse message.

The DUT did not modify JPEG Video Encoder Configuration.

Note: See Annex A.5 for Name and Token Parameters Length limitations.

MPEG4 VIDEO ENCODER CONFIGURATION 4.2.6

Test Label: Media Configuration MPEG4 Video Encoder Configuration

Test Case ID: MEDIA-2-1-10

ONVIF Core Specification Coverage: Get video encoder configurations, Get video encoder configuration, Get video encoder configuration options, Modify a video encoder configuration.

Command Under Test: None

WSDL Reference: media.wsdl

Test Purpose: To verify DUT MPEG4 Video Encoder Configurations Setting

Pre-Requisite: MPEG4 is implemented by DUT

Test Configuration: ONVIF Client and DUT

ONVIF Client		DUT
Patriava video	GetVideoEncoderConfigurationsRequest (Empty)	Start DUT
encoder configurations	GetVideoEncoderConfigurations Response (Video Encoder Configurations)	configurations
	GetVideoEncoderConfigurationOptions Request (Video Encoder Cfg token)	
Receive and validate video encoder configuration options	GetVideoEncoderConfigurationOptions Response (Video Encoder Configuration Options)	Send video encoder configuration options
Set MPEG4 video encoder configuration with invalid parameters	SetVideoEncoderConfigurationRequest (Invalid MPEG4 Cfg values, FALSE)	
	SOAP Fault Message (InvalidArgVal/ConfigModify)	Send SOAP fault message
Set MPEG4 video encoder configuration	SetVideoEncoderConfigurationRequest (Highest MPEG4 Configuration, FALSE)	
	SetVideoEncoderConfigurationResponse	e Modify MPEG4 video encoder configuration
	GetVideoEncoderConfigurationRequest (MPEG4 Encoder Configuration token)	Send modified MPEG4
Receive and Validate modified MPEG4 Encoder Configuration	GetVideoEncoderConfigurationResponse (MPEG4 Encoder Configuration)	e configuration

ONVIF Client		DUT
Set MPEG4 video encoder configuration	SetVideoEncoderConfigurationRequest (Lowest MPEG4 Configuration, FALSE)	
	SetVideoEncoderConfigurationResponse	Modify MPEG4 video
	GetVideoEncoderConfigurationRequest (MPEG4 Encoder Configuration token)	Send modified MPEG4
Receive and Validate modified MPEG4 Encoder Configuration	GetVideoEncoderConfigurationResponse (MPEG4 Encoder Configuration)	video encoder configuration
Set MPEG4 video encoder configuration	 SetVideoEncoderConfigurationRequest (Median MPEG4 Configuration, FALSE) 	
	SetVideoEncoderConfigurationResponse	Modify MPEG4 video
	GetVideoEncoderConfigurationRequest (MPEG4 Encoder Configuration token)	Send modified MPEG4
Receive and Validate modified MPEG4 Encoder Configuration	GetVideoEncoderConfigurationResponse (MPEG4 Encoder Configuration)	 video encoder configuration

Test Procedure:

- 1. Start an ONVIF Client.
- 2. Start the DUT.



- 3. ONVIF Client invokes GetVideoEncoderConfigurationsRequest to retrieve the list of video encoder configurations supported by DUT.
- 4. DUT sends video encoder configurations in the GetVideoEncoderConfigurationsResponse message.
- 5. ONVIF Client invokes GetVideoEncoderConfigurationOptions Request (Video Encoder Configuration token) to retrieve video encoder configuration options for the specified video encoder configuration.
- 6. DUT sends the range of configurable values for the received video encoder configuration in the GetVideoEncoderConfigurationOptionsResponse message.
- 7. Test steps -5 & 6 have to be repeated for all video encoder configurations until ONVIF Client finds a video encoder configuration with MPEG4 encoding support.
- 8. ONVIF Client invokes SetVideoEncoderConfiguration request with MPEG4 configuration values outside the range defined in the GetVideoEncoderConfigurationOptionsResponse and 'ForcePersistence' flag as 'FALSE'.
- 9. DUT send the SOAP 1.2 fault message (InvalidArgVal/ConfigModify)
- 10. ONVIF Client verifies the SOAP fault message sent by DUT.
- 11. Client invokes SetVideoEncoderConfiguration request (Encoding = "MPEG4", Mpeg4Profile = "SP", if "SP" is not supported "ASP", Resolution = Highest resolution based on number of pixels, Quality = QualityRange.Max, FramerateLimit = FrameRateRange.Max, BitrateLimit = "64000", EncodingInterval = EncodingIntervalRange.Min, GovLength = GovLengthRange.Min, and force persistence = false). These values will be taken from GetVideoEncoderConfigurationOptionsResponse message.
- 12. DUT modifies MPEG4 video encoder configuration and responds with SetVideoEncoderConfigurationResponse message indicating success.
- 13. ONVIF Client verifies the MPEG4 Video Encoder Configuration settings on DUT by invoking GetVideoEncoderConfiguration request.
- 14. DUT sends modified MPEG4 Video Encoder Configuration in the GetVideoEncoderConfigurationResponse message (Encoding = "MPEG4", Mpeg4Profile = "SP", if "SP" is not supported "ASP", Resolution = Highest resolution based on number of pixels).
- 15. ONVIF Client invokes SetVideoEncoderConfiguration request (Encoding = "MPEG4", Mpeg4Profile = "ASP", if "ASP" is not supported "SP", Resolution = Lowest resolution based on number of pixels, Quality = QualityRange.Min, FramerateLimit = FrameRateRange.Min, BitrateLimit = "64000", EncodingInterval = EncodingIntervalRange.Max, GovLength = GovLengthRange.Max, and force persistence = false). These values will be taken from GetVideoEncoderConfigurationOptionsResponse message.
- 16. DUT modifies MPEG4 video encoder configuration and responds with SetVideoEncoderConfigurationResponse message indicating success.
- 17. ONVIF Client verifies the MPEG4 Video Encoder Configuration settings on DUT by invoking GetVideoEncoderConfiguration request.
- 18. DUT sends modified MPEG4 Video Encoder Configuration in the GetVideoEncoderConfigurationResponse message (Encoding = "MPEG4", Mpeg4Profile = "ASP", if "ASP" is not supported "SP", Resolution = Lowest resolution based on number of pixels).



19. ONVIF Client invokes SetVideoEncoderConfiguration request (Encoding = "MPEG4", Mpeg4Profile = "SP", if "SP" is not supported "ASP", Resolution = Median resolution based on number of pixels, Quality = Median value of QualityRange, FramerateLimit = Median value of FrameRateRange, BitrateLimit = "64000", EncodingInterval = Median value of EncodingIntervalRange, GovLength = Median value of GovLengthRange, and force persistence = false). These values will be taken from GetVideoEncoderConfigurationOptionsResponse message.

-47-

- 20. DUT modifies MPEG4 video encoder configuration and responds with SetVideoEncoderConfigurationResponse message indicating success.
- 21. ONVIF Client verifies the MPEG4 Video Encoder Configuration settings on DUT by invoking GetVideoEncoderConfiguration request.
- 22. DUT sends modified MPEG4 Video Encoder Configuration in the GetVideoEncoderConfigurationResponse message (Encoding = "MPEG4", Mpeg4Profile = "SP", if "SP" is not supported "ASP", Resolution = Median resolution based on number of pixels).

Test Result:

PASS –

DUT passes all assertions.

FAIL –

DUT did not send GetVideoEncoderConfigurationsResponse message.

DUT did not send GetVideoEncoderConfigurationOptionsResponse message.

DUT doesn't support MPEG4 encoding.

DUT did not send the SOAP 1.2 fault message (InvalidArgVal/ConfigModify) for invalid SetVideoEncoderConfiguration request.

DUT did not send SetVideoEncoderConfigurationResponse message.

DUT did not send GetVideoEncoderConfigurationResponse message.

The DUT did not modify MPEG4 Video Encoder Configuration.

Note: See Annex A.5 for Name and Token Parameters Length limitations.

4.2.7 H.264 VIDEO ENCODER CONFIGURATION

Test Label: Media Configuration H.264 Video Encoder Configuration

Test Case ID: MEDIA-2-1-11

ONVIF Core Specification Coverage: Get video encoder configurations, Get video encoder configuration, Get video encoder configuration options, Modify a video encoder configuration.

Command Under Test: None

WSDL Reference: media.wsdl

Test Purpose: To verify DUT H.264 Video Encoder Configurations Setting

Pre-Requisite: H.264 is implemented by DUT Test Configuration: ONVIF Client and DUT Test Sequence:

ONVIF Client		DUT
Retrieve video encoder configurations	GetVideoEncoderConfigurationsRequest (Empty) GetVideoEncoderConfigurations Response (Video Encoder Configurations)	Start DUT Send all video encoder configurations
Receive and validate	GetVideoEncoderConfigurationOptions Request (Video Encoder Cfg token) GetVideoEncoderConfigurationOptions Response (Video Encoder Configuration	Send video encoder
configuration options	Options)	
Set H.264 video encoder configuration with invalid parameters	SetVideoEncoderConfigurationRequest (Invalid H.264 Cfg values, FALSE)	-
	SOAP Fault Message (InvalidArgVal/ConfigModify)	Send SOAP fault message
Set H.264 video encoder configuration	SetVideoEncoderConfigurationRequest (Highest H.264 Configuration, FALSE)	
	SetVideoEncoderConfigurationResponse	Modify H.264 video encoder configuration
	GetVideoEncoderConfigurationRequest (H.264 Encoder Configuration token)	Send modified H.264
Receive and Validate modified H.264 Encoder Configuration	GetVideoEncoderConfigurationResponse (H.264 Encoder Configuration)	configuration

ONVIF	Client	DUT
Set H.264 video encoder configuration	SetVideoEncoderConfigurationRequest (Lowest H.264 Configuration, FALSE)	
	SetVideoEncoderConfigurationResponse	Modify H.264 video
Dessive and Validate	GetVideoEncoderConfigurationRequest (H.264 Encoder Configuration token)	Send modified H.264
modified H.264 Encoder Configuration	GetVideoEncoderConfigurationResponse (H.264 Encoder Configuration)	 video encoder configuration
Set H.264 video encoder configuration	 SetVideoEncoderConfigurationRequest (Median H.264 Configuration, FALSE) 	
	SetVideoEncoderConfigurationResponse	Modify H.264 video encoder configuration
Receive and Validate	GetVideoEncoderConfigurationRequest (H.264 Encoder Configuration token)	Send modified H.264
modified H.264 Encoder Configuration	GetVideoEncoderConfigurationResponse (H.264 Encoder Configuration) ◀	configuration

Test Procedure:

- 1. Start an ONVIF Client.
- 2. Start the DUT.



3. ONVIF Client invokes GetVideoEncoderConfigurationsRequest to retrieve the list of video encoder configurations supported by DUT.

-51-

- 4. DUT sends video encoder configurations in the GetVideoEncoderConfigurationsResponse message.
- 5. ONVIF Client invokes GetVideoEncoderConfigurationOptions Request (Video Encoder Configuration token) to retrieve video encoder configuration options for the specified video encoder configuration.
- 6. DUT sends the range of configurable values for the received video encoder configuration in the GetVideoEncoderConfigurationOptionsResponse message.
- 7. Test steps -5 & 6 have to be repeated for all video encoder configurations until ONVIF Client finds a video encoder configuration with H.264 encoding support
- 8. ONVIF Client invokes SetVideoEncoderConfiguration request with H.264 configuration values outside the range defined in the GetVideoEncoderConfigurationOptionsResponse and 'ForcePersistence' flag as 'FALSE'.
- 9. DUT send the SOAP 1.2 fault message (InvalidArgVal/ConfigModify)
- 10. ONVIF Client verifies the SOAP fault message sent by DUT.
- 11. ONVIF Client invokes SetVideoEncoderConfiguration request (Encoding = "H264", H264Profile = "Baseline", if "Baseline" is not supported "Main", if "Main" is not supported "Extended" and if "Extended" is not supported "High", Resolution = Highest resolution based on number of pixels, Quality = QualityRange.Max, FramerateLimit = FrameRateRange.Max, BitrateLimit = "64000", EncodingInterval = EncodingIntervalRange.Min, GovLength = GovLengthRange.Min, and force persistence = false). These values will be taken from GetVideoEncoderConfigurationOptionsResponse message.
- 12. DUT modifies H.264 video encoder configuration and responds with SetVideoEncoderConfigurationResponse message indicating success.
- 13. ONVIF Client verifies the H.264 Video Encoder Configuration settings on DUT by invoking GetVideoEncoderConfiguration request.
- 14. DUT sends modified H.264 Video Encoder Configuration in the GetVideoEncoderConfigurationResponse message (Encoding = "H264", H264Profile "Baseline", if "Baseline" is not supported "Main", if "Main" is not supported "Extended" and if "Extended" is not supported "High", Resolution = Highest resolution based on number of pixels).
- 15. ONVIF Client invokes SetVideoEncoderConfiguration request (Encoding = "H264", H264Profile = "Main", if "Main" is not supported "Extended", if "Extended" is not supported "High" and if "High" is not supported "Baseline", Resolution = Lowest resolution based on number of pixels, Quality = QualityRange.Min, FramerateLimit = FrameRateRange.Min, BitrateLimit = "64000", EncodingInterval = EncodingIntervalRange.Max, GovLength = GovLengthRange.Max, and force persistence = false). These values will be taken from GetVideoEncoderConfigurationOptionsResponse message.
- 16. DUT modifies H.264 video encoder configuration and responds with SetVideoEncoderConfigurationResponse message indicating success.
- 17. ONVIF Client verifies the H.264 Video Encoder Configuration settings on DUT by invoking GetVideoEncoderConfiguration request.



- Configuration 18. DUT modified H.264 Video sends Encoder in the GetVideoEncoderConfigurationResponse message (Encoding = "H264", H264Profile = "Main", if "Main" is not supported "Extended", if "Extended" is not supported "High" and if "High" is not supported "Baseline", Resolution = Lowest resolution based on number of pixels).
- 19. ONVIF Client invokes SetVideoEncoderConfiguration request (Encoding = "H264", H264Profile = "Extended", if "Extended" is not supported "High", if "High" is not supported "Baseline" and if "Baseline" is not supported "Main", Resolution = Median resolution based on number of pixels, Quality = Median value of QualityRange, FramerateLimit = Median value of FrameRateRange, BitrateLimit = "64000", EncodingInterval = Median value of EncodingIntervalRange, GovLength = Median value of GovLengthRange, and force persistence = false). These values will be taken from GetVideoEncoderConfigurationOptionsResponse message.
- 20. DUT modifies H.264 video encoder configuration and responds with SetVideoEncoderConfigurationResponse message indicating success.
- 21. ONVIF Client verifies the H.264 Video Encoder Configuration settings on DUT by invoking GetVideoEncoderConfiguration request.
- 22. DUT sends modified H.264 Video Encoder Configuration the in GetVideoEncoderConfigurationResponse message (Encoding = "H264", H264Profile "Extended", if "Extended" is not supported "High", if "High" is not supported "Baseline" and if "Baseline" is not supported "Main", Resolution = Median resolution based on number of pixels).

Test Result:

PASS -

DUT passes all assertions.

FAIL -

DUT did not send GetVideoEncoderConfigurationsResponse.

DUT did not send GetVideoEncoderConfigurationOptionsResponse message.

DUT doesn't support H.264 encoding.

DUT did not send the SOAP 1.2 fault message (InvalidArgVal/ConfigModify) for invalid SetVideoEncoderConfiguration request.

DUT did not send GetVideoEncoderConfigurationResponse message.

DUT did not send SetVideoEncoderConfigurationResponse message.

The DUT did not modify H.264 Video Encoder Configuration.

Note: See Annex A.5 for Name and Token Parameters Length limitations.

4.3 Video Source Configuration

4.3.1 VIDEO SOURCE CONFIGURATIONS AND PROFILES CONSISTENCY

Test Label: Media Service DUT GetVideoSourceConfigurations Command and GetProfiles Command Consistency Validation.

-53-

Test Case ID: MEDIA-2-2-1

ONVIF Core Specification Coverage: Get media profiles, Get video source configurations

Command Under Test: GetProfiles, GetVideoSourceConfigurations

WSDL Reference: media.wsdl

Test Propose: To check that GetVideoSourceConfigurations command and GetProfiles command are consistent.

Pre-Requisite: ONVIF Client gets the Media Service entry point by GetCapabilities command.

Test Configuration: ONVIF Client and DUT

Test Sequence:

ONVIF	Client	DUT	
Retrieve list of	GetProfilesRequest	Start DUT	
profiles Receive and validate profiles	GetProfilesResponse (list of existing profiles)	Send list of existing profiles	
Retrieve list of VideoSource	GetVideoSourceConfigurationsReques	st	
Configurations Receive and validate response	GetVideoSourceConfigurationsRespor (list of existing VideoSourceConfigurations) ◀	Send list of existing VideoSource Configurations	

Test Procedure:

- 1. Start an ONVIF Client.
- 2. Start the DUT.
- 3. ONVIF Client invokes GetProfilesRequest message to retrieve list of profiles and their configurations.
- 4. The DUT sends GetProfilesResponse message.



5. ONVIF Client invokes GetVideoSourceConfigurationsRequest message to retrieve list of Video Source Configurations from device.

-54-

- 6. The DUT sends GetVideoSourceConfigurationsResponse message.
- 7. Check that each VideoSourceConfiguration from GetVideoSourceConfigurationsResponse message has unique token.
- 8. Check that each VideoSourceConfigurations from the GetProfilesResponse message are included in the GetVideoSourceConfigurationsResponse message.
- 9. Check that VideoSourceConfiguration parameters are same in the GetProfilesResponse message and in the GetVideoSourceConfigurationsResponse message for each VideoSourceConfiguration.

Test Result:

PASS -

DUT passes all assertions.

FAIL –

The DUT did not send GetProfilesResponse message.

The DUT did not send valid GetProfilesResponse message.

The DUT did not send GetVideoSourceConfigurationsResponse message.

The DUT did not send valid GetVideoSourceConfigurationsResponse message.

The DUT return two or more VideoSourceConfigurations in GetVideoSourceConfigurationsResponse message with the same ConfigurationToken.

The DUT returned the GetProfilesResponse message with VideoSourceConfigurations that were not included in the GetVideoSourceConfigurationsResponse message.

The DUT returned different parameters list and parameters values for the same VideoSourceConfiguration in the GetVideoSourceConfigurationsResponse message and in the GetProfilesResponse message.

4.3.2 VIDEO SOURCE CONFIGURATIONS AND VIDEO SOURCE CONFIGURATION CONSISTENCY

Test Label: Media Service DUT GetVideoSourceConfigurations Command and GetVideoSourceConfiguration Command Consistency Validation.

Test Case ID: MEDIA-2-2-2

ONVIF Core Specification Coverage: Get video source configurations, Get video source configuration

Command Under Test: GetVideoSourceConfigurations, GetVideoSourceConfiguration

WSDL Reference: media.wsdl

Test Propose: To check that GetVideoSourceConfigurations and GetVideoSourceConfiguration are consistent.

Pre-Requisite: ONVIF Client gets the Media Service entry point by GetCapabilities command.

Test Configuration: ONVIF Client and DUT

Test Sequence:

ONVIF Client DUT Start DUT **GetVideoSourceConfigurationsRequest** Retrieve list of VideoSource Send list of Configurations **GetVideoSourceConfigurations** existing Response (list of existing VideoSource Receive and VideoSourceConfigurations) Configurations validate response 4 **GetVideoSourceConfigurationRequest** Retrieve (ConfigurationToken) parameters of video source configuration Send parameters GetVideoSourceConfigurationsResponse of video source (parameters for video source configuration Receive and configuration) validate response

Test Procedure:

- 1. Start an ONVIF Client.
- 2. Start the DUT.
- 3. ONVIF Client invokes GetVideoSourceConfigurationsRequest message to retrieve list of Video Source Configurations from device.
- 4. The DUT sends GetVideoSourceConfigurationsResponse message.
- 5. ONVIF Client invokes GetVideoSourceConfigurationRequest (ConfigurationToken) message to retrieve parameters of Video Source Configuration from device.
- 6. The DUT sends GetVideoSourceConfigurationResponse message.
- 7. Verify that all parameters and their values for video source ConfigurationToken from GetVideoSourceConfigurationsResponse message are the same.
- 8. Repeat steps 5-7 for other VideoSourceConfigurations from the GetVideoSourceConfigurationsResponse message.

Test Result:

PASS –

DUT passes all assertions.

FAIL -

The DUT did not send GetVideoSourceConfigurationsResponse message.

The DUT did not send valid GetVideoSourceConfigurationsResponse message.

The DUT did not send GetVideoSourceConfigurationResponse message.

The DUT did not send valid GetVideoSourceConfigurationResponse message.

The DUT did not send equal parameters for VideoSourceConfiguration in the GetVideoSourceConfigurationResponse message and in the GetVideoSourceConfigurationsResponse message.

4.3.3 VIDEO SOURCE CONFIGURATIONS AND VIDEO SOURCE CONFIGURATION OPTIONS CONSISTENCY

Test Label: Media Service DUT GetVideoSourceConfigurations Command and GetVideoSourceConfigurationOptions Command Consistency Validation.

Test Case ID: MEDIA-2-2-3

ONVIF Core Specification Coverage: Get video source configuration options, Get video source configurations

Command Under Test: GetVideoSourceConfigurations, GetVideoSourceConfigurationOptions

WSDL Reference: media.wsdl

Test Propose: To check that GetVideoSourceConfigurations Command and GetVideoSourceConfigurationOptions Command are consistent.

Pre-Requisite: ONVIF Client gets the Media Service entry point by GetCapabilities command.

Test Configuration: ONVIF Client and DUT

ONVIF Client		UT
Start an ONVIF Client		Start DUT
Retrieve list of	GetvideoSourceConfigurationsRequest	
gurations	GetVideoSourceConfigurationsRespons (list of existing	e Send list of existing VideoSource
Receive and validate response	VideoSourceConfigurations)	Configurations
Retrieve Video Source	GetVideoSourceConfigurationOptions Request (ConfigurationToken)	
options	GetVideoSourceConfigurationOptions	Send Video
Receive and validate response	Response (options)	Sources options

Test Procedure:

- 1. Start an ONVIF Client.
- 2. Start the DUT.
- 3. ONVIF Client invokes GetVideoSourceConfigurationsRequest message to retrieve list of Video Source Configurations from device.
- 4. The DUT sends GetVideoSourceConfigurationsResponse message.
- ONVIF Client invokes GetVideoSourceConfigurationOptionsRequest (ConfigurationToken) message to retrieve Video Source Configuration Options for video source configuration from device.
- 6. The DUT sends GetVideoSourceConfigurationOptionsResponse message.
- 7. Check that parameters for video source configuration are available according to options for video source configuration.
- 8. Repeat steps 5-7 for each VideoSourceConfiguration from the GetVideoSourceConfigurationsResponse message.

Test Result:

PASS -

DUT passes all assertions.

FAIL –

The DUT did not send GetVideoSourceConfigurationsResponse message.

The DUT did not send valid GetVideoSourceConfigurationsResponse message.

The DUT did not send GetVideoSourceConfigurationOptionsResponse message.

The DUT did not send valid GetVideoSourceConfigurationOptionsResponse message.

The DUT did send inconsistent GetVideoSourceConfigurationOptionsResponse message:

Source token of VideoSourceConfiguration does not exist in VideoSourceTokensAvailable list for this VideoSourceConfiguration in GetVideoSourceConfigurationOptionsResponse.

VideoSourceConfiguration.Bounds.x is not between BoundsRange.XRange.Min and BoundsRange.XRange.Max.

VideoSourceConfiguration.Bounds.y is not between BoundsRange.YRange.Min and BoundsRange.YRange.Max.

VideoSourceConfiguration.Bounds.width is not between BoundsRange.WidthRange.Min and BoundsRange.WidthRange.Max.

VideoSourceConfiguration.Bounds.height is not between BoundsRange.HeightRange.Min and BoundsRange.HeightRange.Max.

4.3.4 PROFILES AND VIDEO SOURCE CONFIGURATION OPTIONS CONSISTENCY

Test Label: Media Service DUT GetProfiles Command and GetVideoSourceConfigurationOptions Command Consistency Validation.

Test Case ID: MEDIA-2-2-4

ONVIF Core Specification Coverage: Get video source configuration options, Get media profiles

Command Under Test: GetProfiles, GetVideoSourceConfigurationOptions

WSDL Reference: media.wsdl

Test Propose: To check that GetProfiles command and GetVideoSourceConfigurationOptions command are consistent.

Pre-Requisite: ONVIF Client gets the Media Service entry point by GetCapabilities command.

Test Configuration: ONVIF Client and DUT



-59-

Test Procedure:

- 1. Start an ONVIF Client.
- 2. Start the DUT.
- 3. ONVIF Client invokes GetProfilesRequest message to retrieve list of profiles from device.
- 4. The DUT sends GetProfilesResponse message.
- 5. ONVIF Client invokes GetVideoSourceConfigurationOptionsRequest (ProfileToken, ConfigurationToken) message to retrieve VideoSourceConfigurationOptions for the video source configuration from device.
- 6. The DUT sends GetVideoSourceConfigurationOptionsResponse message.
- 7. Check that all parameters for the video source configuration from the profile are correct according to options for video source configuration.
- 8. Repeat steps 5-7 for other profiles with VideoSourceConfiguration from the GetProfilesResponse message.

Test Result:

PASS –

DUT passes all assertions.

FAIL -

The DUT did not send GetProfilesResponse message.

The DUT did not send valid GetProfilesResponse message.

The DUT did not send GetVideoSourceConfigurationOptionsResponse message.

The DUT did not send valid GetVideoSourceConfigurationOptionsResponse message.

The DUT did send inconsistent GetVideoSourceConfigurationOptionsResponse message:

Source token of VideoSourceConfiguration does not exist in VideoSourceTokensAvailable list for this VideoSourceConfiguration in GetVideoSourceConfigurationOptionsResponse.

VideoSourceConfiguration.Bounds.x is not between BoundsRange.XRange.Min and BoundsRange.XRange.Max.

VideoSourceConfiguration.Bounds.y is not between BoundsRange.YRange.Min and BoundsRange.YRange.Max.

VideoSourceConfiguration.Bounds.width is not between BoundsRange.WidthRange.Min and BoundsRange.WidthRange.Max.

VideoSourceConfiguration.Bounds.height is not between BoundsRange.HeightRange.Min and BoundsRange.HeightRange.Max.

4.3.5 VIDEO SOURCE CONFIGURATIONS AND VIDEO SOURCES CONSISTENCY

Test Label: Media Service DUT GetVideoSourceConfigurations Command and GetVideoSources Command Validation.

Test Case ID: MEDIA-2-2-5

ONVIF Core Specification Coverage: GetVideoSources, Get video source configurations

Command Under Test: GetVideoSourceConfigurations, GetVideoSources

WSDL Reference: media.wsdl

Test Propose: To check that GetVideoSourceConfigurations Command and GetVideoSources Command are consistent.

Pre-Requisite: ONVIF Client gets the Media Service entry point by GetCapabilities command.

Test Configuration: ONVIF Client and DUT



-61-

Test Procedure:

- 1. Start an ONVIF Client.
- 2. Start the DUT.
- 3. ONVIF Client invokes GetVideoSourceConfigurationsRequest message to retrieve list of Video Source Configurations from device.
- 4. The DUT sends GetVideoSourceConfigurationsResponse message.
- 5. ONVIF Client invokes GetVideoSourcesRequest message to retrieve list of Video Sources from device.
- 6. The DUT sends GetVideoSourcesResponse message.
- 7. Check that each VideoSourceConfiguration from GetVideoSourcesResponse message has unique token.
- 8. Check that every VideoSourceConfiguration.SourceToken from the GetVideoSourceConfigurationsResponse message exists in GetVideoSourcesResponse message (VideoSource.token).

Test Result:

PASS -

DUT passes all assertions.

FAIL -

The DUT did not send GetVideoSourceConfigurationsResponse message.

The DUT did not send valid GetVideoSourceConfigurationsResponse message.

The DUT did not send GetVideoSourcesResponse message.

The DUT did not send valid GetVideoSourcesResponse message.

The DUT return two or more VideoSources in GetVideoSourcesResponse message with the same token.

The DUT returned the GetVideoSourceConfigurationsResponse message with VideoSources that were not included in the GetVideoSourceConfigurationsResponse message.

4.3.6 VIDEO SOURCE CONFIGURATION USE COUNT (CURRENT STATE)

Test Label: Media Video Source Configuration Use Count Validation.

Test Case ID: MEDIA-2-2-6

ONVIF Core Specification Coverage: Get media profiles, Get video source configurations, Get video source configuration

Command Under Test: GetProfiles, GetVideoSourceConfigurations, GetVideoSourceConfiguration

WSDL Reference: media.wsdl

Test Propose: To check Video Source Configuration use count.

Pre-Requisite: Media is supported by DUT. ONVIF Client gets the Media Service entry point by GetCapabilities command.

Test Configuration: ONVIF Client and DUT

obal standardization
obal standardization

ONVIF Client		Т
Retrieve list of	GetVideoSourceConfigurationsRequest	Start DUT
VSCs Receive and validate VSCs	GetVideoSourceConfigurationsResponse (list of existing VideoSourceConfigurations)	 Send list of existing VideoSource Configurations
Retrieve list of existing profiles	GetProfileRequest	
Receive and	GetProfilesResponse (list of existing profiles)	Send list of existing profiles
Retrieve VideoSource	GetVideoSourceConfigurationRequest (VideoSourceConfiguration)	
Configuration Receive and validate parameters	GetVideoSourceConfigurationResponse (parameters for VideoSourceConfiguration) ◀	Send parameters of VideoSource Configuration

Test Procedure:

- 1. Start an ONVIF Client.
- 2. Start the DUT.
- 3. ONVIF Client invokes GetVideoSourceConfigurationsRequest message to retrieve list of video source configurations from device.
- 4. The DUT sends GetVideoSourceConfigurationsResponse message.
- 5. ONVIF Client invokes GetProfilesRequest message to retrieve list of profiles and their video source configurations from device.
- 6. The DUT sends GetProfilesResponse message.
- 7. Check the **UseCount = usecount1** value for the first VideoSourceConfiguration (VSC1) in the list on step 3.
- 8. Check that there are not more than **usecount1** profiles with this VideoSourceConfiguration in the list from step 6.
- 9. Check that **UseCount** value in GetProfilesResponse for every occurrence of this VideoSourceConfiguration is **usecount1**.
- ONVIF Client invokes GetVideoSourceConfigurationRequest (ConfigurationToken = the first VideoSourceConfiguration from list on step 4) message to retrieve video source configuration parameters.

- 11. The DUT sends GetVideoSourceConfigurationResponse message.
- 12. Check the UseCount value in GetVideoSourceConfigurationResponse (UseCount = usecount1).
- 13. Repeat steps 7-10 for all other VideoSourceConfigurations from the list on step 4.

-64-

Test Result:

PASS –

DUT passes all assertions.

FAIL –

The DUT did not send GetProfilesResponse message.

The DUT did not send valid GetProfilesResponse message.

The DUT did not send GetVideoSourceConfigurationsResponse message.

The DUT did not send valid GetVideoSourceConfigurationsResponse message.

The DUT did not send GetVideoSourceConfigurationResponse message.

The DUT did not send valid GetVideoSourceConfigurationResponse message.

The DUT sent UseCount value which is less than amount of profiles with VideoSourceConfiguration.

The DUT sent different UseCount values in GetProfilesResponse, GetVideoSourceConfigurationsResponse and GetVideoSourceConfigurationResponse messages.

4.3.7 VIDEO SOURCE CONFIGURATION USE COUNT (ADD SAME VIDEO SOURCE CONFIGURATION TO PROFILE TWICE)

Test Label: Media Service DUT Video Source Configuration Use Count Validation after Adding of Same Video Source Configuration to Profile Twice.

Test Case ID: MEDIA-2-2-12

ONVIF Core Specification Coverage: Get media profiles, Get media profile, Add video source configuration to a profile, Get video source configurations, Get video source configuration.

Command Under Test: GetVideoSourceConfigurations, GetVideoSourceConfiguration, AddVideoSourceConfiguration

WSDL Reference: media.wsdl

Test Propose: To check Video Source Configuration use count after adding same video source configuration to profile twice.

Pre-Requisite: Media is supported by DUT. ONVIF Client gets the Media Service entry point by GetCapabilities command.

Test Configuration: ONVIF Client and DUT

Test Sequence:

ONVIF

-65-

Onvir Driving IP-based physical security through global standardization

ONVIF	Client	DUT
Retrieve list of	GetVideoSourceConfigurationsReques	Start DUT
VSCs Receive and	GetVideoSourceConfigurationsRespor (list of existing VideoSourceConfigurations)	Send list of existing VideoSource Configurations
Create new profile	CreateProfileRequest (ProfileToken = New Profile)	
Receive and validate response	CreateProfileResponse (empty)	Send CreateProfile Response
Add VideoSource Configuration to	AddVideoSourceConfigurationReques (ProfileToken = New Profile, ConfigurationToken = VSC1)	t Cond
profile Receive and	AddVideoSourceConfigurationRespone (empty)	se AddVideoSource Configuration Response
Retrieve parameters of	GetVideoSourceConfigurationRequest (ConfigurationToken = VSC1)	
VSC1 Receive and validate VSC1	GetVideoSourceConfigurationRespons (parameters for VSC1)	Send parameters for VSC1
Add VideoSource Configuration to profile Receive and validate response Retrieve parameters of VSC1	AddVideoSourceConfigurationRequest (ProfileToken = New Profile, ConfigurationToken = VSC1) AddVideoSourceConfigurationRespont (empty) GetVideoSourceConfigurationRequest (ConfigurationToken = VSC1)	se Send AddVideoSource Configuration Response
Add VideoSource Configuration to profile Receive and validate response Retrieve parameters of VSC1	(Profile loken = New Profile, ConfigurationToken = VSC1) AddVideoSourceConfigurationRespons (empty) ◀ GetVideoSourceConfigurationRequest (ConfigurationToken = VSC1)	Send AddVideoSou Configuration Response

Send parameters

Send DeleteProfile

for VSC1

Response

Driving IP-based physical security through global standardization ONVIF Client DUT GetVideoSourceConfigurationResponse (parameters for VSC1)

DeleteProfileRequest (ProfileToken = New Profile)

DeleteProfileResponse (empty)

Delete Profile Receive and validate DeleteProfile Response

Receive and validate VSC1

Test Procedure:

- 1. Start an ONVIF Client.
- 2. Start the DUT.
- 3. ONVIF Client invokes GetVideoSourceConfigurationsRequest message to retrieve list of video source configurations from device.
- 4. The DUT sends GetVideoSourceConfigurationsResponse message.
- 5. ONVIF Client invokes CreateProfileRequest (**Name = NewName, Token = New Profile**) message to create profile.
- 6. If the DUT will generate a SOAP 1.2 fault message (Action/MaxNVTProfiles),

If there is profile with "fixed" attribute value "false" in profiles list (profile1)

ONVIF Client invokes RemoveVideoEncoderConfigurationRequest and RemoveVideoSourceConfigurationRequest (**ProfileToken = profile1**) message to remove VideoEncoderConfiguration and VideoSourceConfiguration from profile1

The DUT sends RemoveVideoEncoderConfigurationResponse and RemoveVideoSourceConfigurationResponse message. The UseCount of value of the VideoSourceConfiguration is reduced by 1.

Execute steps 8-17

If there is no profile with "fixed" attribute value "false" in profiles list, end test.

- 7. If the DUT sends CreateProfileResponse message, validate CreateProfileResponse message from the DUT.
- 8. ONVIF Client invokes AddVideoSourceConfigurationRequest (ConfigurationToken = first video source from the list on step 4 (VSC1), ProfileToken = New Profile) message to add VideoSourceConfiguration to the new profile.
- 9. The DUT sends AddVideoSourceConfigurationResponse message.
- 10. ONVIF Client invokes GetVideoSourceConfigurationRequest (**ConfigurationToken = VSC1**) message to retrieve video source configuration parameters.



- 11. The DUT sends GetVideoSourceConfigurationResponse message.
- 12. Check the UseCount value in GetVideoSourceConfigurationResponse message (UseCount = usecount1+1, usecount1 is value of UseCount for VSC1 from the list on step 4). If step 6 is executed, the usecount1 could be reduced by 1.
- 13. ONVIF Client invokes AddVideoSourceConfigurationRequest (**ProfileToken = New Profile, ConfigurationToken = VSC1**) message to replace video source configuration in profile.
- 14. The DUT sends AddVideoSourceConfigurationResponse message.
- 15. ONVIF Client invokes GetVideoSourceConfigurationRequest (**ConfigurationToken = VSC1**) message to retrieve video source configuration parameters.
- 16. The DUT sends GetVideoSourceConfigurationResponse message.
- 17. Check that **UseCount = usecount1+1**, in GetVideoSourceConfigurationResponse message.
- 18. ONVIF Client invokes DeleteProfileRequest (**ProfileToken = New Profile**) message to remove profile with video source configuration.
- 19. The DUT sends DeleteProfileResponse message.

Test Result:

PASS –

DUT passes all assertions.

FAIL –

The DUT did not send GetVideoSourceConfigurationsResponse message.

The DUT did not send valid GetVideoSourceConfigurationsResponse message.

The DUT did not send AddVideoSourceConfigurationResponse message.

The DUT did not send valid AddVideoSourceConfigurationResponse message.

The DUT did not send GetVideoSourceConfigurationResponse message.

The DUT did not send valid GetVideoSourceConfigurationResponse message.

UseCount value is not increased by 1 after adding of the VideoSourceConfiguration to one more profile.

UseCount value is changed 1 after repeated adding of the VideoSourceConfiguration in profile.

Note: See Annex A.5 for Name and Token Parameters Length limitations.

4.3.8 VIDEO SOURCE CONFIGURATION USE COUNT (ADD DIFFERENT VIDEO SOURCE CONFIGURATIONS IN PROFILE)

Test Label: Media Service DUT Video Source Configuration Use Count Validation after Adding Different Video Source Configurations to the Profile.

Test Case ID: MEDIA-2-2-13



ONVIF Core Specification Coverage: Get media profiles, Get media profile, Add video source configuration to a profile, Get video source configurations, Get video source configuration.

-68-

Command Under Test: GetVideoSourceConfigurations, GetVideoSourceConfiguration, AddVideoSourceConfiguration

WSDL Reference: media.wsdl

Test Propose: To check Video Source Configuration use count after adding different video source configurations to profile.

Pre-Requisite: Media is supported by DUT. There are at least two VideoSourceConfigurations. ONVIF Client gets the Media Service entry point by GetCapabilities command. List of media profiles is received by GetProfiles command.

Test Configuration: ONVIF Client and DUT

ONVIF Client		DUT
Retrieve list of	GetVideoSourceConfigurationsReque	Start DUT
VSCs Receive and validate VSCs	GetVideoSourceConfigurationsRespo (list of existing VideoSourceConfigurations) CreateProfileRequest (ProfileToken = New Profile)	Send list of existing VideoSource Configurations
Create new profile Receive and	CreateProfileResponse (empty)	Send CreateProfile Response
Add VideoSource Configuration to profile Receive and validate response Retrieve parameters of VSC1 Receive and validate VSC1	AddVideoSourceConfigurationReques (ProfileToken = New Profile, ConfigurationToken = VSC1) AddVideoSourceConfigurationRespor (empty) GetVideoSourceConfigurationReques (ConfigurationToken = VSC1) GetVideoSourceConfigurationRespon (parameters for VSC1)	st Send AddVideoSource Configuration Response t se Send parameters for VSC1
Add VideoSource Configuration to profile Receive and validate response Retrieve parameters of VSC1	AddVideoSourceConfigurationReques (ProfileToken = Profile1, ConfigurationToken = VSC2) AddVideoSourceConfigurationRespor (empty) GetVideoSourceConfigurationReques (ConfigurationToken = VSC1)	st Send AddVideoSource Configuration Response

ONVIF	Client DUT	DUT	
Receive and	GetVideoSourceConfigurationResponse (parameters for VSC1)	Send parameters for VSC1	
Retrieve parameters of	GetVideoSourceConfigurationRequest (ConfigurationToken = VSC2)		
VSC2 Receive and	GetVideoSourceConfigurationResponse (parameters for VSC2)	Send parameters for VSC2	
Validate VSC2	DeleteProfileRequest (ProfileToken = New Profile)		
Delete Profile			
Receive and validate DeleteProfile Response	DeleteProfileResponse (empty) ◀	Send DeleteProfile Response	

Test Procedure:

- 1. Start an ONVIF Client.
- 2. Start the DUT.
- 3. ONVIF Client invokes GetVideoSourceConfigurationsRequest message to retrieve list of video source configurations from device.
- 4. The DUT sends GetVideoSourceConfigurationsResponse message.
- 5. If there is only one VideoSourceConfiguration in the GetVideoSourceConfigurationsResponse message go to the next test.
- 6. ONVIF Client invokes CreateProfileRequest (**Name = NewName, Token = New Profile**) message to create profile.

If the DUT will generate a SOAP 1.2 fault message (Action/MaxNVTProfiles),

If there is profile with "fixed" attribute value "false" in profiles list (profile1)

ONVIF Client invokes RemoveVideoEncoderConfigurationRequest and RemoveVideoSourceConfigurationRequest (**ProfileToken = profile1**) message to remove VideoEncoderConfiguration and VideoSourceConfiguration from profile1.

The DUT sends RemoveVideoEncoderConfigurationResponse and RemoveVideoSourceConfigurationResponse message. The UseCount value of the two configurations will be reduced by 1.

Execute steps 8-17

If there is no profile with "fixed" attribute value "false" in profiles list, end test.



7. If the DUT sends CreateProfileResponse message, validate CreateProfileResponse message from the DUT.

-71-

- 8. ONVIF Client invokes AddVideoSourceConfigurationRequest (ConfigurationToken = first video source from the list on step 4 (VSC1), ProfileToken = New Profile) message to add VideoSourceConfiguration to the new profile.
- 9. The DUT sends AddVideoSourceConfigurationResponse message.
- 10. ONVIF Client invokes GetVideoSourceConfigurationRequest (**ConfigurationToken = VSC1**) message to retrieve video source configuration parameters.
- 11. The DUT sends GetVideoSourceConfigurationResponse message.
- 12. Check the UseCount value in GetVideoSourceConfigurationResponse message. (UseCount = usecount1+1, usecount1 is the value of UseCount for VSC1 from the list on step 4). If test step 6 is executed, the usecount1 could be reduced by 1.
- ONVIF Client invokes AddVideoSourceConfigurationRequest (ProfileToken = New Profile, ConfigurationToken = VSC2, where VSC2 is other VSC from list on step4) message to replace video source configuration in profile.
- 14. The DUT sends AddVideoSourceConfigurationResponse message.
- 15. ONVIF Client invokes GetVideoSourceConfigurationRequest (**ConfigurationToken = VSC1**) message to retrieve video source configuration parameters.
- 16. The DUT sends GetVideoSourceConfigurationResponse message.
- 17. Check that **UseCount = usecount1**, in GetVideoSourceConfigurationResponse.
- 18. ONVIF Client invokes GetVideoSourceConfigurationRequest (**ConfigurationToken = VSC2**) message to retrieve video source configuration parameters.
- 19. The DUT sends GetVideoSourceConfigurationResponse message.
- 20. Check that **UseCount = usecount2+1**, in GetVideoSourceConfigurationResponse where usecount2 is UseCount value for VSC2 in the list on step 4.
- 21. ONVIF Client invokes DeleteProfileRequest (**ProfileToken = New Profile**) message to remove profile with video source configuration.
- 22. The DUT sends DeleteProfileResponse message.

Test Result:

PASS -

DUT passes all assertions.

FAIL –

The DUT did not send GetVideoSourceConfigurationsResponse message.

The DUT did not send valid GetVideoSourceConfigurationsResponse message.

The DUT did not send AddVideoSourceConfigurationResponse message.

The DUT did not send valid AddVideoSourceConfigurationResponse message.

The DUT did not send GetVideoSourceConfigurationResponse message.

The DUT did not send valid GetVideoSourceConfigurationResponse message.

UseCount value is not increased by 1 after adding of the VideoSourceConfiguration to one more profile.

UseCount value is not decreased by 1 after replacing of the VideoSourceConfiguration in profile for removed VideoSourceConfiguration and UseCount value is not increased by 1 after replacing of the VideoSourceConfiguration in profile for new VideoSourceConfiguration.

Note: See Annex A.5 for Name and Token Parameters Length limitations.

4.3.9 VIDEO SOURCE CONFIGURATION USE COUNT (REMOVE VIDEO SOURCE CONFIGURATION)

Test Label: Media Service DUT Video Source Configuration Use Count Validation after Removing of Video Source Configuration in Profile Validation.

Test Case ID: MEDIA-2-2-14

ONVIF Core Specification Coverage: Remove video source configuration from a profile, Get video source configurations, Get video source configuration

Command Under Test: GetProfiles, GetVideoSourceConfigurations, GetVideoSourceConfiguration, RemoveVideoSourceConfiguration

WSDL Reference: media.wsdl

Test Propose: To check Video Source Configuration use count after removing video source configuration to profile.

Pre-Requisite: Media is supported by DUT. ONVIF Client gets the Media Service entry point by GetCapabilities command. List of media profile is received by GetProfiles command.

Test Configuration: ONVIF Client and DUT
ONVIF	Client	DUT
Retrieve list of	GetVideoSourceConfigurationsRequest	Start DUT
VSCs	GetVideoSourceConfigurationsResponse (list of existing VideoSourceConfigurations)	Send list of existing VideoSource
Receive and validate VSCs	CreateProfileRequest (ProfileToken = New Profile)	
Create new profile	,	-
Receive and	CreateProfileResponse (empty)	Create profiles
Add VideoSource	AddVideoSourceConfigurationRequest (ProfileToken = New Profile, ConfigurationToken = VSC1)	
Configuration to profile	AddVideoSourceConfigurationResponse	Add video source configuration to
Receive and validate response	RemoveVideoSourceConfiguration	
Remove VSC from	Request (Profile loken = New Profile)	→ Remove video
Receive and	RemoveVideoSourceConfiguration Response (empty)	source configuration from profile
validate response Retrieve parameters of	GetVideoSourceConfigurationRequest (ConfigurationToken = VSC1)	
VSC1 Receive and	GetVideoSourceConfigurationResponse (parameters for VSC1)	Send parameters for VSC1
validate VSC1	 DeleteProfileRequest (ProfileToken = New Profile) 	
Delete Profile		→
Receive and validate response	DeleteProfileResponse (empty)	Delete profile

Test Procedure:

1. Start an ONVIF Client.

- 2. Start the DUT.
- 3. ONVIF Client invokes GetVideoSourceConfigurationsRequest message to retrieve list of video source configurations from device.

-74-

- 4. The DUT sends GetVideoSourceConfigurationsResponse message.
- 5. ONVIF Client invokes CreateProfileRequest (**Name = NewName, Token = New Profile**) message to create profile.
- 6. If the DUT will generate a SOAP 1.2 fault message (Action/MaxNVTProfiles),

If there is profile with "fixed" attribute value "false" in profiles list (**profile1**)

ONVIF Client invokes RemoveVideoEncoderConfigurationRequest and RemoveVideoSourceConfigurationRequest (**ProfileToken = profile1**) message to remove VideoEncoderConfiguration and VideoSourceConfiguration from profile1

The DUT sends RemoveVideoEncoderConfigurationResponse and RemoveVideoSourceConfigurationResponse message. The UseCount value of the two configurations will be reduced by 1.

Execute steps 8-13

If there is no profile with "fixed" attribute value "false" in profiles list, end test.

- 7. If the DUT sends CreateProfileResponse message, validate CreateProfileResponse message from the DUT.
- 8. ONVIF Client invokes AddVideoSourceConfigurationRequest (ConfigurationToken = first video source from the list on step 4 (VSC1), ProfileToken = New Profile) message to add VideoSourceConfiguration to the new profile.
- 9. The DUT sends AddVideoSourceConfigurationResponse message.
- 10. ONVIF Client invokes RemoveVideoSourceConfigurationRequest (**ProfileToken = New Profile**) message to remove video source configuration from NewProfileToken.
- 11. The DUT sends RemoveVideoSourceConfigurationResponse message.
- 12. ONVIF Client invokes GetVideoSourceConfigurationRequest (**ConfigurationToken = VSC1**) message to retrieve video source configuration parameters. The DUT sends GetVideoSourceConfigurationResponse message.
- 13. Check the UseCount value in GetVideoSourceConfigurationResponse. (UseCount = usecount1, usecount1 is the UseCount value for VSC1 in the list on step 4). If test step 6 is executed, the usecount1 value could be reduced by 1.
- 14. ONVIF Client invokes DeleteProfileRequest (**ProfileToken = New Profile**) message to remove profile.
- 15. The DUT sends DeleteProfileResponse message.

Test Result:

PASS –

DUT passes all assertions.

FAIL –



The DUT did not send GetVideoSourceConfigurationsResponse message.

The DUT did not send valid GetVideoSourceConfigurationsResponse message.

The DUT did not send AddVideoSourceConfigurationResponse message.

The DUT did not send valid AddVideoSourceConfigurationResponse message.

The DUT did not send RemoveVideoSourceConfigurationResponse message.

The DUT did not send valid RemoveVideoSourceConfigurationResponse message.

The DUT did not send GetVideoSourceConfigurationResponse message.

The DUT did not send valid GetVideoSourceConfigurationResponse message.

UseCount value is not decreased by 1 after removing of the VideoSourceConfiguration from profile.

Note: See Annex A.5 for Name and Token Parameters Length limitations.

4.3.10 VIDEO SOURCE CONFIGURATION USE COUNT (DELETION PROFILE WITH VIDEO SOURCE CONFIGURATION)

Test Label: Media Service DUT Video Source Configuration Use Count Validation after Deletion of Profile with Video Source Configuration Validation.

Test Case ID: MEDIA-2-2-15

ONVIF Core Specification Coverage: Get video source configurations, Get video source configuration, Delete media profile

Command Under Test: GetVideoSourceConfigurations, GetVideoSourceConfiguration, DeleteProfile.

WSDL Reference: media.wsdl

Test Propose: To check Video Source Configuration use count after deletion of profile with source configuration in it.

Pre-Requisite: Media is supported by DUT. ONVIF Client gets the Media Service entry point by GetCapabilities command. List of profiles is received by GetProfiles.

Test Configuration: ONVIF Client and DUT

ONVIF (Client [DUT
Retrieve list of	GetVideoSourceConfigurationsRequest	Start DUT
Receive and validate VSCs	GetVideoSourceConfigurationsResponse (list of existing VideoSourceConfigurations)	Send list of existing VideoSource Configurations
Create new profile Receive and	CreateProfileRequest (ProfileToken = New Profile)	
validate CreateProfile Response	CreateProfileResponse (empty)	➡ Create profile
Add VideoSource Configuration to profile	 AddVideoSourceConfigurationRequest (ProfileToken = New Profile, ConfigurationToken = VSC1) 	
Receive and validate AddVideoSource Configuration	AddVideoSourceConfigurationResponse (empty)	Add video source configuration to profile
Response Delete Profile	DeleteProfileRequest (ProfileToken = New Profile)	
Receive and validate DeleteProfile	DeleteProfileResponse (empty)	Delete profile
Response	GetVideoSourceConfigurationRequest (ConfigurationToken = VSC1)	
Receive and validate VSC1	GetAudioEncoderConfigurationResponse (parameters for VSC1) ◀	Send parameters for VSC1
	1	1

Test Procedure:

- 1. Start an ONVIF Client.
- 2. Start the DUT.
- 3. ONVIF Client invokes GetVideoSourceConfigurationsRequest message to retrieve list of video source configurations from device.
- $\label{eq:constraint} \textbf{4}. \quad \text{The DUT sends GetVideoSourceConfigurationsResponse message}.$



5. ONVIF Client invokes CreateProfileRequest (Name = NewName, Token = New Profile) message to create profile.

-77-

6. If the DUT will generate a SOAP 1.2 fault message (Action/MaxNVTProfiles),

If there is profile with "fixed" attribute value "false" in profiles list (**profile1**)

ONVIF Client invokes RemoveVideoEncoderConfiguration and RemoveVideoSourceConfigurationRequest (**ProfileToken = profile1**) message to remove VideoEncoderConfiguration and VideoSourceConfiguration from profile1

The DUT sends RemoveVideoEncoderConfigurationResponse and RemoveVideoSourceConfigurationResponse message. The UseCount values for the two configurations will be reduced by 1.

Execute steps 8-14

If there is no profile with "fixed" attribute value "false" in profiles list, end test.

- 7. If the DUT sends CreateProfileResponse message, validate CreateProfileResponse message from the DUT.
- 8. ONVIF Client invokes AddVideoSourceConfigurationRequest (ConfigurationToken = first video source from the list on step 4 (VSC1), ProfileToken = New Profile) message to add VideoSourceConfiguration to the new profile.
- 9. The DUT sends AddVideoSourceConfigurationResponse message.
- 10. ONVIF Client invokes DeleteProfileRequest (**ProfileToken = New Profile**) message to remove profile with video source configuration.
- 11. The DUT sends DeleteProfileResponse message.
- 12. ONVIF Client invokes GetVideoSourceConfigurationRequest (**ConfigurationToken = VSC1**) message to retrieve video source configuration parameters.
- 13. The DUT sends GetVideoSourceConfigurationResponse message.
- 14. Check the UseCount value in GetVideoSourceConfigurationResponse message (UseCount = usecount1, usecount1 is the UseCount value for VSC1 in the list on step 4). If test step 6 is executed, the UseCount value could be reduced by 1.

Test Result:

PASS –

DUT passes all assertions.

FAIL –

The DUT did not send GetVideoSourceConfigurationsResponse message.

The DUT did not send valid GetVideoSourceConfigurationsResponse message.

The DUT did not send AddVideoSourceConfigurationResponse message.

The DUT did not send valid AddVideoSourceConfigurationResponse message.

The DUT did not send RemoveVideoSourceConfigurationResponse message.



The DUT did not send valid RemoveVideoSourceConfigurationResponse message.

The DUT did not send GetVideoSourceConfigurationResponse message.

The DUT did not send valid GetVideoSourceConfigurationResponse message.

UseCount value is not decreased by 1 after deletion of the profile with the VideoSourceConfiguration.

Note: See Annex A.5 for Name and Token Parameters Length limitations.

4.3.11 VIDEO SOURCE CONFIGURATION USE COUNT (SET VIDEO SOURCE CONFIGURATION)

Test Label: Media Service DUT Video Source Configuration Use Count Validation after Changing of Video Source Configuration Validation.

Test Case ID: MEDIA-2-2-16

ONVIF Core Specification Coverage: Get video source configurations, Get video source configuration, Modify a video source configuration.

Command Under Test: GetVideoSourceConfigurations, GetVideoSourceConfiguration, SetVideoSourceConfiguration

WSDL Reference: media.wsdl

Test Propose: To check Video Source Configuration use count after setting video source configuration parameters.

Pre-Requisite: Media is supported by DUT. ONVIF Client gets the Media Service entry point by GetCapabilities command.

Test Configuration: ONVIF Client and DUT

UNVIF

Driving IP-based	physical security	through global	standardization

ONVIF Client		DUT	
Retrieve list of	GetVideoSourceConfigurationsRequest	Start DUT	
Receive and	GetVideoSourceConfigurationsResponse (list of existing VideoSourceConfigurations)	Send list of existing VideoSource Configurations	
Set new UseCount	 SetVideoSourceConfigurationRequest (VideoSourceConfiguration = AEC1, UseCount = NewUseCount) 		
Receive and	SetVideoSourceConfigurationResponse (empty)	Set video source configuration	
Retrieve parameters of	GetVideoSourceConfigurationRequest (ConfigurationToken = VSC1)		
VSC1 Receive and validate VSC1	GetVideoSourceConfigurationResponse (parameters for VSC1)	Send parameters for VSC1	

Test Procedure:

- 1. Start an ONVIF Client.
- 2. Start the DUT.
- 3. ONVIF Client invokes GetVideoSourceConfigurationsRequest message to retrieve list of video source configurations from device.
- 4. The DUT sends GetVideoSourceConfigurationsResponse message.
- 5. ONVIF Client invokes SetVideoSourceConfigurationReguest (token = the first Video Source Configuration from the list on step 4(VSC1), UseCount = NewUseCount) message to set parameters for video source interval profile with video source configuration.
- 6. The DUT sends SetVideoSourceConfigurationResponse message.
- 7. ONVIF Client invokes GetVideoSourceConfigurationRequest (ConfigurationToken = VSC1) message to retrieve video source configuration parameters.
- 8. The DUT sends GetVideoSourceConfigurationResponse message.
- 9. Check that UseCount = usecount1, in GetVideoSourceConfigurationResponse where usecount1 is UseCount value for VSC1 in the list on step 4.

Test Result:

PASS –

DUT passes all assertions.

FAIL -

The DUT did not send GetVideoSourceConfigurationsResponse message.

The DUT did not send valid GetVideoSourceConfigurationsResponse message.

The DUT did not send SetVideoSourceConfigurationResponse message.

The DUT did not send valid SetVideoSourceConfigurationResponse message.

The DUT did not send GetVideoSourceConfigurationResponse message.

The DUT did not send valid GetVideoSourceConfigurationResponse message.

UseCount value changed after trying to set UseCount value.

Note: See Annex A.5 for Name and Token Parameters Length limitations.

4.4 Video Encoder Configuration

4.4.1 VIDEO ENCODER CONFIGURATIONS AND PROFILES CONSISTENCY

Test Label: Media Service DUT GetVideoEncoderConfigurations Command and GetProfiles Command Consistency Validation.

Test Case ID: MEDIA-2-3-1

ONVIF Core Specification Coverage: Get media profiles, Get video encoder configurations

Command Under Test: GetProfiles, GetVideoEncoderConfigurations

WSDL Reference: media.wsdl

Test Propose: To check that GetVideoEncoderConfigurations command and GetProfiles command are consistent.

Pre-Requisite: ONVIF Client gets the Media Service entry point by GetCapabilities command.

Test Configuration: ONVIF Client and DUT



-81-

Test Procedure:

- 1. Start an ONVIF Client.
- 2. Start the DUT.
- 3. ONVIF Client invokes GetProfilesRequest message to retrieve list of profiles and their video encoder configurations.
- 4. The DUT sends GetProfilesResponse message.
- 5. ONVIF Client invokes GetVideoEncoderConfigurationsRequest message to retrieve list of Video Encoder Configurations from device.
- 6. The DUT sends GetVideoEncoderConfigurationsResponse message.
- 7. Check that each VideoEncoderConfiguration from GetVideoEncoderConfigurationsResponse message has unique token.
- 8. Check that all VideoEncoderConfigurations from the GetProfilesResponse message are included in the GetVideoEncoderConfigurationsResponse message.
- 9. Check that VideoEncoderConfiguration parameters are same in GetProfilesResponse message and in GetVideoEncoderConfigurationsResponse message for each VideoEncoderConfiguration.

Test Result:

PASS –

DUT passes all assertions.

FAIL -

The DUT did not send GetProfilesResponse message.

The DUT did not send valid GetProfilesResponse message.

The DUT did not send GetVideoEncoderConfigurationsResponse message.

The DUT did not send valid GetVideoEncoderConfigurationsResponse message.

The DUT return two or more VideoEncoderConfigurations in GetVideoEncoderConfigurationsResponse message with the same ConfigurationToken.

The DUT returned the GetProfilesResponse message with VideoEncoderConfigurations that were not included in the GetVideoEncoderConfigurationsResponse message.

The DUT returned different parameters list and parameters values for the same VideoEncoderConfiguration in the GetVideoEncoderConfigurationsResponse message and in the GetProfilesResponse message.

4.4.2 VIDEO ENCODER CONFIGURATIONS AND VIDEO ENCODER CONFIGURATION CONSISTENCY

Test Label: Media Service DUT GetVideoEncoderConfigurations Command and GetVideoEncoderConfiguration Command Consistency Validation.

Test Case ID: MEDIA-2-3-2

ONVIF Core Specification Coverage: Get media profiles, Get video encoder configurations

Command Under Test: GetVideoEncoderConfiguration, GetVideoEncoderConfigurations

WSDL Reference: media.wsdl

Test Propose: To check that GetVideoEncoderConfigurations command and GetVideoEncoderConfiguration command are consistent.

Pre-Requisite: ONVIF Client gets the Media Service entry point by GetCapabilities command.

Test Configuration: ONVIF Client and DUT



-83-

Test Procedure:

- 1. Start an ONVIF Client.
- 2. Start the DUT.
- 3. ONVIF Client invokes GetVideoEncoderConfigurationsRequest message to retrieve list of Video Encoder Configurations from device.
- 4. The DUT sends GetVideoEncoderConfigurationsResponse message.
- 5. ONVIF Client invokes GetVideoEncoderConfigurationRequest (**ConfigurationToken**) message to retrieve parameters of video encoder configuration from device.
- 6. The DUT sends GetVideoEncoderConfigurationResponse message.
- 7. Check that all parameters for video encoder configuration in the GetVideoEncoderConfigurationsResponse message are the same.
- 8. Repeat steps 5-7 for each Video Encoder Configurations from the GetVideoEncoderConfigurationsResponse message.

Test Result:

PASS -

DUT passes all assertions.

FAIL –

The DUT did not send GetVideoEncoderConfigurationsResponse message.

The DUT did not send valid GetVideoEncoderConfigurationsResponse message.



The DUT did not send GetVideoEncoderConfigurationResponse message.

The DUT did not send valid GetVideoEncoderConfigurationResponse message.

The DUT did not send equal parameters for one or more VideoEncoderConfiguration in the GetVideoEncoderConfigurationResponse message and in the GetVideoEncoderConfigurationsResponse message.

4.4.3 VIDEO ENCODER CONFIGURATIONS AND VIDEO ENCODER CONFIGURATION OPTIONS CONSISTENCY

Test Label: Media Service DUT GetVideoEncoderConfigurations Command and GetVideoEncoderConfigurationOptions Command Consistency Validation.

Test Case ID: MEDIA-2-3-3

ONVIF Core Specification Coverage: Get video encoder configurations, Get video encoder configuration options

Command Under Test: GetVideoEncoderConfigurations, GetVideoEncoderConfigurationOptions

WSDL Reference: media.wsdl

Test Propose: To check that GetVideoEncoderConfigurations command and GetVideoEncoderConfigurationOptions command are consistent.

Pre-Requisite: ONVIF Client gets the Media Service entry point by GetCapabilities command.

Test Configuration: ONVIF Client and DUT



Test Procedure:

- 1. Start an ONVIF Client.
- 2. Start the DUT.
- 3. ONVIF Client invokes GetVideoEncoderConfigurationsRequest message to retrieve list of Video Encoder Configurations from device.
- 4. The DUT sends GetVideoEncoderConfigurationsResponse message.
- 5. ONVIF Client invokes GetVideoEncoderConfigurationOptionsRequest (**ConfigurationToken**) message to retrieve VideoEncoderConfigurationOptions from device.
- 6. The DUT sends GetVideoEncoderConfigurationOptionsResponse message.
- 7. Check that all parameters of video encoder configuration are available according to options for video encoder configuration.
- 8. Repeat steps 5-7 for other video encoder configurations from the GetVideoEncoderConfigurationsResponse message.

Test Result:

PASS –

DUT passes all assertions.

FAIL –

The DUT did not send GetVideoEncoderConfigurationsResponse message.

The DUT did not send valid GetVideoEncoderConfigurationsResponse message.

The DUT did not send GetVideoEncoderConfigurationOptionsResponse message.

The DUT did not send valid GetVideoEncoderConfigurationOptionsResponse message.

The DUT sent for at list one VideoEncoderConfiguration with inconsistent parameters:

Encoding value from video encoder configuration is not included in options

There is no pair "Resolution.Height, Resolution.Width" from video encoder configuration in AvailableResolution list for previously checked Encoder from options.

Quality value for video encoder configuration is not between QualityRange.Min and QualityRange.Max from options.

RateControl.FrameRateLimit for video encoder configuration is not between Encoding.FrameRateRange.Min and Encoding.FrameRateRange.Max for current encoding from options

RateControl.EncodingInterval for video encoder configuration is not between EncodingIntervalRange.Min and EncodingIntervalRange.Max for current encoding from options.

RateControl.BitrateLimit for video encoder configuration is not between BitrateRange.Min and BitrateRange.Max for current encoding from options.



If Encoding is H.264, H264.H264Profile value does not exist in H264.H264ProfilesSupported list from options.

If Encoding is MPEG4, MPEG4.MPEG4Profile value does not exist in MPEG4.MPEG4ProfilesSupported list from options.

If Encoding is H264 or MPEG4, Enc.GovLength value is not between Enc.GovLengthRange.Min and Enc.GovLengthRange.Max for corresponded encoding from options.

4.4.4 PROFILES AND VIDEO ENCODER CONFIGURATION OPTIONS CONSISTENCY

Test Label: Media Service DUT GetProfiles Command and GetVideoEncoderConfigurationOptions Command Consistency Validation.

Test Case ID: MEDIA-2-3-4

ONVIF Core Specification Coverage: Get media profile, Get video encoder configuration options

Command Under Test: GetProfiles, GetVideoEncoderConfigurationOptions

WSDL Reference: media.wsdl

Test Propose: To check that GetProfiles command and GetVideoEncoderConfigurationOptions command are consistent.

Pre-Requisite: ONVIF Client gets the Media Service entry point by GetCapabilities command.

Test Configuration: ONVIF Client and DUT

Test Sequence:



Test Procedure:

- 1. Start an ONVIF Client.
- 2. Start the DUT.
- 3. ONVIF Client invokes GetProfilesRequest message to retrieve list of profiles from device.
- 4. The DUT sends GetProfilesResponse message.
- 5. ONVIF Client invokes GetVideoEncoderConfigurationOptionsRequest (**ProfileToken**, **ConfigurationToken**) message to retrieve VideoEncoderConfigurationOptions for video encoder configuration from the DUT.
- 6. The DUT sends GetVideoEncoderConfigurationOptionsResponse message.
- 7. Check that all parameters of video encoder configuration in profile are available according to options for video encoder configuration.
- 8. Repeat steps 5 -7 for other profiles from the GetProfilesResponse message.

Test Result:

PASS –

DUT passes all assertions.

FAIL –

The DUT did not send GetProfilesResponse message.

The DUT did not send valid GetProfilesResponse message.

The DUT did not send GetVideoEncoderConfigurationOptionsResponse message.

The DUT did not send valid GetVideoEncoderConfigurationOptionsResponse message.

The DUT sent for at list one VideoEncoderConfiguration with inconsistent parameters:

Encoding value from video encoder configuration is not included in options

There is no pair "Resolution.Height, Resolution.Width" from video encoder configuration in AvailableResolution list for previously checked Encoder from options.

Quality value for video encoder configuration is not between QualityRange.Min and QualityRange.Max from options.

RateControl.FrameRateLimit for video encoder configuration is not between Encoding.FrameRateRange.Min and Encoding.FrameRateRange.Max for current encoding from options

RateControl.EncodingInterval for video encoder configuration is not between EncodingIntervalRange.Min and EncodingIntervalRange.Max for current encoding from options.

RateControl.BitrateLimit for video encoder configuration is not between BitrateRange.min and BitrateRange.max for current encoding from options.

If Encoding is H.264, H264.H264Profile value does not exist in H264.H264ProfilesSupported list from options.



If Encoding is MPEG4, MPEG4.MPEG4Profile value does not exist in MPEG4.MPEG4ProfilesSupported list from options.

-88-

If Encoding is H264 or MPEG4, Enc.GovLength value is not between Enc.GovLengthRange.Min and Enc.GovLengthRange.Max for corresponded encoding from options.

4.4.5 VIDEO ENCODER CONFIGURATION USE COUNT (CURRENT STATE)

Test Label: Media Service DUT Video Encoder Configuration Use Count Validation.

Test Case ID: MEDIA-2-3-5

ONVIF Core Specification Coverage: Get media profiles, Get video encoder configurations, Get video encoder configuration

WSDL Reference: media.wsdl

Test Propose: To check Video Encoder Configuration use count.

Pre-Requisite: Media is supported by DUT. Profile with VideoEncoderConfiguration exists. ONVIF Client gets the Media Service entry point by GetCapabilities command.

Test Configuration: ONVIF Client and DUT

	P-based physical security through global standardization	
ONVIF	Client DUT	
Retrieve list of	GetVideoEncoderConfigurationsRequest	Start DUT
VECs Receive and validate VECs	GetVideoSourceConfigurationsResponse (list of existing VideoEncoderConfigurations)	Send list of existing VideoEncoderConf igurations

GetProfileRequest

(parameters for

profiles)

GetProfilesResponse (list of existing

GetVideoEncoderConfigurationRequest

GetVideoEncoderConfigurationResponse

(VideoEncoderConfiguration)

VideoEncoderConfiguration)

Send list of existing profiles

Send parameters of VideoEncoder Configuration

Retrieve list of existing profiles

Receive and validate profiles

Retrieve VideoEncoder Configuration

Receive and validate parameters

Test Procedure:

- 1. Start an ONVIF Client.
- 2. Start the DUT.
- 3. ONVIF Client invokes GetVideoEncoderConfigurationsRequest message to retrieve list of video encoder configurations from device.
- 4. The DUT sends GetVideoEncoderConfigurationsResponse message.
- 5. ONVIF Client invokes GetProfilesRequest message to retrieve list of profiles and their video encoder configurations from device.
- 6. The DUT sends GetProfilesResponse message.
- 7. Check the UseCount = usecount1 value for the first VideoEncoderConfiguration (VEC1) in the list on step 4.
- 8. Check that there are not more than usecount1 profiles with this VideoEncoderConfiguration in the list from step 6.
- 9. Check that UseCount value in GetProfilesResponse for every occurrence of this VideoEncoderConfiguration is usecount1.

- 10. ONVIF Client invokes GetVideoEncoderConfigurationRequest (**ConfigurationToken = VEC1**) message to retrieve video encoder configuration parameters.
- 11. The DUT sends GetVideoEncoderConfigurationResponse message.
- Check the UseCount value in GetVideoEncoderConfigurationResponse (UseCount = usecount1).
- 13. Repeat steps 7-10 for all other VideoEncoderConfigurations from the list on step 4.

Test Result:

PASS –

DUT passes all assertions.

FAIL –

The DUT did not send GetProfilesResponse message.

The DUT did not send valid GetProfilesResponse message.

The DUT did not send GetVideoEncoderConfigurationsResponse message.

The DUT did not send valid GetVideoEncoderConfigurationsResponse message.

The DUT did not send GetVideoEncoderConfigurationResponse message.

The DUT did not send valid GetVideoEncoderConfigurationResponse message.

The DUT sent UseCount value which is less than amount of profiles with VideoEncoderConfiguration.

The DUT sent different UseCount values in GetProfilesResponse, GetVideoEncoderConfigurationsResponse and GetVideoEncoderConfigurationResponse messages.

4.4.6 VIDEO ENCODER CONFIGURATIONS – ALL SUPPORTED VIDEO ENCODINGS

Test Label: Set All Supported Encodings Verification.

Test Case ID: DRAFT-MEDIA-2-3-12

ONVIF Core Specification Coverage: None

Command Under Test: None

WSDL Reference: media.wsdl

Test Purpose: To verify whether all supported encodings can be set.

Pre-Requisite: Media is supported by DUT. Media Service was received from the DUT.

Test Configuration: ONVIF Client and DUT

ONVIF Client		DUT	
	GetVideoEncoderConfigurations Request		Start DUT
Receive and Validate GetVideoEncoder	GetVideoEncoderConfigurations Response (list of existing VideoEncoderConfigurations)		Send list of existing VideoEncoder Configurations
Response message	GetVideoEncoderConfiguration OptionsRequest (video encoder configuration token)		
Receive and Validate GetVideoEncoder ConfigurationOptions	GetVideoEncoderConfiguration OptionsResponse (Options)		Send options for VideoEncoder Configuration
Response message	SetVideoEncoderConfiguration Request (Configuration parameters)	_	
Receive and Validate SetVideoEncoder ConfigurationResponse message	SetVideoEncoderConfiguration Response		Modify video encoder configuration
	GetVideoEncoderConfigurations Request (video encoder configuration token)		
Receive and Validate GetVideoEncoder Configurations Response message	GetVideoEncoderConfigurations Response (video encoder configuration) ◀		Send modified video encoder configuration

Test Procedure:

- 1. Start an ONVIF Client.
- 2. Start the DUT.
- 3. ONVIF Client will invoke GetVideoEncoderConfigurationsRequest message to retrieve all DUT video encoder configurations.
- 4. Verify the GetVideoEncoderConfigurationsResponse message from the DUT.
- 5. ONVIF Client will invoke GetVideoEncoderConfigurationOptionsRequest message (ConfigurationToken = "Token1", where "Token1" is a first video encoder configuration token

from GetVideoEncoderConfigurationsResponse message, no ProfileToken) to retrieve supported video encoder configuration options.

6. Verify the GetVideoEncoderConfigurationOptionsResponse message from the DUT.

-92-

- 7. ONVIF SetVideoEncoderConfigurationReguest Client will invoke message (Configuration.token = "Token1", where "Token1" is a first video encoder configuration token GetVideoEncoderConfigurationsResponse message, Configuration.Encoding from "codec1", supported "codec1" where is а first codec from GetVideoEncoderConfigurationOptionsResponse message, other Configuration parameters that are applicable for selected Encoding) to change video encoder configuration settings.
- 8. Verify the SetVideoEncoderConfigurationResponse message from the DUT.
- ONVIF Client will invoke GetVideoEncoderConfigurationRequest message (ConfigurationToken = "Token1", where "Token1" is a first video encoder configuration token from GetVideoEncoderConfigurationsResponse message) to retrieve DUT video encoder configuration for specified token.
- 10. Verify the GetVideoEncoderConfigurationResponse message from the DUT. Check that video encoder setting was applied.
- 11. Repeat steps 7-10 for the rest Video Encodings supported by selected configuration.
- 12. If number of Video encoder configurations in GetVideoEncoderConfigurationsResponse message is more than one, repeat steps 5-11 for the last video encoder configuration token from GetVideoEncoderConfigurationsResponse message.
- If number of Video encoder configurations in GetVideoEncoderConfigurationsResponse message is more than two, Repeat steps 5-11 for the video encoder configuration token between the first and the last VEC tokens from GetVideoEncoderConfigurationsResponse message.

Test Result:

PASS –

The DUT passes all assertions.

FAIL –

The DUT did not send valid GetVideoEncoderConfigurationsResponse message.

The DUT did not send valid GetVideoEncoderConfigurationOptionsResponse message.

The DUT did not send valid SetVideoEncoderConfigurationResponse message.

The DUT sent GetVideoEncoderConfigurationResponse message with Configuration.Encoding other then was set up with SetVideoEncoderConfigurationRequest message.

The DUT did not send valid GetVideoEncoderConfigurationResponse message.

Note: See Annex A.5 for Name and Token Parameters Length limitations.



4.4.7 VIDEO ENCODER CONFIGURATION USE COUNT (ADD SAME VIDEO ENCODER CONFIGURATION TO PROFILE TWICE)

Test Label: Media Service DUT Video Encoder Configuration Use Count Validation after Adding of Same Video Encoder Configuration to Profile Twice.

-93-

Test Case ID: MEDIA-2-3-13

ONVIF Core Specification Coverage: Add video encoder configuration to a profile, Get video encoder configurations, Get video encoder configuration.

Command Under Test: GetVideoEncoderConfigurations, GetVideoEncoderConfiguration, AddVideoEncoderConfiguration

WSDL Reference: media.wsdl

Test Propose: To check Video Encoder Configuration use count after adding same video encoder configuration to profile twice.

Pre-Requisite: Media is supported by DUT. ONVIF Client gets the Media Service entry point by GetCapabilities command. List of profiles is received by GetProfiles command

Test Configuration: ONVIF Client and DUT

DUT

	GetVideoEncoderConfigurationsRequest	Start DUT
Retrieve list of VECs	GetVideoEncoderConfigurations Response (list of existing	Send list of existing VideoEncoder
Receive and validate VECs	VideoEncoderConfigurations)	Configurations
Retrieve list of VSCs	GetVideoSourceConfigurationsRequest	Send list of
Receive and	GetVideoSourceConfigurationsResponse (list of existing VideoSourceConfigurations)	existing VideoSource Configurations
validate VSCs	CreateProfileRequest (ProfileToken = New Profile)	,
Create new profile		
Receive and validate response	CreateProfileResponse (empty)	Send CreateProfile Response
Get compatible VideoSource Configurations for profile	GetCompatibleVideoSource ConfigurationsRequest (ProfileToken = New Profile)	Send
Receive and validate response	GetCompatibleVideoSource ConfigurationsResponse (empty)	GetCompatible VideoSource Configurations Response
Add VideoSource Configuration to profile	AddVideoSourceConfigurationRequest (ProfileToken = New Profile, ConfigurationToken = VSC1)	Sond
	AddVideoSourceConfigurationResponse (empty)	AddVideoSource Configuration
Receive and validate response	•	Response
Get compatible VideoEncoder Configurations for profile	GetCompatibleVideoEncoder ConfigurationsRequest (ProfileToken = New Profile)	Send
· Receive and validate response	GetCompatibleVideoEncoder ConfigurationsResponse (empty)	GetCompatible VideoEncoder Configurations Response

DUT

Driving IP-based physical security through global standardization

ONVIE Client

Old h		
Add VideoEncoder Configuration to	AddVideoEncoderConfigurationRequest (ProfileToken = New Profile, ConfigurationToken = VEC1)	Send
profile Receive and	AddVideoEncoderConfiguration Response (empty)	AddVideoEncoder Configuration Response
Retrieve parameters of	GetVideoEncoderConfigurationRequest (ConfigurationToken = VEC1)	
VEC1 Receive and validate VEC1	GetVideoEncoderConfigurationResponse (parameters for VEC1)	Send parameters for VEC1
Add VideoEncoder Configuration to	AddVideoEncoderConfigurationRequest (ProfileToken = New Profile, ConfigurationToken = VEC1)	Send AddVideoEncoder
profile	AddVideoEncoderConfigurationResponse (empty)	Response
validate response		
Retrieve parameters of	GetVideoEncoderConfigurationRequest (ConfigurationToken = VEC1)	
Receive and validate VEC1	GetVideoEncoderConfigurationResponse (parameters for VEC1)	Send parameters for VEC1

Test Procedure:

- 1. Start an ONVIF Client.
- 2. Start the DUT.
- 3. ONVIF Client invokes GetVideoEncoderConfigurationsRequest message to retrieve list of video encoder configurations from device.
- 4. The DUT sends GetVideoEncoderConfigurationsResponse message.
- 5. ONVIF Client invokes GetVideoSourceConfigurationsRequest message to retrieve list of video source configurations from device.
- 6. The DUT sends GetVideoSourceConfigurationsResponse message.



- 7. ONVIF Client invokes CreateProfileRequest (Name = NewName, Token = New Profile) message to create profile.
- 8. If the DUT will generate a SOAP 1.2 fault message (Action/MaxNVTProfiles),

If there is profile with "fixed" attribute value "false" in profiles list (profile1)

ONVIF Client invokes RemoveVideoEncoderConfigurationRequest (**ProfileToken = profile1**) message to remove VideoEncoderConfiguration from profile1

The DUT sends RemoveVideoEncoderConfigurationResponse message.

Execute steps 10-26.

If there is no profile with "fixed" attribute value "false" in profiles list, end test.

- 9. If the DUT sends CreateProfileResponse message, validate CreateProfileResponse message from the DUT.
- 10. ONVIF Client invokes GetCompatibleVideoSourceConfigurationsRequest (**ProfileToken = New Profile**) message to get compatible VideoSourceConfigurations for new profile.
- 11. The DUT sends GetCompatibleVideoSourceConfigurationsResponse message.
- 12. ONVIF Client invokes AddVideoSourceConfigurationRequest (ConfigurationToken = first video source configuration from the list on step 11 (VSC1), ProfileToken = New Profile) message to add VideoSourceConfiguration to the new profile.
- 13. The DUT sends AddVideoSourceConfigurationResponse message.
- 14. ONVIF Client invokes GetCompatibleVideoEncoderConfigurationsRequest (ProfileToken = New Profile) message to get compatible VideoEncoderConfigurations for new profile with VideoSourceConfiguration.
- 15. The DUT sends GetCompatibleVideoEncoderConfigurationsResponse message. If response does not contain any VideoEncoderConfiguration, repeat steps 12-15 with next video source configuration from step 11. If there are no other video source configuration, skip other steps and go to the next test.
- 16. ONVIF Client invokes AddVideoEncoderConfigurationRequest (ConfigurationToken = first video encoder configuration from the list on step 15 (VEC1), ProfileToken = New Profile) message to add VideoEncoderConfiguration to the new profile.
- 17. The DUT sends AddVideoEncoderConfigurationResponse message.
- 18. ONVIF Client invokes GetVideoEncoderConfigurationRequest (**ConfigurationToken = VEC1**) message to retrieve video encoder configuration parameters.
- 19. The DUT sends GetVideoEncoderConfigurationResponse message.
- 20. Check the UseCount value in GetVideoEncoderConfigurationResponse message (UseCount = usecount1+1, usecount1 is the UseCount value for VEC1 from the list on step 4). If test step 8 is executed, the usecount1 value could be reduced by 1.
- 21. ONVIF Client invokes AddVideoEncoderConfigurationRequest (**ProfileToken = New Profile**, **ConfigurationToken = VEC1**) message to replace video encoder configuration in profile.
- 22. The DUT sends AddVideoEncoderConfigurationResponse message.

23. ONVIF Client invokes GetVideoEncoderConfigurationRequest (**ConfigurationToken = VEC1**) message to retrieve video encoder configuration parameters.

-97-

- 24. The DUT sends GetVideoEncoderConfigurationResponse message.
- 25. Check that **UseCount = usecount1+1**, in GetVideoEncoderConfigurationResponse message.
- 26. ONVIF Client restores DUT configuration.

Test Result:

PASS –

DUT passes all assertions.

FAIL –

The DUT did not send valid GetVideoEncoderConfigurationsResponse message.

 $\label{eq:compatible} The \ \ DUT \ \ did \ \ not \ \ send \ \ valid \ \ \ GetCompatibleVideoEncoderConfigurationsResponse message.$

The DUT did not send valid GetCompatibleVideoSourceConfigurationsResponse message.

The DUT did not send valid AddVideoEncoderConfigurationResponse message.

The DUT did not send valid GetVideoEncoderConfigurationResponse message.

UseCount value is not increased by 1 after adding of the VideoEncoderConfiguration to one more profile.

UseCount value is changed after repeated adding of the VideoEncoderConfiguration in profile.

Note: See Annex A.5 for Name and Token Parameters Length limitations.

4.4.8 VIDEO ENCODER CONFIGURATION USE COUNT (ADD DIFFERENT VIDEO ENCODER CONFIGURATIONS IN PROFILE)

Test Label: Media Service DUT Video Encoder Configuration Use Count Validation after Adding Different Video Encoder Configurations to the Profile.

Test Case ID: MEDIA-2-3-14

ONVIF Core Specification Coverage: Add video encoder configuration to a profile, Get video encoder configurations, Get video encoder configuration.

Command Under Test: GetVideoEncoderConfigurations, GetVideoEncoderConfiguration, AddVideoEncoderConfiguration

WSDL Reference: media.wsdl

Test Propose: To check Video Encoder Configuration use count after adding different video encoder configurations to profile.

Pre-Requisite: Media is supported by DUT. There are at least two VideoEncoderConfigurations. ONVIF Client gets the Media Service entry point by GetCapabilities command. List of profiles is received by GetProfiles command



-98-

Test Configuration: ONVIF Client and DUT

DUT

	GetVideoEncoderConfigurationsRequest	Start DUT
Retrieve list of VECs		Send list of
Receive and	GetVideoEncoderConfigurations Response (list of existing VideoEncoderConfigurations)	existing VideoEncoder Configurations
validate VECs	GetVideoSourceConfigurationsRequest	
VSCs	GetVideoSourceConfigurations Response (list of existing	Send list of existing
Receive and validate VSCs	VideoSourceConfigurations)	Configurations
Create new profile	New Profile)	
Receive and validate response	CreateProfileResponse (empty)	Send CreateProfile Response
Get compatible VideoSource Configurations for profile	GetCompatibleVideoSource ConfigurationsRequest (ProfileToken = New Profile)	Send
Receive and validate response	GetCompatibleVideoSource ConfigurationsResponse (empty)	GetCompatible VideoSource Configurations Response
Add VideoSource Configuration to	AddVideoSourceConfigurationRequest (ProfileToken = New Profile, ConfigurationToken = VSC1)	Send
profile Receive and validate response	AddVideoSourceConfigurationResponse (empty)	AddVideoSource Configuration Response
Get compatible VideoEncoder Configurations for profile	GetCompatibleVideoEncoder ConfigurationsRequest (ProfileToken = New Profile)	Send
Receive and validate response	GetCompatibleVideoEncoder ConfigurationsResponse (empty)	GetCompatible VideoEncoder Configurations Response

-99-

ONVIF Client		DUT	
Add VideoEncoder Configuration to	AddVideoEncoderConfigurationRequest (ProfileToken = New Profile, ConfigurationToken = VEC1)	Send	
profile Receive and	AddVideoEncoderConfigurationResponse (empty)	AddVideoEncoder Configuration Response	
Retrieve parameters of	GetVideoEncoderConfigurationRequest (ConfigurationToken = VEC1)		
VEC1 Retrieve parameters of	GetVideoEncoderConfigurationResponse (parameters for VEC1)	Send parameters for VEC1	
VEC1 Add VideoEncoder Configuration to profile	AddVideoEncoderConfigurationRequest (ProfileToken = Profile1, ConfigurationToken = VEC2) AddVideoEncoderConfigurationResponse	► Send AddVideoEncoder	
Receive and validate response	(empty) ◄ Get\/ideoEncoderConfigurationRequest	Configuration Response	
Retrieve parameters of VEC1	(ConfigurationToken = VEC1)	Send parameters	
Receive and validate VEC1	(parameters for VEC1)	for VEC1	
Retrieve parameters of	GetVideoEncoderConfigurationRequest (ConfigurationToken = VEC2)		
VEC2 Receive and validate VEC2	GetVideoEncoderConfigurationResponse (parameters for VEC2)	Send parameters for VEC2	
		1	

Test Procedure:

- 1. Start an ONVIF Client.
- 2. Start the DUT.
- 3. ONVIF Client invokes GetVideoEncoderConfigurationsRequest message to retrieve list of video encoder configurations from device.
- 4. The DUT sends GetVideoEncoderConfigurationsResponse message.



- 5. If there is only one VideoEncoderConfiguration in the GetVideoEncoderConfigurationsResponse message go to the next test.
- 6. ONVIF Client invokes GetVideoSourceConfigurationsRequest message to retrieve list of video source configurations from device.
- 7. The DUT sends GetVideoSourceConfigurationsResponse message.
- 8. ONVIF Client invokes CreateProfileRequest (Name = New Name, Token = New Profile) message to create profile.

If the DUT will generate a SOAP 1.2 fault message (Action/MaxNVTProfiles),

If there is profile with "fixed" attribute value "false" in profiles list (profile1)

ONVIF Client invokes RemoveVideoEncoderConfigurationRequest (ProfileToken = profile1) message to remove VideoEncoderConfiguration from profile1

The DUT sends RemoveVideoEncoderConfigurationResponse message

Execute steps 10-29

If there is no profile with "fixed" attribute value "false" in profiles list, end test.

- 9. If the DUT sends CreateProfileResponse message, validate CreateProfileResponse message from the DUT.
- 10. ONVIF Client invokes GetCompatibleVideoSourceConfigurationsReguest (ProfileToken = New Profile) message to get compatible VideoSourceConfigurations for new profile.
- 11. The DUT sends GetCompatibleVideoSourceConfigurationsResponse message.
- 12. ONVIF Client invokes AddVideoSourceConfigurationRequest (ConfigurationToken = first video source configuration from the list on step 11 (VSC1), ProfileToken = New Profile) message to add VideoSourceConfiguration to the new profile.
- 13. The DUT sends AddVideoSourceConfigurationResponse message.
- 14. ONVIF Client invokes GetCompatibleVideoEncoderConfigurationsRequest (ProfileToken = New Profile) message to get compatible VideoEncoderConfigurations for new profile with VideoSourceConfiguration.
- 15. The DUT sends GetCompatibleVideoEncoderConfigurationsResponse message. If response contains number of VideoEncoderConfiguration less than 2, repeat steps 12-15 with next video source configuration from step 11. If there are no other video source configuration, skip other steps and go to the next test.
- 16. ONVIF Client invokes AddVideoEncoderConfigurationReguest (ConfigurationToken = first video encoder configuration from the list on step 15 (VEC1), ProfileToken = New Profile) message to add VideoEncoderConfiguration to the new profile.
- 17. The DUT sends AddVideoEncoderConfigurationResponse message.
- 18. ONVIF Client invokes GetVideoEncoderConfigurationRequest (ConfigurationToken = VEC1) message to retrieve video encoder configuration parameters.
- 19. The DUT sends GetVideoEncoderConfigurationResponse message.



- 20. Check the UseCount value in GetVideoEncoderConfigurationResponse (**UseCount = usecount1+1**, usecount1 is value of UseCount for VEC1 from the list on step 4). If step 8 is executed; the usecount1 could be reduced by 1.
- 21. ONVIF Client invokes AddVideoEncoderConfigurationRequest (**ProfileToken = New Profile**, **ConfigurationToken = VEC2**, where VEC2 is other VEC from list on step 15) message to replace video encoder configuration in profile.
- 22. The DUT sends AddVideoEncoderConfigurationResponse message.
- 23. ONVIF Client invokes GetVideoEncoderConfigurationRequest (**ConfigurationToken = VEC1**) message to retrieve video encoder configuration parameters.
- 24. The DUT sends GetVideoEncoderConfigurationResponse message.
- 25. Check that **UseCount = usecount1**, in GetVideoEncoderConfigurationResponse.
- 26. ONVIF Client invokes GetVideoEncoderConfigurationRequest (**ConfigurationToken = VEC2**) message to retrieve video encoder configuration parameters.
- 27. The DUT sends GetVideoEncoderConfigurationResponse message.
- 28. Check that **UseCount = usecount2+1**, in GetVideoEncoderConfigurationResponse where usecount2 is UseCount value for VEC2 in the list on step 4.
- 29. ONVIF Client restores DUT configuration.

Test Result:

PASS –

DUT passes all assertions.

FAIL –

The DUT did not send valid GetVideoEncoderConfigurationsResponse message.

The DUT did not send valid AddVideoEncoderConfigurationResponse message.

The DUT did not send valid GetVideoEncoderConfigurationResponse message.

The DUT did not send valid GetCompatibleVideoEncoderConfigurationsResponse message.

The DUT did not send valid GetCompatibleVideoSourceConfigurationsResponse message.

UseCount value is not increased by 1 after adding of the VideoEncoderConfiguration to one more profile.

UseCount value is not decreased by 1 after replacing of the VideoEncoderConfiguration in profile for removed VideoEncoderConfiguration and UseCount value is not increased by 1 after replacing of the VideoEncoderConfiguration in profile for new VideoEncoderConfiguration.

Note: See Annex A.5 for Name and Token Parameters Length limitations.



4.4.9 VIDEO ENCODER CONFIGURATION USE COUNT (REMOVE VIDEO ENCODER CONFIGURATION)

Test Label: Media Service DUT Video Encoder Configuration Use Count Validation after Removing of Video Encoder Configuration in Profile Validation.

-103-

Test Case ID: MEDIA-2-3-15

ONVIF Core Specification Coverage: Remove video encoder configuration from a profile, Get video encoder configurations, Get video encoder configuration.

Command Under Test: GetVideoEncoderConfigurations, GetVideoEncoderConfiguration, RemoveVideoEncoderConfiguration

WSDL Reference: media.wsdl

Test Propose: To check Video Encoder Configuration use count after removing video encoder configuration from a profile.

Pre-Requisite: Media is supported by DUT. ONVIF Client gets the Media Service entry point by GetCapabilities command. List of profiles is received by GetProfiles command.

Test Configuration: ONVIF Client and DUT

ONVIF Client		Т
Retrieve list of VECs	GetVideoEncoderConfigurationsRequest	Start DUT
	GetVideoEncoderConfigurations Response (list of existing VideoEncoderConfigurations)	Send list of existing VideoEncoder Configurations
Receive and validate VECs		
Retrieve list of VSCs	GetVideoSourceConfigurationsRequest	Send list of
Receive and validate VSCs	Response (list of existing VideoSourceConfigurations)	VideoSource Configurations
Crate new profile	CreateProfileRequest (ProfileToken = New Profile)	
Receive and validate response	CreateProfileResponse (empty)	Create profile
Get compatible VideoSource Configurations for profile	GetCompatibleVideoSource ConfigurationsRequest (ProfileToken = New Profile)	Send
Receive and validate response	GetCompatibleVideoSource ConfigurationsResponse (empty)	GetCompatible VideoSource Configurations Response
Add VideoSource Configuration to profile	AddVideoSourceConfigurationRequest (ProfileToken = New Profile, ConfigurationToken = VSC1)	• • • • • •
Receive and validate response	AddVideoSourceConfigurationResponse (empty)	Add video source configuration to profile
Get compatible VideoEncoder Configurations for profile	GetCompatibleVideoEncoder ConfigurationsRequest (ProfileToken = New Profile)	Send
Receive and validate response	GetCompatibleVideoEncoder ConfigurationsResponse (empty)	VideoEncoder Configurations Response



Test Procedure:

- 1. Start an ONVIF Client.
- 2. Start the DUT.
- 3. ONVIF Client invokes GetVideoEncoderConfigurationsRequest message to retrieve list of video encoder configurations from device.
- 4. The DUT sends GetVideoEncoderConfigurationsResponse message.
- 5. ONVIF Client invokes GetVideoSourceConfigurationsRequest message to retrieve list of video source configurations from device.
- 6. The DUT sends GetVideoSourceConfigurationsResponse message.
- 7. ONVIF Client invokes CreateProfileRequest (Name = NewName, Token = New Profile) message to create profile.
- 8. If the DUT will generate a SOAP 1.2 fault message (Action/MaxNVTProfiles),

If there is profile with "fixed" attribute value "false" in profiles list (profile1)

ONVIF Client invokes RemoveVideoEncoderConfigurationRequest (**ProfileToken = profile1**) message to remove VideoEncoderConfiguration from profile1

The DUT sends RemoveVideoEncoderConfigurationResponse message

Execute steps 10-23

If there is no profile with "fixed" attribute value "false" in profiles list, end test.

9. If the DUT sends CreateProfileResponse message, validate CreateProfileResponse message from the DUT.



- 10. ONVIF Client invokes GetCompatibleVideoSourceConfigurationsRequest (**ProfileToken = New Profile**) message to get compatible VideoSourceConfigurations for new profile.
- 11. The DUT sends GetCompatibleVideoSourceConfigurationsResponse message.
- 12. ONVIF Client invokes AddVideoSourceConfigurationRequest (ConfigurationToken = first video source configuration from the list on step 11 (VSC1), ProfileToken = New Profile) message to add VideoSourceConfiguration to the new profile.
- 13. The DUT sends AddVideoSourceConfigurationResponse message.
- ONVIF Client invokes GetCompatibleVideoEncoderConfigurationsRequest (ProfileToken = New Profile) message to get compatible VideoEncoderConfigurations for new profile with VideoSourceConfiguration.
- 15. The DUT sends GetCompatibleVideoEncoderConfigurationsResponse message. If response does not contain any VideoEncoderConfiguration, repeat steps 12-15 with next video source configuration from step 11. If there are no other video source configuration, skip other steps and go to the next test.
- 16. ONVIF Client invokes AddVideoEncoderConfigurationRequest (ConfigurationToken = first video encoder configuration from the list on step 15 (VEC1), ProfileToken = New Profile) message to add VideoEncoderConfiguration to the new profile.
- 17. The DUT sends AddVideoEncoderConfigurationResponse message.
- 18. ONVIF Client invokes RemoveVideoEncoderConfigurationRequest (**ProfileToken = New Profile**) message to remove video encoder configuration from NewProfileToken.
- 19. The DUT sends RemoveVideoEncoderConfigurationResponse message.
- 20. ONVIF Client invokes GetVideoEncoderConfigurationRequest (**ConfigurationToken = VEC1**) message to retrieve video encoder configuration parameters.
- 21. The DUT sends GetVideoEncoderConfigurationResponse message.
- 22. Check that **UseCount = usecount1**, in GetVideoEncoderConfigurationResponse where usecount1 is UseCount value for VEC1 in the list on step 4.If step 8 is executed, the usecount1 could be reduced by 1.
- 23. ONVIF Client restores DUT configuration.

Test Result:

PASS –

DUT passes all assertions.

FAIL –

The DUT did not send valid GetVideoEncoderConfigurationsResponse message.

The DUT did not send valid AddVideoEncoderConfigurationResponse message.

The DUT did not send valid RemoveVideoEncoderConfigurationResponse message.

The DUT did not send valid GetVideoEncoderConfigurationResponse message.

The DUT did not send valid GetCompatibleVideoEncoderConfigurationsResponse message.



The DUT did not send valid GetCompatibleVideoSourceConfigurationsResponse message.

UseCount value is not decreased by 1 after removing of the VideoEncoderConfiguration from profile.

Note: See Annex A.5 for Name and Token Parameters Length limitations.

4.4.10 VIDEO ENCODER CONFIGURATION USE COUNT (PROFILE DELETION WITH VIDEO ENCODER CONFIGURATION)

Test Label: Media Service DUT Video Encoder Configuration Use Count Validation after Deletion of Profile with Video Encoder Configuration Validation.

Test Case ID: MEDIA-2-3-16

ONVIF Core Specification Coverage: Get video encoder configurations, Get video encoder configuration, Delete media profile

Command Under Test: GetVideoEncoderConfigurations, GetVideoEncoderConfiguration, DeleteProfile

WSDL Reference: media.wsdl

Test Propose: To check Video Encoder Configuration use count after deletion of the profile with video encoder configuration

Pre-Requisite: Media is supported by DUT. ONVIF Client gets the Media Service entry point by GetCapabilities command. List of profiles is received by GetProfiles command.

Test Configuration: ONVIF Client and DUT

ONVIF Client		UT	
Retrieve list of VECs	GetVideoEncoderConfigurationsRequest	:	Start DUT
	GetVideoEncoderConfigurations Response (list of existing VideoEncoderConfigurations)	→	existing VideoEncoder Configurations
Receive and validate VECs			g
Retrieve list of VSCs	GetVideoSourceConfigurationsRequest	→	Send list of
Receive and validate VSCs	GetVideoSourceConfigurations Response (list of existing VideoSourceConfigurations)		VideoSource Configurations
Crate new profile	CreateProfileRequest (ProfileToken = New Profile)		
Receive and validate response	CreateProfileResponse (empty)		Create profile
Get compatible VideoSource Configurations for profile	GetCompatibleVideoSource ConfigurationsRequest (ProfileToken = New Profile)	-	Send
Receive and validate response	GetCompatibleVideoSource ConfigurationsResponse (empty)		GetCompatible VideoSource Configurations Response
Add VideoSource Configuration to profile	AddVideoSourceConfigurationRequest (ProfileToken = New Profile, ConfigurationToken = VSC1)	-	
Receive and validate response	AddVideoSourceConfigurationResponse (empty)		configuration to profile
Get compatible VideoEncoder Configurations for profile	GetCompatibleVideoEncoder ConfigurationsRequest (ProfileToken = New Profile)		Send
Receive and validate response	GetCompatibleVideoEncoder ConfigurationsResponse (empty)		GetCompatible VideoEncoder Configurations Response
ONVIF	Client D	UT	
--------------------------------------	---	--	
Add VideoEncoder Configuration to	AddVideoEncoderConfigurationRequest (ProfileToken = New Profile, ConfigurationToken = VEC1)		
profile Receive and	AddVideoEncoderConfigurationRespons (empty)	e Add video encoder configuration to profile	
Delete Profile	DeleteProfileRequest (ProfileToken = New Profile)		
Receive and	DeleteProfileResponse (empty)	Delete profile	
Retrieve parameters of	GetVideoEncoderConfigurationRequest (ConfigurationToken = VEC1)		
VEC1 Receive and validate VEC1	GetAudioEncoderConfigurationResponse (parameters for VEC1) ◀	e Send parameters for VEC1	

Test Procedure:

- 1. Start an ONVIF Client.
- 2. Start the DUT.
- 3. ONVIF Client invokes GetVideoEncoderConfigurationsRequest message to retrieve list of video encoder configurations from device.
- 4. The DUT sends GetVideoEncoderConfigurationsResponse message.
- 5. ONVIF Client invokes GetVideoSourceConfigurationsRequest message to retrieve list of video source configurations from device.
- 6. The DUT sends GetVideoSourceConfigurationsResponse message.
- 7. ONVIF Client invokes CreateProfileRequest (Name = NewName, Token = New Profile) message to create profile.
- 8. If the DUT will generate a SOAP 1.2 fault message (Action/MaxNVTProfiles),

If there is profile with "fixed" attribute value "false" in profiles list (profile1)

ONVIF Client invokes RemoveVideoEncoderConfigurationRequest (**ProfileToken = profile1**) message to remove VideoEncoderConfiguration from profile1

The DUT sends RemoveVideoEncoderConfigurationResponse message

Execute steps 10-23



If there is no profile with "fixed" attribute value "false" in profiles list, end test.

- 9. If the DUT sends CreateProfileResponse message, validate CreateProfileResponse message from the DUT.
- 10. ONVIF Client invokes GetCompatibleVideoSourceConfigurationsRequest (**ProfileToken = New Profile**) message to get compatible VideoSourceConfigurations for new profile.
- 11. The DUT sends GetCompatibleVideoSourceConfigurationsResponse message.
- 12. ONVIF Client invokes AddVideoSourceConfigurationRequest (ConfigurationToken = first video source configuration from the list on step 11 (VSC1), ProfileToken = New Profile) message to add VideoSourceConfiguration to the new profile.
- 13. The DUT sends AddVideoSourceConfigurationResponse message.
- ONVIF Client invokes GetCompatibleVideoEncoderConfigurationsRequest (ProfileToken = New Profile) message to get compatible VideoEncoderConfigurations for new profile with VideoSourceConfiguration.
- 15. The DUT sends GetCompatibleVideoEncoderConfigurationsResponse message. If response does not contain any VideoEncoderConfiguration, repeat steps 12-15 with next video source configuration from step 11. If there are no other video source configuration, skip other steps and go to the next test.
- 16. ONVIF Client invokes AddVideoEncoderConfigurationRequest (ConfigurationToken = first video encoder configuration from the list on step 15 (VEC1), ProfileToken = New Profile) message to add VideoEncoderConfiguration to the new profile.
- 17. The DUT sends AddVideoEncoderConfigurationResponse message.
- 18. ONVIF Client invokes DeleteProfileRequest (**ProfileToken = New Profile**) message to remove profile with video encoder configuration.
- 19. The DUT sends DeleteProfileResponse message.
- 20. ONVIF Client invokes GetVideoEncoderConfigurationRequest (**ConfigurationToken = VEC1**) message to retrieve video encoder configuration parameters.
- 21. The DUT sends GetVideoEncoderConfigurationResponse message.
- 22. Check that **UseCount = usecount1**, in GetVideoEncoderConfigurationResponse where usecount1 is UseCount value for VEC1 in the list on step 4. If step 8 is executed, the usecount1 could be reduced by 1.
- 23. ONVIF Client restores DUT configuration.

Test Result:

PASS –

DUT passes all assertions.

FAIL -

The DUT did not send valid GetVideoEncoderConfigurationsResponse message.

The DUT did not send valid AddVideoEncoderConfigurationResponse message.

The DUT did not send valid DeleteProfileResponse message.



The DUT did not send valid GetVideoEncoderConfigurationResponse message.

The DUT did not send valid GetCompatibleVideoEncoderConfigurationsResponse message.

The DUT did not send valid GetCompatibleVideoSourceConfigurationsResponse message.

UseCount value is not decreased by 1 after deletion of the profile with the VideoEncoderConfiguration.

Note: See Annex A.5 for Name and Token Parameters Length limitations.

4.4.11 VIDEO ENCODER CONFIGURATION USE COUNT (SET VIDEO ENCODER CONFIGURATION)

Test Label: Media Service Video Encoder Configuration Use Count Validation after Changing of Video Encoder Configuration Validation.

Test Case ID: MEDIA-2-3-17

ONVIF Core Specification Coverage: Get video encoder configurations, Get video encoder configuration, Modify a video encoder configuration.

Command Under Test: GetVideoEncoderConfigurations, GetVideoEncoderConfiguration, SetVideoEncoderConfiguration

WSDL Reference: media.wsdl

Test Propose: To check Video Encoder Configuration use count after setting video encoder configuration parameters.

Pre-Requisite: Media is supported by DUT. ONVIF Client gets the Media Service entry point by GetCapabilities command.

Test Configuration: ONVIF Client and DUT

ONVIF	Client	DUT	
Retrieve list of	GetVideoEncoderConfigura	itionsRequest	Start DUT
Receive and	GetVideoEncoderConfigura Response (list of existing VideoEncoderConfiguration	itions	Send list of existing VideoEncoder Configurations
Set new UseCount	SetVideoEncoderConfigura (VideoEncoderConfiguration UseCount = NewUseCount)	tionRequest n = VEC1,)	
for VEC1 Receive and	SetVideoEncoderConfigurat (empty)	tionResponse	Change video encoder configuration
Retrieve parameters of	GetVideoEncoderConfigura (ConfigurationToken = VEC	tionRequest 1)	
VEC1 Receive and validate VEC1	GetVideoEncoderConfigura (parameters for VEC1) ◀	tionResponse	Send parameters for VEC1

Test Procedure:

- 1. Start an ONVIF Client.
- 2. Start the DUT.
- 3. ONVIF Client invokes GetVideoEncoderConfigurationsRequest message to retrieve list of video encoder configurations from device.
- 4. The DUT sends GetVideoEncoderConfigurationsResponse message.
- 5. ONVIF Client invokes SetVideoEncoderConfigurationRequest (token = the first Video Encoder Configuration from the list on step 4(VEC1), UseCount = NewUseCount) message to set parameters for video encoder configuration.
- 6. The DUT sends SetVideoEncoderConfigurationResponse message.
- 7. ONVIF Client invokes GetVideoEncoderConfigurationRequest (ConfigurationToken = VEC1) message to retrieve video encoder configuration parameters.
- 8. The DUT sends GetVideoEncoderConfigurationResponse message.
- 9. Check that UseCount = usecount1, in GetVideoEncoderConfigurationResponse where usecount1 is UseCount value for VEC1 in the list on step 4.

Test Result:

PASS –

DUT passes all assertions.

FAIL -

The DUT did not send GetVideoEncoderConfigurationsResponse message.

The DUT did not send valid GetVideoEncoderConfigurationsResponse message.

The DUT did not send RemoveVideoEncoderConfigurationResponse message.

The DUT did not send valid RemoveVideoEncoderConfigurationResponse message.

The DUT did not send GetVideoEncoderConfigurationResponse message.

The DUT did not send valid GetVideoEncoderConfigurationResponse message.

UseCount value is changed after trying to set UseCount value.

Note: See Annex A.5 for Name and Token Parameters Length limitations.

4.4.12 VIDEO ENCODER CONFIGURATIONS – ALL SUPPORTED VIDEO ENCODINGS (ALL VIDEO ENCODER CONFIGURATIONS)

Test Label: Set All Supported Encodings Verification.

Test Case ID: MEDIA-2-3-18

ONVIF Core Specification Coverage: None

Command Under Test: None

WSDL Reference: media.wsdl

Test Purpose: To verify whether all supported encodings can be set.

Pre-Requisite: Media is supported by DUT. Media Service was received from the DUT.

Test Configuration: ONVIF Client and DUT

ient	DUT
GetVideoEncoderConfigurations Request	Start DUT
GetVideoEncoderConfigurations Response (list of existing VideoEncoderConfigurations)	Send list of existing VideoEncoder Configurations
GetVideoEncoderConfiguration OptionsRequest (video encoder configuration token)	
GetVideoEncoderConfiguration OptionsResponse (Options)	Send options for VideoEncoder Configuration
SetVideoEncoderConfiguration Request (Configuration parameters)	
SetVideoEncoderConfiguration Response	Modify video encoder configuration
GetVideoEncoderConfigurations Request (video encoder configuration token)	
GetVideoEncoderConfigurations Response (video encoder configuration) ◀	Send modified video encoder configuration
	tient GetVideoEncoderConfigurations Request GetVideoEncoderConfigurations Response (list of existing VideoEncoderConfiguration OptionsRequest (video encoder configuration token) GetVideoEncoderConfiguration OptionsResponse (Options) SetVideoEncoderConfiguration Request (Configuration parameters) SetVideoEncoderConfiguration Response GetVideoEncoderConfiguration Response GetVideoEncoderConfigurations Request (video encoder configuration token) GetVideoEncoderConfigurations Request (video encoder configuration token)

Test Procedure:

- 1. Start an ONVIF Client.
- 2. Start the DUT.
- 3. ONVIF Client will invoke GetVideoEncoderConfigurationsRequest message to retrieve all DUT video encoder configurations.
- 4. Verify the GetVideoEncoderConfigurationsResponse message from the DUT.
- 5. ONVIF Client will invoke GetVideoEncoderConfigurationOptionsRequest message (ConfigurationToken = "Token1", where "Token1" is a first video encoder configuration token



from GetVideoEncoderConfigurationsResponse message, no ProfileToken) to retrieve supported video encoder configuration options.

- 6. Verify the GetVideoEncoderConfigurationOptionsResponse message from the DUT.
- SetVideoEncoderConfigurationReguest 7. ONVIF Client will invoke message (Configuration.token = "Token1", where "Token1" is a first video encoder configuration token GetVideoEncoderConfigurationsResponse message, Configuration.Encoding from "codec1", "codec1" where is а first supported codec from GetVideoEncoderConfigurationOptionsResponse message, other Configuration parameters that are applicable for selected Encoding) to change video encoder configuration settings.
- 8. Verify the SetVideoEncoderConfigurationResponse message from the DUT.
- ONVIF Client will invoke GetVideoEncoderConfigurationRequest message (ConfigurationToken = "Token1", where "Token1" is a first video encoder configuration token from GetVideoEncoderConfigurationsResponse message) to retrieve DUT video encoder configuration for specified token.
- 10. Verify the GetVideoEncoderConfigurationResponse message from the DUT. Check that video encoder setting was applied.
- 11. Repeat steps 7-10 for the rest Video Encodings supported by selected configuration.
- 12. Repeat steps 5-11 for the rest Video Encoder Configuration supported by the DUT.

Test Result:

PASS –

The DUT passes all assertions.

FAIL –

The DUT did not send valid GetVideoEncoderConfigurationsResponse message.

The DUT did not send valid GetVideoEncoderConfigurationOptionsResponse message.

The DUT did not send valid SetVideoEncoderConfigurationResponse message.

The DUT sent GetVideoEncoderConfigurationResponse message with Configuration.Encoding other then was set up with SetVideoEncoderConfigurationRequest message.

The DUT did not send valid GetVideoEncoderConfigurationResponse message.

Note: See Annex A.5 for Name and Token Parameters Length limitations.

4.5 Audio Configuration

4.5.1 GET AUDIO SOURCE CONFIGURATION OPTIONS

Test Label: Media Service DUT GetAudioSourceConfigurationOptions Command Validation.

Test Case ID: MEDIA-3-1-7

ONVIF Core Specification Coverage: Get audio source configuration options

Command Under Test: GetAudioSourceConfigurationOptions

WSDL Reference: media.wsdl

Test Propose: To verify the behavior of GetAudioSourceConfigurationOptions command with different parameters.

Pre-Requisite: Audio is supported by DUT. ONVIF Client gets the Media Service entry point by GetCapabilities command.

Test Configuration: ONVIF Client and DUT

ONVIF Client

DUT

Retrieve list of profiles	GetProfilesRequest	Start DUT
Receive and validate list of	GetProfilesResponse	Send list of existing profiles.
profiles Retrieve list of	GetAudioSourceConfigurationsRequest	
Configurations Receive and	GetAudioSourceConfigurationsResponse (list of AudioSourceConfigurations)	Send list of AudioSource Configurations
Retrieve AudioSource	GetAudioSourceConfigurationOptions Request (ConfigurationToken)	
Configuration options Receive and	► GetAudioSourceConfigurationOptions Response	Send GetAudioSource Configuration options
validate response Retrieve	 GetAudioSourceConfigurationOptions Request (ProfileToken) 	·
Configuration options	➡ GetAudioSourceConfigurationOptions Response	Send GetAudioSource Configuration options
validate response Retrieve AudioSource	GetAudioSourceConfigurationOptions Request (ProfileToken, ConfigurationToken)	Quad
Configuration options Receive and	GetAudioSourceConfigurationOptions Response	GetAudioSource Configuration
validate response Retrieve AudioSource	 GetAudioSourceConfigurationOptions Request (empty) 	options
Configuration options	► GetAudioSourceConfigurationOptions Response	Send GetAudioSource Configuration
Receive and validate response	<	οριιοπε

Test Procedure:

- 1. Start an ONVIF Client.
- 2. Start the DUT.

- 3. ONVIF Client invokes GetProfilesRequest message to get list of existing profiles from device.
- 4. The DUT sends GetProfilesResponse message.
- 5. ONVIF Client invokes GetAudioSourceConfigurationsRequest message to get list of existing audio source configurations.
- 6. The DUT sends GetAudioSourceConfigurationsResponse message.
- 7. ONVIF Client invokes GetAudioSourceConfigurationOptionsRequest (ConfigurationToken) message.
- 8. The DUT sends GetAudioSourceConfigurationOptionsResponse message.
- 9. Verify GetAudioSourceConfigurationOptionsResponse message.
- 10. ONVIF Client invokes GetAudioSourceConfigurationOptionsRequest (ProfileToken) message.
- 11. The DUT sends GetAudioSourceConfigurationOptionsResponse message.
- 12. Verify GetAudioSourceConfigurationOptionsResponse message.
- 13. ONVIF Client invokes GetAudioSourceConfigurationOptionsRequest (ProfileToken, ConfigurationToken) message.
- 14. The DUT sends GetAudioSourceConfigurationOptionsResponse message.
- 15. Verify GetAudioSourceConfigurationOptionsResponse message.
- 16. ONVIF Client invokes GetAudioSourceConfigurationOptionsRequest (empty) message.
- 17. The DUT sends GetAudioSourceConfigurationOptionsResponse message.
- 18. Verify GetAudioSourceConfigurationOptionsResponse message.

Test Result:

PASS –

DUT passes all assertions.

FAIL –

The DUT did not send GetProfilesResponse message.

The DUT did not send valid GetProfilesResponse message.

The DUT did not send GetAudioSourceConfigurationsResponse message.

The DUT did not send valid GetAudioSourceConfigurationsResponse message.

The DUT did not send GetAudioSourceConfigurationOptionsResponse message.

The DUT did not send valid GetAudioSourceConfigurationOptionsResponse message for one or more GetAudioSourceConfigurationOptionsRequest messages.

4.5.2 G.711 AUDIO ENCODER CONFIGURATION

Test Label: Media Configuration DUT G.711 Audio Encoder Configuration

Test Case ID: MEDIA-3-1-14

ONVIF Core Specification Coverage: Get audio encoder configurations, Get audio encoder configuration, Get audio encoder configuration options, Modify audio encoder configurations

Command Under Test: None

WSDL Reference: media.wsdl

Test Purpose: To verify DUT G.711 Audio Encoder Configuration Setting

Pre-Requisite: Audio is supported by DUT

Test Configuration: ONVIF Client and DUT

ONVIF Client		DUT
	GetAudioEncoderConfigurationsRequest (Empty)	Start DUT
Receive and Validate Audio Encoder	GetAudioEncoderConfigurations Response (Audio Encoder configurations)	Send all Audio Encoder Configurations
Comgarations	GetAudioEncoderConfigurationOptions Request (Audio Encoder Cfg Token)	Send audio encoder configuration options
Receive and validate audio encoder configuration options	GetAudioEncoderConfigurationOptions Response (Audio Encoder Cfg Options)	
Set G.711 audio encoder configuration	SetAudioEncoderConfigurationRequest (Invalid G.711 Cfg values, FALSE)	
with invalid parameters	SOAP Fault Message (InvalidArgVal/ConfigModify)	Sond SOAP fault
	 SetAudioEncoderConfigurationRequest (G.711 Configuration, FALSE) 	message
Set G.711 audio encoder configuration	SetAudioEncoderConfigurationResponse (Empty)	•
	GetAudioEncoderConfigurationRequest (G.711 Encoder Configuration token)	Modify G.711 audio encoder configuration
Receive and Validate modified G.711 Encoder Configuration	GetAudioEncoderConfigurationResponse (G.711 Encoder Configuration)	Send modified G.711 audio encoder configuration
		1

Test Procedure:

- 1. Start an ONVIF Client.
- 2. Start the DUT.
- 3. ONVIF Client invokes GetAudioEncoderConfigurations request.
- 4. DUT sends the list of supported audio encoder configurations in GetAudioEncoderConfigurationsResponse message.
- 5. ONVIF Client invokes GetAudioEncoderConfigurationOptions Request (Audio Encoder Configuration token) to retrieve audio encoder configuration options for the specified audio encoder configuration.
- 6. DUT sends the range of configurable values for the received audio encoder configuration in the GetAudioEncoderConfigurationOptionsResponse message.
- 7. Test steps -5 & 6 have to be repeated for all audio encoder configurations until ONVIF Client finds a audio encoder configuration with G.711 encoding support
- 8. ONVIF Client invokes SetAudioEncoderConfiguration request with G.711 configuration values outside the range defined in the GetAudioEncoderConfigurationOptionsResponse and 'ForcePersistence' flag as 'FALSE'.
- 9. DUT send the SOAP 1.2 fault message (InvalidArgVal/ConfigModify).
- 10. ONVIF Client verifies the SOAP fault message sent by DUT.
- 11. ONVIF Client invokes SetAudioEncoderConfiguration request (Encoding = "G711", Bit Rate = r1, Sample Rate = r2 and force persistence = false). These values will be taken from the GetAudioEncoderConfigurationOptionsResponse message.
- 12. DUT modifies G.711 audio encoder configuration and responds with SetAudioEncoderConfigurationResponse message indicating success.
- 13. ONVIF Client verifies the G.711 audio Encoder Configuration settings on DUT by invoking GetAudioEncoderConfiguration request.
- 14. DUT sends modified G.711 audio Encoder Configuration in the GetAudioEncoderConfigurationResponse message (Encoding = "G711", Bit Rate = r1, Sample Rate = r2).

Test Result:

PASS –

DUT passes all assertions.

FAIL -

DUT did not send GetAudioEncoderConfigurationsResponse message.

DUT did not send GetAudioEncoderConfigurationOptionsResponse message.

DUT doesn't support G.711 audio encoding.

DUT did not send the SOAP 1.2 fault message (InvalidArgVal/ConfigModify) for invalid SetAudioEncoderConfiguration request.



DUT did not send SetAudioEncoderConfigurationResponse message.

DUT did not send GetAudioEncoderConfigurationResponse message.

The DUT did not modify G.711 Audio Encoder Configuration.

Note: See Annex A.5 for Name and Token Parameters Length limitations.

4.5.3 G.726 AUDIO ENCODER CONFIGURATION

Test Label: Media Configuration DUT G.726 Audio Encoder Configuration

Test Case ID: MEDIA-3-1-15

ONVIF Core Specification Coverage: Get audio encoder configurations, Get audio encoder configuration, Get audio encoder configuration options, Modify audio encoder configurations

Command Under Test: None

WSDL Reference: media.wsdl

Test Purpose: To verify DUT G.726 Audio Encoder Configuration Setting

Pre-Requisite: Audio is supported by DUT and G.726 is implemented by DUT.

Test Configuration: ONVIF Client and DUT

ONVIF Client		DUT
	GetAudioEncoderConfigurationsRequest (Empty)	Start DUT
Receive and Validate Audio Encoder Configurations	GetAudioEncoderConfigurations Response (Audio Encoder configurations)	Send all Audio Encoder Configurations
	GetAudioEncoderConfigurationOptions Request (Audio Encoder Cfg Token)	Send audio encoder configuration options
Receive and validate audio encoder configuration options	GetAudioEncoderConfigurationOptions Response (Audio Encoder Cfg Options)	
Set G.726 audio encoder configuration with invalid parameters	SetAudioEncoderConfigurationRequest (Invalid G.726 Cfg values, FALSE)	
	SOAP Fault Message (InvalidArgVal/ConfigModify)	
	 SetAudioEncoderConfigurationRequest (G.726 Configuration, FALSE) 	message
Set G.726 audio encoder configuration	SetAudioEncoderConfigurationResponse (Empty)	
	 GetAudioEncoderConfigurationRequest (G.726 Encoder Configuration token) 	Modify G.726 audio encoder configuration
Receive and Validate modified G.726 Encoder Configuration	GetAudioEncoderConfigurationResponse (G.726 Encoder Configuration)	Send modified G.726 audio encoder configuration

Test Procedure:

- 1. Start an ONVIF Client.
- 2. Start the DUT.
- 3. ONVIF Client invokes GetAudioEncoderConfigurations request.
- 4. DUT sends the list of supported audio encoder configurations in GetAudioEncoderConfigurationsResponse message.

-124-

- 5. ONVIF Client invokes GetAudioEncoderConfigurationOptions Request (Audio Encoder Configuration token) to retrieve audio encoder configuration options for the specified audio encoder configuration.
- 6. DUT sends the range of configurable values for the received audio encoder configuration in the GetAudioEncoderConfigurationOptionsResponse message.
- 7. Test steps -5 & 6 have to be repeated for all audio encoder configurations until ONVIF Client finds a audio encoder configuration with G.726 encoding support
- 8. ONVIF Client invokes SetAudioEncoderConfiguration request with G.726 configuration values outside the range defined in the GetAudioEncoderConfigurationOptionsResponse and **'ForcePersistence'** flag as **'FALSE'**.
- 9. DUT send the SOAP 1.2 fault message (InvalidArgVal/ConfigModify).
- 10. ONVIF Client verifies the SOAP fault message sent by DUT.
- 11. ONVIF Client invokes SetAudioEncoderConfiguration request (Encoding = "G726", Bit Rate = r1, Sample Rate = r2 and force persistence = false). These values will be taken from the GetAudioEncoderConfigurationOptionsResponse message.
- 12. DUT modifies G.726 audio encoder configuration and responds with SetAudioEncoderConfigurationResponse message indicating success.
- 13. ONVIF Client verifies the G.726 audio Encoder Configuration settings on DUT by invoking GetAudioEncoderConfiguration request.
- 14. DUT sends the modified G.726 audio Encoder Configuration in the GetAudioEncoderConfigurationResponse message (Encoding = "G726", Bit Rate = r1, Sample Rate = r2).

Test Result:

PASS –

DUT passes all assertions.

FAIL –

DUT did not send GetAudioEncoderConfigurationsResponse message.

DUT did not send GetAudioEncoderConfigurationOptionsResponse message.

DUT doesn't support G.726 audio encoding.

DUT did not send the SOAP 1.2 fault message (InvalidArgVal/ConfigModify) for invalid SetAudioEncoderConfiguration request.

DUT did not send SetAudioEncoderConfigurationResponse message.

DUT did not send GetAudioEncoderConfigurationResponse message.

The DUT did not modify G.726 Audio Encoder Configuration.

Note: See Annex A.5 for Name and Token Parameters Length limitations.

4.5.4 AAC AUDIO ENCODER CONFIGURATION

Test Label: Media Configuration AAC Audio Encoder Configuration

Test Case ID: MEDIA-3-1-16

ONVIF Core Specification Coverage: Get audio encoder configurations, Get audio encoder configuration, Get audio encoder configuration options, Modify audio encoder configurations

Command Under Test: None

WSDL Reference: media.wsdl

Test Purpose: To verify DUT AAC Audio Encoder Configuration Setting

Pre-Requisite: Audio is supported by DUT and AAC is implemented by DUT.

Test Configuration: ONVIF Client and DUT

ONVIF Client		DUT
	GetAudioEncoderConfigurationsRequest (Empty)	Start DUT
Receive and Validate Audio Encoder	GetAudioEncoderConfigurations Response (Audio Encoder configurations)	Send all Audio Encoder) Configurations
Configurations	 GetAudioEncoderConfigurationOptions Request (Audio Encoder Cfg Token) 	Send audio encoder configuration options
Receive and validate audio encoder configuration options	GetAudioEncoderConfigurationOptions Response (Audio Encoder Cfg Options)	
Set AAC audio	 SetAudioEncoderConfigurationRequest (Invalid AAC Cfg values, FALSE) 	
with invalid parameters	SOAP Fault Message (InvalidArgVal/ConfigModify)	->
	 SetAudioEncoderConfigurationRequest (AAC Configuration, FALSE) 	Send SOAP fault message
Set AAC audio encoder configuration	SetAudioEncoderConfigurationResponse (Empty)	
	 GetAudioEncoderConfigurationRequest (AAC Encoder Configuration token) 	Modify AAC audio encoder configuration
Receive and Validate modified AAC Encoder Configuration	GetAudioEncoderConfigurationResponse (AAC Encoder Configuration)	Send modified AAC audio encoder configuration

Test Procedure:

- 1. Start an ONVIF Client.
- 2. Start the DUT.
- 3. ONVIF Client invokes GetAudioEncoderConfigurations request.
- 4. DUT sends the list of supported audio encoder configurations in GetAudioEncoderConfigurationsResponse message.
- 5. ONVIF Client invokes GetAudioEncoderConfigurationOptions Request (Audio Encoder Configuration token) to retrieve audio encoder configuration options for the specified audio encoder configuration.
- 6. DUT sends the range of configurable values for the received audio encoder configuration in the GetAudioEncoderConfigurationOptionsResponse message.
- 7. Test steps -5 & 6 have to be repeated for all audio encoder configurations until ONVIF Client finds a audio encoder configuration with AAC encoding support
- 8. ONVIF Client invokes SetAudioEncoderConfiguration request with AAC configuration values outside the range defined in the GetAudioEncoderConfigurationOptionsResponse and 'ForcePersistence' flag as 'FALSE'.
- 9. DUT send the SOAP 1.2 fault message (InvalidArgVal/ConfigModify).
- 10. ONVIF Client verifies the SOAP fault message sent by DUT.
- 11. ONVIF Client invokes SetAudioEncoderConfiguration request (Encoding = "AAC", Bit Rate = r1, Sample Rate = r2 and force persistence = false). These values will be taken from the GetAudioEncoderConfigurationOptionsResponse message.
- 12. DUT modifies AAC audio encoder configuration and responds with SetAudioEncoderConfigurationResponse message indicating success.
- 13. ONVIF Client verifies the AAC audio Encoder Configuration settings on DUT by invoking GetAudioEncoderConfiguration request.
- 14. DUT sends modified AAC audio Encoder Configuration in the GetAudioEncoderConfigurationResponse message (Encoding = "AAC", Bit Rate = r1, Sample Rate = r2).

Test Result:

PASS –

DUT passes all assertions.

FAIL -

DUT did not send GetAudioEncoderConfigurationsResponse message.

DUT did not send GetAudioEncoderConfigurationOptionsResponse message.

DUT doesn't support AAC audio encoding.

DUT did not send the SOAP 1.2 fault message (InvalidArgVal/ConfigModify) for invalid SetAudioEncoderConfiguration request.



DUT did not send SetAudioEncoderConfigurationResponse message.

DUT did not send GetAudioEncoderConfigurationResponse message.

The DUT did not modify AAC Audio Encoder Configuration.

Note: See Annex A.5 for Name and Token Parameters Length limitations.

4.5.5 GET AUDIO SOURCE CONFIGURATION – INVALID CONFIGURATIONTOKEN

Test Label: Media Service DUT GetAudioSourceConfiguration Command Validation (Invalid Configuration Token).

Test Case ID: MEDIA-3-1-17

ONVIF Core Specification Coverage: Get audio source configuration

Command Under Test: GetAudioSourceConfiguration

WSDL Reference: media.wsdl

Test Propose: To verify the behavior of GetAudioSourceConfiguration command in case of invalid ConfigurationToken.

Pre-Requisite: Audio is supported by DUT. ONVIF Client gets the Media Service entry point by GetCapabilities command.

Test Configuration: ONVIF Client and DUT

Test Sequence:



Test Procedure:

- 1. Start an ONVIF Client.
- 2. Start the DUT.
- 3. ONVIF Client invokes GetAudioSourceConfigurationRequest (invalid ConfigurationToken) message.
- 4. The DUT will generate a SOAP 1.2 fault message (InvalidArgVal/NoConfig).

Test Result:

-129-

Onvir Driving IP-based physical security through global standardization

PASS –

DUT passes all assertions.

FAIL -

The DUT did not send SOAP 1.2 fault message.

The DUT sent incorrect SOAP 1.2 fault message (fault code, namespace, etc.).

Note: Other faults than specified in the test are acceptable but specified is preferable.

Note: See Annex A.5 for Name and Token Parameters Length limitations.

4.5.6 GET AUDIO SOURCE CONFIGURATION OPTIONS – INVALID PROFILETOKEN

Test Label: Media Service DUT GetAudioSourceConfigurationOptions Command Validation (Invalid Profile Token).

Test Case ID: MEDIA-3-1-18

ONVIF Core Specification Coverage: Get audio source configuration options

Command Under Test: GetAudioSourceConfigurationOptions

WSDL Reference: media.wsdl

Test Propose: To verify the behavior of GetAudioSourceConfigurationOptions command in case of invalid ProfileToken.

Pre-Requisite: Audio is supported by DUT. ONVIF Client gets the Media Service entry point by GetCapabilities command.

Test Configuration: ONVIF Client and DUT

Test Sequence:



Test Procedure:

- 1. Start an ONVIF Client.
- 2. Start the DUT.



- 3. ONVIF Client invokes GetAudioSourceConfigurationOptionsRequest (invalid ProfileToken) message.
- 4. The DUT will generate a SOAP 1.2 fault message (InvalidArgVal/NoProfile).

Test Result:

PASS –

DUT passes all assertions.

FAIL -

The DUT did not send SOAP 1.2 fault message.

The DUT sent incorrect SOAP 1.2 fault message (fault code, namespace, etc.).

Note: Other faults than specified in the test are acceptable but specified is preferable.

Note: See Annex A.5 for Name and Token Parameters Length limitations.

4.5.7 GET AUDIO SOURCE CONFIGURATION OPTIONS - INVALID CONFIGURATION TOKEN

Test Label: Media Service DUT GetAudioSourceConfigurationOptions Command Validation (Invalid ConfigurationToken).

Test Case ID: MEDIA-3-1-19

ONVIF Core Specification Coverage: Get audio source configuration options

Command Under Test: GetAudioSourceConfigurationOptions

WSDL Reference: media.wsdl

Test Propose: To verify the behavior of GetAudioSourceConfigurationOptions command in case of invalid ConfigurationToken.

Pre-Requisite: Audio is supported by DUT. ONVIF Client gets the Media Service entry point by GetCapabilities command.

Test Configuration: ONVIF Client and DUT





Test Procedure:

- 1. Start an ONVIF Client.
- 2. Start the DUT.
- 3. ONVIF Client invokes GetAudioSourceConfigurationOptionsRequest (invalid ConfigurationToken) message.
- 4. The DUT will generate a SOAP 1.2 fault message (InvalidArgVal/NoConfig).

Test Result:

PASS -

DUT passes all assertions.

FAIL -

The DUT did not send SOAP 1.2 fault message.

The DUT sent incorrect SOAP 1.2 fault message (fault code, namespace, etc.).

Note: Other faults than specified in the test are acceptable but specified is preferable.

Note: See Annex A.5 for Name and Token Parameters Length limitations.

4.5.8 SET AUDIO SOURCE CONFIGURATION - INVALID TOKEN

Test Label: Media Service DUT SetAudioSourceConfiguration Command Validation (Invalid Token).

Test Case ID: MEDIA-3-1-20

ONVIF Core Specification Coverage: Modify an audio source configuration

Command Under Test: SetAudioSourceConfiguration

WSDL Reference: media.wsdl

Test Propose: To verify the behavior of SetAudioSourceConfiguration command in case of invalid ConfigurationToken.

Pre-Requisite: Audio is supported by DUT. ONVIF Client gets the Media Service entry point by GetCapabilities command.

Test Configuration: ONVIF Client and DUT



Generate SOAP 1.2 fault message

Start DUT

1.2 fault

Receive and validate the SOAP

Test Procedure:

- 1. Start an ONVIF Client.
- 2. Start the DUT.
- 3. ONVIF Client invokes SetAudioSourceConfigurationRequest (invalid ConfigurationToken) message.
- 4. The DUT will generate a SOAP 1.2 fault message (InvalidArgVal/NoConfig).

(InvalidArgVal/NoConfig)

Test Result:

PASS -

DUT passes all assertions.

FAIL –

The DUT did not send SOAP 1.2 fault message.

The DUT sent incorrect SOAP 1.2 fault message (fault code, namespace, etc.).

Note: Other faults than specified in the test are acceptable but specified is preferable.

Note: See Annex A.5 for Name and Token Parameters Length limitations.

4.5.9 SET AUDIO ENCODER CONFIGURATION

Test Label: Media Configuration Audio Encoder Configuration

Test Case ID: MEDIA-3-1-21

ONVIF Core Specification Coverage: Create media profile, Add audio source configuration to a profile, Add audio encoder configuration to a profile, Remove audio source configuration from a profile, Remove audio encoder configuration from a profile, Delete media profile, Get audio source configurations, Get audio encoder configurations, Get compatible audio encoder configurations, Set audio encoder configuration

Command Under Test: SetAudioEncoderConfiguration

WSDL Reference: media.wsdl

Test Purpose: To verify DUT Audio Encoder Configuration Operations ONVIF www.onvif.org

info@onvif.org

-133-

Onvir Driving IP-based physical security through global standardization

Pre-Requisite: Audio is supported by DUT. Test Configuration: ONVIF Client and DUT Test Sequence:

ONVIF C	lient	DUT	
	CreateProfileRequest (ProfileToken = 'testprofileX')	St	art DUT
Receive and Validate CreateProfileResponse message	CreateProfileResponse or SOAP 1.2 fault message (Action/MaxNVTProfiles)	Cr se me	reate media profile or end SOAP 1.2 fault essage
	Annex A.3		
	GetCompatibleAudioSource ConfigurationsRequest message (ProfileToken = 'testprofileX')		
Receive and Validate GetCompatibleAudio	GetCompatibleAudioSource ConfigurationsResponse (Audio Source Configurations)	Se so co	end list of audio ource configurations ompatible with profile
SourceConfigurations Response message	AddAudioSourceConfiguration Request (ProfileToken = 'testprofileX', Audio Source ConfigurationToken)		
Receive and Validate GetCompatibleAudio	AddAudioSourceConfiguration Response	Ac co re:	dd audio source onfiguration and send sponse
Response message	GetCompatibleAudioEncoder ConfigurationsRequest (ProfileToken = 'testprofileX')		
Receive and Validate GetCompatibleAudio	GetCompatibleAudioEncoder ConfigurationsResponse (Audio Encoder Configurations)	Se en co	end list of audio acoder configurations ampatible with profile
ConfigurationsRespon se message	AddAudioEncoderConfiguration Request (ProfileToken = 'testprofileX', Audio Encoder ConfigurationToken)		
AddAudioEncoder Configuration Response message	AddAudioEncoderConfiguration Response	Ac co res	dd audio encoder onfiguration and send sponse

ONVIF C	ient	DUT	
	GetAudioEncoderConfiguration OptionsRequest (ProfileToken = 'testprofileX')		
Receive and Validate GetAudioEncoder ConfigurationOptions Response message	GetAudioEncoderConfiguration OptionsResponse (Audio encoder configuration options)		Send audio encoder options for specified profile
	SetAudioEncoderConfiguration Request message (ConfigurationToken, Encoding = [other than current], Bitrate = [other than current], SampleRate = [other than current], ForcePersistence = false		
Receive and Validate SetAudioEncoder	SetAudioEncoderConfiguration Response	-	Modify audio encoder configuration
Configuration Response message	GetAudioEncoderConfiguration Request (Audio encoder configuration token)		
Receive and Validate GetAudioEncoder ConfigurationRespons	GetAudioEncoderConfiguration Response (Audio encoder configuration)	→	Send modified audio encoder configuration
e message	DeleteProfileRequest (ProfileToken = 'testprofileX')	-	
Receive and Validate DeleteProfile Response message	DeleteProfileResponse		Delete created profile and send response

Test Procedure:

- 1. Start an ONVIF Client.
- 2. Start the DUT.
- 3. ONVIF Client invokes CreateProfileRequest message (ProfileToken = 'testprofileX') to create new profile.



- 4. Verify CreateProfileResponse message or SOAP 1.2 fault message (Action/MaxNVTProfiles) from the DUT. If fault was received execute Annex A.3.
- 5. ONVIF Client will invoke GetCompatibleAudioSourceConfigurationsRequest message (ProfileToken = 'testprofileX') to retrieve the list of audio source configurations compatible with profile.
- 6. ONVIF Client verifies the list of audio source configurations sent by DUT.
- 7. ONVIF Client invokes AddAudioSourceConfigurationRequest message (ProfileToken = 'testprofileX', ConfigurationToken as one of the tokens received in the GetCompatibleAudioSourceConfigurationsResponse message) to add audio source configuration to profile.
- 8. DUT adds the audio source configuration to the profile and sends the response.
- 9. ONVIF Client invokes GetCompatibleAudioEncoderConfigurationsRequest message (ProfileToken = 'testprofileX') to retrieve audio encoder configurations compatible with profile.
- 10. DUT sends the list of audio encoder configurations compatible with the received media profile token.
- 11. ONVIF Client invokes AddAudioEncoderConfigurationRequest message (ProfileToken = 'testprofileX', ConfigurationToken as one of the tokens received in the GetCompatibleAudioEncoderConfigurationsResponse message) to add audio encoder configuration to profile.
- 12. DUT adds the audio encoder configuration to the profile and sends the response.
- 13. ONVIF Client invokes GetAudioEncoderConfigurationOptionsRequest (ProfileToken = 'testprofileX') request to retrieve audio encoder options for specified profile.
- 14. DUT sends the audio encoder configuration options which could be applied to audio encoder from specified profile.
- 15. ONVIF Client invokes SetAudioEncoderConfigurationRequest message (ConfigurationToken, Encoding = [other than current], Bitrate = [other than current], SampleRate = [other than current], ForcePersistence = false, where all values was taken from audio encoder configuration options) to change
- 16. DUT sends SetAudioEncoderConfigurationResponse message.
- 17. ONVIF Client invokes GetAudioEncoderConfigurationRequest message (ConfigurationToken) to get new audio encoder configuration parameters.
- 18. DUT sends GetAudioEncoderConfigurationResponse message with parameters specified in set request.
- 19. ONVIF Client checks that Audio configuration in GetAudioEncoderConfigurationResponse message is the same as in SetAudioEncoderConfigurationRequest message.
- If used created Media Profile then ONVIF Client invokes DeleteProfileRequest message (ProfileToken = 'testprofileX'). Otherwise ONVIF client skip rest steps and restore profile settings.
- 21. DUT deletes the media profile and sends the response.

Test Result:

PASS -

The DUT passes all assertions.

FAIL -

The DUT did not send valid CreateProfileResponse message.

The DUT did not send valid GetCompatibleAudioSourceConfigurationsResponse message.

The DUT did not send valid AddAudioSourceConfigurationResponse message.

The DUT did not send valid GetCompatibleAudioEncoderConfigurationsResponse message.

The DUT did not send valid AddAudioEncoderConfigurationResponse message.

The DUT did not send valid GetAudioEncoderConfigurationOptionsResponse message.

The DUT sent Audio configuration in GetAudioEncoderConfigurationResponse message different with SetAudioEncoderConfigurationRequest message.

Note: See Annex A.5 for Name and Token Parameters Length limitations.

4.5.10 AUDIO SOURCE CONFIGURATION

Test Label: Media Configuration Audio Source Configuration

Test Case ID: DRAFT-MEDIA-3-1-22

ONVIF Core Specification Coverage: Create media profile, Add audio source configuration to a profile, Remove audio source configuration from a profile, Delete media profile, Get audio sources, Get audio source configurations, Get audio source configuration, Get compatible audio source configurations, Get audio source configuration options, Modify an audio source configuration.

Command Under Test: None

WSDL Reference: media.wsdl

Test Purpose: To verify DUT Audio Source Configuration Operations

Pre-Requisite: Audio is supported by the DUT

Test Configuration: ONVIF Client and DUT

ONVI	F Client	DUT
Create new media	CreateProfileRequest (Name = "testprofileX")	Start DUT
	CreateProfileResponse (Empty profile wit token) or SOAP 1.2 fault message (Action/MaxNVTProfiles)	h Send create profile response
	Annex A.3	>
	GetAudioSourcesRequest (Empty)	
Receive and Validate	GetAudioSourcesResponse (Audio sources)	Send Audio Sources of the device
Audio Sources	 GetAudioSourceConfigurationsRequest (Empty) 	Send all Audio Source Configurations
Receive and Validate Audio Source	GetAudioSourceConfigurationsResponse (Audio Source Configurations)	
Configurations	GetCompatibleAudioSource ConfigurationsRequest (Media Profile token)	
Receive and Validate Audio Source Configurations	GetCompatibleAudioSource ConfigurationsResponse (Audio Source Configurations)	Send Audio Source Configurations compatible with the received media profile
Add audio source configuration to new media profile	AddAudioSourceConfigurationRequest (Profile Token, audio source cfg)	
	AddAudioSourceConfigurationResponse (empty message)	Add audio source configuration and send response
	←	

ONVIF	Client	DUT
	GetAudioSourceConfigurationOptions Request (Audio Source Cfg)	Send configurable options for audio Source configuration
Receive and Validate	GetAudioSourceConfigurationOptions Response (Audio Source Cfg options)	
Audio Source configurable options Set Invalid audio	 SetAudioSourceConfigurationRequest (Invalid Audio Source cfg values, FALSE) 	_
source cfg values	SOAP Fault Message (InvalidArgVal/ConfigModify)	Send SOAP 1.2 fault
Set Audio Source	SetAudioSourceConfigurationRequest (Audio Source Cfg, FALSE)	Modify Audio Source
Configuration	SetAudioSourceConfigurationResponse (Empty)	Configuration
	GetAudioSourceConfigurationRequest (Audio Source Cfg Token)	
Receive and Validate Audio Source	GetAudioSourceConfigurationResponse (Audio Source Cfg)	Send the modified Audio Source Configuration
Configuration	RemoveAudioSourceConfigurationRequest (Profile Token)	Remove Audio Source
Receive the message	RemoveAudioSourceConfigurationRespon se (empty message)	Cfg from media profile.
Send Delete Profile request	DeleteProfileRequest (Profile Token)	
	DeleteProfileResponse (Empty message)	Delete media profile.

Test Procedure:

- 1. Start an ONVIF Client.
- 2. Start the DUT.
- 3. ONVIF Client invokes CreateProfile request with ProfileToken = 'testprofileX'.
- 4. Verify CreateProfileResponse message or SOAP 1.2 fault message (Action/MaxNVTProfiles) from the DUT. If fault was received execute Annex A.3.
- 5. ONVIF Client invokes GetAudioSources request to retrieve the existing audio sources of DUT.
- 6. ONVIF Client verifies the list of audio sources sent by DUT.
- 7. ONVIF Client invokes GetAudioSourceConfigurations request to retrieve the list of audio source configurations supported by the DUT.
- 8. ONVIF Client verifies the list of audio source configurations sent by DUT.
- 9. ONVIF Client invokes GetCompatibleAudioSourceConfigurations request with 'testprofileX' as ProfileToken.
- 10. The DUT sends the list of audio source configurations compatible with the received media profile token.
- 11. ONVIF Client invokes AddAudioSourceConfiguration request message with ProfileToken as 'testprofileX' and ConfigurationToken as one of the tokens in GetCompatibleAudioSourceConfigurations response.
- 12. The DUT adds the audio source configuration to the media profile and sends the response.
- 13. ONVIF Client invokes GetAudioSourceConfigurationOptions request with ConfigurationToken as the same token sent in the AddAudioSourceConfiguration request.
- 14. The DUT sends the configurable options supported for the received audio source configuration.
- 15. ONVIF Client invokes SetAudioSourceConfiguration request with audio source configuration values outside the range defined in GetAudioSourceConfigurationOptions response and 'ForcePersistence' flag as 'FALSE'.
- 16. The DUT send the SOAP 1.2 fault message (InvalidArgVal/ConfigModify).
- 17. ONVIF Client verifies the SOAP fault message sent by DUT.
- ONVIF Client invokes SetAudioSourceConfiguration request with audio source configuration values as defined in GetAudioSourceConfigurationOptions response and 'ForcePersistence' flag as 'FALSE'.
- 19. The DUT modifies the audio source configuration and sends the SetAudioSourceConfigurationResponse message indicating success.
- 20. ONVIF Client verifies the modified audio source configuration by invoking the GetAudioSourceConfiguration request.
- 21. The DUT sends the modified audio source configuration in GetAudioSourceConfiguration response.



- 22. ONVIF Client invokes RemoveAudioSourceConfiguration request with ProfileToken as 'testprofileX'.
- 23. The DUT removes the audio source configuration token from media profile and sends the response.
- 24. If profile with profile token = testprofileX was created during test execution, then ONVIF Client invokes DeleteProfile request with ProfileToken as 'testprofileX'. Otherwise, ONVIF client skip the rest steps and restore profile settings.
- 25. The DUT deletes the media profile and sends the response.

Test Result:

PASS –

The DUT passed all assertions.

FAIL –

The DUT did not send CreateProfileResponse message.

The DUT did not send valid GetAudioSourcesResponse message.

The DUT did not send valid GetAudioSourceConfigurationsResponse message.

The DUT did not send GetCompatibleAudioSourceConfigurationsResponse message.

The DUT did not send AddAudioSourceConfigurationResponse message.

The DUT did not send GetAudioSourceConfigurationOptionsResponse message.

The DUT did not send the SOAP 1.2 fault message (InvalidArgVal/ConfigModify) for invalid SetAudioSourceConfiguration request.

The DUT did not send SetAudioSourceConfigurationResponse message.

The DUT did not send GetAudioSourceConfigurationResponse message.

The DUT did not modify audio source configuration correctly.

The DUT did not send RemoveAudioSourceConfigurationResponse message.

The DUT did not send DeleteProfileResponse message.

Note: See Annex A.5 for Name and Token Parameters Length limitations.

Note: If profile was deleted during Annex A.3 execution, ONVIF Client restores the deleted profile and profile settings.

4.5.11 AUDIO ENCODER CONFIGURATION

Test Label: Media Configuration Audio Encoder Configuration

Test Case ID: DRAFT-MEDIA-3-1-23

ONVIF Core Specification Coverage: Create media profile, Add audio source configuration to a profile, Add audio encoder configuration to a profile, Remove audio source configuration from a

profile, Remove audio encoder configuration from a profile, Delete media profile, Get audio source configurations, Get audio encoder configurations, Get compatible audio encoder configurations.

Command Under Test: None

WSDL Reference: media.wsdl

Test Purpose: To verify DUT Audio Encoder Configuration Operations

Pre-Requisite: Audio is supported by the DUT

Test Configuration: ONVIF Client and DUT

ONVIF	Client	DUT
Create new media	CreateProfileRequest (Name = "testprofileX")	Start DUT
create new media profile	CreateProfileResponse (Empty profile with token) or SOAP 1.2 fault message (Action/MaxNVTProfiles)	Send create profile response
	Annex A.3	•
	GetAudioSourceConfigurationsRequest (Empty)	
Receive and Validate Audio Source Configurations	GetAudioSourceConfigurationsResponse (Audio Source Configurations)	Send all Audio Source Configurations
Add audio source configuration to new	AddAudioSourceConfigurationRequest (Profile Token, audio source cfg token)	
media profile	AddAudioSourceConfigurationResponse (empty message)	Add audio source configuration and send response
	 GetAudioEncoderConfigurationsRequest (Empty) 	
Receive and Validate Audio Encoder Configurations	GetAudioEncoderConfigurationsResponse (Audio Encoder Configurations)	Send all Audio Encoder Configurations
	GetCompatibleAudioEncoder ConfigurationsRequest (Media Profile token)	Send Audio Encoder
Receive and Validate Audio Encoder Configurations	GetCompatibleAudioEncoder ConfigurationsResponse (Audio Encoder Configurations)	 Configurations compatible with the received media profile

ONVIF Client		DUT
Add audio encoder configuration to new media profile	AddAudioEncoderConfigurationRequest (Profile Token, audio encoder cfg token)	
	AddAudioEncoderConfigurationResponse (empty message)	Add audio encoder configuration and send response
	RemoveAudioEncoderConfiguration Request (Profile Token)	Remove Audio Encoder
Receive the message	RemoveAudioEncoderConfiguration Response (empty message)	Cig iron media prome.
	 RemoveAudioSourceConfigurationRequest (Profile Token) 	Remove Audio Source
Receive the message	RemoveAudioSourceConfiguration Response (empty message)	
Send Delete Profile request	DeleteProfileRequest (Profile Token)	
	DeleteProfileResponse (Empty message)	Delete media profile.

Test Procedure:

- 1. Start an ONVIF Client.
- 2. Start the DUT.
- 3. ONVIF Client invokes CreateProfile request with ProfileToken = 'testprofileX'.


4. Verify CreateProfileResponse message or SOAP 1.2 fault message (Action/MaxNVTProfiles) from the DUT. If fault was received, execute Annex A.3.

-145-

- 5. ONVIF Client will invoke GetAudioSourceConfigurations request to retrieve the list of audio source configurations supported by DUT.
- 6. ONVIF Client verifies the list of audio source configurations sent by DUT.
- 7. ONVIF Client invokes AddAudioSourceConfiguration request message with ProfileToken as 'testprofileX' and ConfigurationToken as one of the tokens received in the GetAudioSourceConfigurations response.
- 8. The DUT adds the audio source configuration to the profile and sends the response.
- 9. ONVIF Client will invoke GetAudioEncoderConfigurations request to retrieve the list of audio encoder configurations supported by DUT.
- 10. ONVIF Client verifies the list of audio encoder configurations sent by DUT.
- 11. ONVIF Client invokes GetCompatibleAudioEncoderConfigurations request with 'testprofileX' as ProfileToken.
- 12. The DUT sends the list of audio encoder configurations compatible with the received media profile token.
- 13. ONVIF Client invokes AddAudioEncoderConfiguration request message with ProfileToken as 'testprofileX' and ConfigurationToken as one of the tokens received in the GetCompatibleAudioEncoderConfigurations response.
- 14. The DUT adds the audio encoder configuration to the profile and sends the response.
- 15. ONVIF Client invokes RemoveAudioEncoderConfiguration request with ProfileToken as 'testprofileX'.
- 16. The DUT removes the audio encoder configuration token from media profile and sends the response.
- 17. ONVIF Client invokes RemoveAudioSourceConfiguration request with ProfileToken as 'testprofileX'.
- 18. The DUT removes the audio source configuration token from media profile and sends the response.
- If profile with profile token = testprofileX was created during test execution, then ONVIF Client invokes DeleteProfile request with ProfileToken as 'testprofileX'. Otherwise, ONVIF client skip rest steps and restore profile settings.
- 20. The DUT deletes the media profile and sends the response.

Test Result:

PASS –

The DUT passed all assertions.

FAIL -

The DUT did not send CreateProfileResponse message.

The DUT did not send valid GetAudioSourceConfigurationsResponse message.

The DUT did not send AddAudioSourceConfigurationResponse message.

The DUT did not send valid GetAudioEncoderConfigurationsResponse message.

The DUT did not send GetCompatibleAudioEncoderConfigurationsResponse message.

The DUT did not send AddAudioEncoderConfigurationResponse message.

The DUT did not send RemoveAudioEncoderConfigurationResponse message.

The DUT did not send RemoveAudioSourceConfigurationResponse message.

DUT did not send DeleteProfileResponse message.

Note: See Annex A.5 for Name and Token Parameters Length limitations.

Note: If profile was deleted during Annex A.3 execution, ONVIF Client restores the deleted profile and profile settings.

4.6 Audio Source Configuration

4.6.1 AUDIO SOURCE CONFIGURATIONS AND PROFILES CONSISTENCY

Test Label: Media Service DUT GetAudioSourceConfigurations Command and GetProfiles Command Consistency Validation.

Test Case ID: MEDIA-3-2-1

ONVIF Core Specification Coverage: Get media profiles, Get audio source configurations

Command Under Test: GetProfiles, GetAudioSourceConfigurations

WSDL Reference: media.wsdl

Test Propose: To check that GetAudioSourceConfigurations command and GetProfiles command are consistent.

Pre-Requisite: Audio is supported by DUT. ONVIF Client gets the Media Service entry point by GetCapabilities command.

Test Configuration: ONVIF Client and DUT

Test Sequence:



Test Procedure:

- 1. Start an ONVIF Client.
- 2. Start the DUT.
- 3. ONVIF Client invokes GetProfilesRequest message to retrieve list of profiles and their configurations.
- 4. The DUT sends GetProfilesResponse message.



- 5. ONVIF Client invokes GetAudioSourceConfigurationsRequest message to retrieve list of Audio Source Configurations from device.
- 6. The DUT sends GetAudioSourceConfigurationsResponse message.
- 7. Check that each AudioSourceConfiguration from GetAudioSourceConfigurationsResponse message has unique token.
- 8. Check that each AudioSourceConfiguration from the GetProfilesResponse message are included in the GetAudioSourceConfigurationsResponse message.
- 9. Check that AudioSourceConfiguration parameters are same in the GetProfilesResponse message and in the GetAudioSourceConfigurationsResponse message for each AudioSourceConfiguration.

Test Result:

PASS -

DUT passes all assertions.

FAIL –

The DUT did not send GetProfilesResponse message.

The DUT did not send valid GetProfilesResponse message.

The DUT did not send GetAudioSourceConfigurationsResponse message.

The DUT did not send valid GetAudioSourceConfigurationsResponse message.

The DUT return two or more AudioSourceConfiguration in GetAudioSourceConfigurationsResponse message with the same ConfigurationToken.

The DUT returned the GetProfilesResponse message with AudioSourceConfigurations that were not included in the GetAudioSourceConfigurationsResponse message.

The DUT returned different parameters list and parameters values in the GetAudioSourceConfigurationsResponse message and in the GetProfilesResponse message for the same AudioSourceConfiguration.

4.6.2 AUDIO SOURCE CONFIGURATIONS AND AUDIO SOURCE CONFIGURATION CONSISTENCY

Test Label: Media Service DUT GetAudioSourceConfigurations Command and GetAudioSourceConfiguration Command Consistency Validation.

Test Case ID: MEDIA-3-2-2

ONVIF Core Specification Coverage: Get audio source configurations, Get audio source configuration

Command Under Test: GetAudioSourceConfigurations, GetAudioSourceConfiguration

WSDL Reference: media.wsdl

Test Propose: To check that GetAudioSourceConfigurations command and GetAudioSourceConfiguration command are consistent.

Pre-Requisite: Audio is supported by DUT. ONVIF Client gets the Media Service entry point by GetCapabilities command.

Test Configuration: ONVIF Client and DUT

Test Sequence:

ONVIF Client DUT Start DUT GetAudioSourceConfigurationsRequest Retrieve list of AudioSource Send list of Configurations existing GetAudioSourceConfigurationsResponse AudioSource (list of existing Receive and Configurations AudioSourceConfigurations) validate response **GetAudioSourceConfigurationRequest** Retrieve (ConfigurationToken) parameters of audio source configuration Send parameters GetAudioSourceConfigurationResponse of audio source (parameters for audio source configuration configuration) Receive and validate response

Test Procedure:

- 1. Start an ONVIF Client.
- 2. Start the DUT.
- 3. ONVIF Client invokes GetAudioSourceConfigurationsRequest message to retrieve list of Audio Source Configurations from device.
- 4. The DUT sends GetAudioSourceConfigurationsResponse message.
- 5. ONVIF Client invokes GetAudioSourceConfigurationRequest (ConfigurationToken) message to retrieve parameters of Audio Source Configuration from device.
- 6. The DUT sends GetAudioSourceConfigurationResponse message.
- 7. Verify that all parameters and their values for AudioSourceConfigurationToken from GetAudioSourceConfigurationsResponse message and GetAudioSourceConfigurationResponse message are same.
- 8. Repeat steps 5-7 for other AudioSourceConfigurations from the GetAudioSourceConfigurationsResponse message.

Test Result:

PASS -

DUT passes all assertions.

FAIL -

The DUT did not send GetAudioSourceConfigurationsResponse message.

The DUT did not send valid GetAudioSourceConfigurationsResponse message.

The DUT did not send GetAudioSourceConfigurationResponse message.

The DUT did not send valid GetAudioSourceConfigurationResponse message.

The DUT did not send equal parameters for AudioSourceConfiguration in the GetAudioSourceConfigurationResponse message and in the GetAudioSourceConfigurationsResponse message.

4.6.3 AUDIO SOURCE CONFIGURATIONS AND AUDIO SOURCE CONFIGURATION OPTIONS CONSISTENCY

Test Label: Media Service DUT GetAudioSourceConfigurations Command and GetAudioSourceConfigurationOptions Command Consistency Validation.

Test Case ID: MEDIA-3-2-3

ONVIF Core Specification Coverage: Get audio source configurations, Get audio source configuration options

Command Under Test: GetAudioSourceConfigurations, GetAudioSourceConfigurationOptions

WSDL Reference: media.wsdl

Test Propose: To check that GetAudioSourceConfigurations command and AudioSourceConfigurationOptions command are consistent.

Pre-Requisite: Audio is supported by DUT. ONVIF Client gets the Media Service entry point by GetCapabilities command.

Test Configuration: ONVIF Client and DUT



Test Procedure:

- 1. Start an ONVIF Client.
- 2. Start the DUT.
- 3. ONVIF Client invokes GetAudioSourceConfigurationsRequest message to retrieve list of Audio Source Configurations from device.
- 4. The DUT sends GetAudioSourceConfigurationsResponse message.
- 5. ONVIF Client invokes GetAudioSourceConfigurationOptionsRequest (**ConfigurationToken**) message to retrieve available audio source configuration options from device.
- 6. The DUT sends GetAudioSourceConfigurationOptionsResponse message.
- 7. Check that InputTokensAvailable list from the GetAudioSourceConfigurationOptionsResponse message contains only unique tokens.
- 8. Check that SourceToken for the audio source configuration from GetAudioSourceConfigurationsResponse message exists in InputTokensAvailable list from the GetAudioSourceConfigurationOptionsResponse message.
- 9. Repeat steps 5-8 for each audio source configuration from the GetAudioSourceConfigurationsResponse message.

Test Result:

PASS –

DUT passes all assertions.

FAIL -

The DUT did not send GetAudioSourceConfigurationsResponse message.

The DUT did not send valid GetAudioSourceConfigurationsResponse message.

The DUT did not send GetAudioSourceConfigurationOptionsResponse message.

The DUT did not send valid GetAudioSourceConfigurationOptionsResponse message.

The DUT sent InputTokensAvailable list in the GetAudioSourceConfigurationOptionsResponse message that contains not unique tokens.

The DUT did not send current SourceToken for the audio source configuration from the GetAudioSourceConfigurationsResponse message in InputTokensAvailable list in the GetAudioSourceConfigurationOptionsResponse message.

4.6.4 PROFILES AND AUDIO SOURCE CONFIGURATION OPTIONS CONSISTENCY

Test Label: Media Service DUT GetProfiles Command and GetAudioSourceConfigurationOptions Command Consistency Validation.

Test Case ID: MEDIA-3-2-4

ONVIF Core Specification Coverage: Get media profiles, Get audio source configuration options,

Command Under Test: GetProfiles, GetAudioSourceConfigurationOptions

WSDL Reference: media.wsdl

Test Propose: To check that GetProfiles command and GetAudioSourceConfigurationOptions command are consistent.

Pre-Requisite: Audio is supported by DUT. ONVIF Client gets the Media Service entry point by GetCapabilities command.

Test Configuration: ONVIF Client and DUT



Test Procedure:

- 1. Start an ONVIF Client.
- 2. Start the DUT.
- 3. ONVIF Client invokes GetProfilesRequest message to retrieve list of existing profiles from device.
- 4. The DUT sends GetProfilesResponse message.
- 5. ONVIF Client invokes GetAudioSourceConfigurationOptionsRequest (**ProfileToken**, **ConfigurationToken**) message to retrieve available audio source configuration options for profile and audio source configuration from device.
- 6. The DUT sends GetAudioSourceConfigurationOptionsResponse message.
- 7. Check that InputTokensAvailable list from the GetAudioSourceConfigurationOptionsResponse message contains only unique tokens.
- 8. Check that SourceToken for the audio source configuration from GetProfilesResponse message exists in InputTokensAvailable list from the GetAudioSourceConfigurationOptionsResponse message.
- 9. Repeat steps 5-8 for each audio source configuration from the GetProfilesResponse message.

Test Result:

PASS –

DUT passes all assertions.

FAIL –

The DUT did not send GetProfilesResponse message.

The DUT did not send valid GetProfilesResponse message.

The DUT did not send GetAudioSourceConfigurationOptionsResponse message.

The DUT did not send valid GetAudioSourceConfigurationOptionsResponse message.

The DUT sent InputTokensAvailable list in the GetAudioSourceConfigurationOptionsResponse message that contains not unique tokens.

The DUT did not send current SourceToken for the audio source configuration from GetProfilesResponse message in InputTokensAvailable list in the GetAudioSourceConfigurationOptionsResponse message.

4.6.5 AUDIO SOURCE CONFIGURATIONS AND AUDIO SOURCES CONSISTENCY

Test Label: Media Service DUT GetAudioSourceConfigurations Command and GetAudioSources Command Consistency Validation.

Test Case ID: MEDIA-3-2-5

ONVIF Core Specification Coverage: Get audio source configurations, Get audio sources

Command Under Test: GetAudioSourceConfigurations, GetAudioSources

WSDL Reference: media.wsdl

Test Propose: To check that GetAudioSourceConfigurations command and GetAudioSources command are consistent.

Pre-Requisite: Audio is supported by DUT. ONVIF Client gets the Media Service entry point by GetCapabilities command.

Test Configuration: ONVIF Client and DUT

Test Sequence:



Test Procedure:

- 1. Start an ONVIF Client.
- 2. Start the DUT.
- 3. ONVIF Client invokes GetAudioSourceConfigurationsRequest message to retrieve list of Audio Source Configurations from device.
- 4. The DUT sends GetAudioSourceConfigurationsResponse message.
- 5. ONVIF Client invokes GetAudioSourcesRequest message to retrieve list of Audio Sources from the DUT.
- 6. The DUT sends GetAudioSourcesResponse message.
- 7. Check that each AudioSource from GetAudioSourcesResponse message has unique token.
- 8. Check that every AudioSourceConfiguration.SourceToken from GetAudioSourceConfigurationsResponse message exists in GetAudioSourcesResponse message (AudioSource.token).

-155-

Driving IP-based physical security through global standardization

Test Result:

PASS -

DUT passes all assertions.

FAIL -

The DUT did not send GetAudioSourceConfigurationsResponse message.

The DUT did not send valid GetAudioSourceConfigurationsResponse message.

The DUT did not send GetAudioSourcesResponse message.

The DUT did not send valid GetAudioSourcesResponse message.

The DUT return two or more AudioSources in GetAudioSourcesResponse message with the same VideoSourceToken.

The DUT returned the GetAudioSourceConfigurationsResponse message with AudioSources that were not included in the GetAudioSourceConfigurationsResponse message.

AUDIO SOURCE CONFIGURATION USE COUNT (CURRENT STATE) 4.6.6

Test Label: Media Service DUT Audio Source Configuration Use Count Validation.

Test Case ID: MEDIA-3-2-6

ONVIF Core Specification Coverage: Get media profiles, Get audio source configurations, Get audio source configuration

Command Under Test: GetProfiles, GetAudioSourceConfigurations, GetAudioSourceConfiguration

WSDL Reference: media.wsdl

Test Propose: To check Audio Source Configuration use count.

Pre-Requisite: Media is supported by DUT. Audio is supported by DUT. Profile with AudioSourceConfiguration exists. ONVIF Client gets the Media Service entry point by GetCapabilities command.

Test Configuration: ONVIF Client and DUT

ONVIF	Client D	JUT
Retrieve list of audio source	GetAudioSourceConfigurationsRequest	Start DUT
configurations Receive and	GetAudioSourceConfigurationsRespons (list of existing AudioSourceConfigurations)	se Send list of existing AudioSource Configurations
Retrieve list of	GetProfilesRequest	
existing profiles Receive and validate profiles	GetProfilesResponse (list of existing profiles)	Send list of existing profiles
Retrieve AudioSource	GetAudioSourceConfigurationRequest (ConfigurationToken)	
Configuration	GetAudioSourceConfigurationResponse (parameters for AudioSourceConfiguration)	e Send parameters of AudioSource Configuration
validate response	•	

Test Procedure:

- 1. Start an ONVIF Client.
- 2. Start the DUT.
- 3. ONVIF Client invokes GetAudioSourceConfigurationsRequest message to retrieve list of audio source configurations from device.
- 4. The DUT sends GetAudioSourceConfigurationsResponse message.
- 5. ONVIF Client invokes GetProfilesRequest message to retrieve list of profiles and their audio source configurations from device.
- 6. The DUT sends GetProfilesResponse message.
- 7. Check the **UseCount = usecount1** value for the first AudioSourceConfiguration (ASC1) in the list on step 3.
- 8. Check that there are not more than **usecount1** profiles with this AudioSourceConfiguration in the list from step 6.
- 9. Check that **UseCount** value in GetProfilesResponse for every occurrence of this AudioSourceConfiguration is **usecount1**.



- ONVIF Client invokes GetAudioSourceConfigurationRequest (ConfigurationToken = the first AudioSourceConfiguration from list on step 3) message to retrieve audio source configuration parameters.
- 11. The DUT sends GetAudioSourceConfigurationResponse message.
- 12. Check the UseCount value in GetAudioSourceConfigurationResponse (UseCount = usecount1).
- 13. Repeat steps 7-10 for all other AudioSourceConfigurations from the list on step 4.

Test Result:

PASS –

DUT passes all assertions.

FAIL –

The DUT did not send GetProfilesResponse message.

The DUT did not send valid GetProfilesResponse message.

The DUT did not send GetAudioSourceConfigurationsResponse message.

The DUT did not send valid GetAudioSourceConfigurationsResponse message.

The DUT did not send GetAudioSourceConfigurationResponse message.

The DUT did not send valid GetAudioSourceConfigurationResponse message.

The DUT sent UseCount value which is less than amount of profiles with AudioSourceConfiguration.

The DUT sent different UseCount values in GetProfilesResponse, GetAudioSourceConfigurationsResponse and GetAudioSourceConfigurationResponse messages.

4.6.7 AUDIO SOURCE CONFIGURATION USE COUNT (ADD SAME AUDIO SOURCE CONFIGURATION TO PROFILE TWICE)

Test Label: Media Service DUT Audio Source Configuration Use Count Validation after Adding of Same Audio Source Configuration to Profile Twice.

Test Case ID: MEDIA-3-2-12

ONVIF Core Specification Coverage: Get media profiles, Get media profile, Add audio source configuration to a profile, Get audio source configurations, Get audio source configuration

Command Under Test: GetAudioSourceConfigurations, GetAudioSourceConfiguration, AddAudioSourceConfiguration

WSDL Reference: media.wsdl

Test Propose: To check Audio Source Configuration use count after adding same audio source configuration to profile twice.

Pre-Requisite: Media is supported by DUT. Audio is supported by DUT. ONVIF Client gets the

Media Service entry point by GetCapabilities command. List of media profiles is received by GetProfiles Command

Test Configuration: ONVIF Client and DUT

ONVIF	Client	DUT
Retrieve list of	GetAudioSourceConfigurationsReque	Start DUT
ASCs Receive and validate ASCs	GetAudioSourceConfigurationsRespon (list of existing AudioSourceConfigurations) CreateProfileRequest (ProfileToken = New Profile)	→ Send list of existing AudioSource Configurations
Receive and validate response	CreateProfileResponse (empty)	Send CreateProfile Response
Add AudioSource Configuration to	AddAudioSourceConfigurationReque (ProfileToken = New Profile, ConfigurationToken = ASC1)	st ──► Send
profile Receive and validate response	AddAudioSourceConfigurationRespon (empty)	nse AddAudioSource Configuration Response
Retrieve parameters of	GetAudioSourceConfigurationReques (ConfigurationToken = ASC1)	st →
Receive and validate ASC1	GetAudioSourceConfigurationRespor (parameters for ASC1)	ISE Send parameters for ASC1
Add AudioSource Configuration to profile	AddAudioSourceConfigurationReques (ProfileToken = New Profile, ConfigurationToken = ASC1) AddAudioSourceConfigurationRespon	st Send nse AddAudioSource
Receive and validate response Retrieve parameters of ASC1	 GetAudioSourceConfigurationReques (ConfigurationToken = ASC1) 	Response



Test Procedure:

- 1. Start an ONVIF Client.
- 2. Start the DUT.
- 3. ONVIF Client invokes GetAudioSourceConfigurationsRequest message to retrieve list of audio source configurations from device.
- 4. The DUT sends GetAudioSourceConfigurationsResponse message.
- 5. ONVIF Client invokes CreateProfileRequest (**Name = NewName, Token = New Profile**) message to create profile.
- 6. If the DUT will generate a SOAP 1.2 fault message (Action/MaxNVTProfiles),

If there is profile with "fixed" attribute value "false" in profiles list (profile1)

ONVIF Client invokes RemoveAudioEncoderConfigurationRequest and RemoveAudioSourceConfigurationRequest (**ProfileToken = profile1**) message to remove AudioEncoderConfiguration and AudioSourceConfiguration from profile1

The DUT sends RemoveAudioEncoderConfigurationResponse and RemoveAudioSourceConfigurationResponse message

Execute steps 8 - 17.

If there is no profile with "fixed" attribute value "false" in profiles list, end test.

- 7. If the DUT sends CreateProfileResponse message, validate CreateProfileResponse message from the DUT.
- 8. ONVIF Client invokes AddAudioSourceConfigurationRequest (ConfigurationToken = first audio source from the list on step 4 (ASC1), ProfileToken = New Profile) message to add AudioSourceConfiguration to the new profile.
- 9. The DUT sends AddAudioSourceConfigurationResponse message.
- 10. ONVIF Client invokes GetAudioSourceConfigurationRequest (**ConfigurationToken = ASC1**) message to retrieve audio source configuration parameters.



- 11. The DUT sends GetAudioSourceConfigurationResponse message.
- 12. Check the UseCount value in GetAudioSourceConfigurationResponse message (UseCount = usecount1+1, usecount1 is value of UseCount for ASC1 from the list on step 4). If test step 6 is executed, the usecount1 could be reduced by 1.
- 13. ONVIF Client invokes AddAudioSourceConfigurationRequest (**ProfileToken = New Profile**, **ConfigurationToken = ASC1**) message to replace audio source configuration in profile.
- 14. The DUT sends AddAudioSourceConfigurationResponse message.
- 15. ONVIF Client invokes GetAudioSourceConfigurationRequest (**ConfigurationToken = ASC1**) message to retrieve audio source configuration parameters.
- 16. The DUT sends GetAudioSourceConfigurationResponse message.
- 17. Check that **UseCount = usecount1+1**, in GetAudioSourceConfigurationResponse message.
- 18. ONVIF Client invokes DeleteProfileRequest (**ProfileToken = New Profile**) message to remove profile with audio source configuration.
- 19. The DUT sends DeleteProfileResponse message.

Test Result:

PASS –

DUT passes all assertions.

FAIL –

The DUT did not send GetAudioSourceConfigurationsResponse message.

The DUT did not send valid GetAudioSourceConfigurationsResponse message.

The DUT did not send AddAudioSourceConfigurationResponse message.

The DUT did not send valid AddAudioSourceConfigurationResponse message.

The DUT did not send GetAudioSourceConfigurationResponse message.

The DUT did not send valid GetAudioSourceConfigurationResponse message.

UseCount value is not increased by 1 after adding of the AudioSourceConfiguration to one more profile.

UseCount value is changed after repeated adding of the AudioSourceConfiguration in profile.

Note: See Annex A.5 for Name and Token Parameters Length limitations.

4.6.8 AUDIO SOURCE CONFIGURATION USE COUNT (ADD DIFFERENT AUDIO SOURCE CONFIGURATIONS IN PROFILE)

Test Label: Media Service DUT Audio Source Configuration Use Count Validation after Adding Different Audio Source Configurations to the Profile.

Test Case ID: MEDIA-3-2-13



ONVIF Core Specification Coverage: Get media profiles, Get media profile, Add audio source configuration to a profile, Get audio source configurations, Get audio source configuration

Command Under Test: GetAudioSourceConfigurations, GetAudioSourceConfiguration, AddAudioSourceConfiguration

WSDL Reference: media.wsdl

Test Propose: To check Audio Source Configuration use count after adding different audio source configurations to profile.

Pre-Requisite: Media is supported by DUT. Audio is supported by DUT. There are at least two AudioSourceConfigurations. ONVIF Client gets the Media Service entry point by GetCapabilities command. List of media profiles is received by GetProfiles command.

Test Configuration: ONVIF Client and DUT

ONVIF	Client	DUT
Retrieve list of	GetAudioSourceConfigurationsRequest	Start DUT
ASCs Receive and	GetAudioSourceConfigurationsResponse (list of existing AudioSourceConfigurations)	 Send list of existing AudioSource Configurations
validate ASCs	CreateProfileRequest (ProfileToken = New Profile)	
Receive and validate response	CreateProfileResponse (empty)	Send CreateProfile Response
Add AudioSource	AddAudioSourceConfigurationRequest (ProfileToken = New Profile, ConfigurationToken = ASC1)	Sand
profile Receive and	AddAudioSourceConfigurationResponse (empty)	AddAudioSource Configuration Response
Retrieve parameters of	<pre>GetAudioSourceConfigurationRequest (ConfigurationToken = ASC1)</pre>	
ASC1 Receive and	GetAudioSourceConfigurationResponse (parameters for ASC1)	Send parameters for ASC1
Add AudioSource Configuration to profile Receive and validate response Retrieve parameters of ASC1	AddAudioSourceConfigurationRequest (ProfileToken = Profile1, ConfigurationToken = ASC2) AddAudioSourceConfigurationResponse (empty) GetAudioSourceConfigurationRequest (ConfigurationToken = ASC1)	 Send AddAudioSource Configuration Response

ONVIF	Client	DUT
Receive and	GetAudioSourceConfigurationResponse (parameters for ASC1)	Send parameters for ASC1
Retrieve parameters of	GetAudioSourceConfigurationRequest (ConfigurationToken = ASC2)	
ASC2 Receive and	GetAudioSourceConfigurationResponse (parameters for ASC2)	Send parameters for ASC2
	DeleteProfileRequest (ProfileToken = New Profile)	
Receive and validate	DeleteProfileResponse (empty)	→ Send DeleteProfile Response
Response		

Test Procedure:

- 1. Start an ONVIF Client.
- 2. Start the DUT.
- 3. ONVIF Client invokes GetAudioSourceConfigurationsRequest message to retrieve list of audio source configurations from device.
- 4. The DUT sends GetAudioSourceConfigurationsResponse message.
- 5. If there is only one AudioSourceConfiguration in the GetAudioSourceConfigurationsResponse message go to the next test.
- 6. ONVIF Client invokes CreateProfileRequest (**Name = NewName, Token = New Profile**) message to create profile.
- 7. If the DUT will generate a SOAP 1.2 fault message (Action/MaxNVTProfiles),

If there is profile with "fixed" attribute value "false" in profiles list (profile1)

ONVIF Client invokes RemoveAudioEncoderConfigurationRequest and RemoveAudioSourceConfigurationRequest (**ProfileToken = profile1**) message to remove AudioSourceConfiguration from profile1

The DUT sends RemoveAudioEncoderConfigurationResponse and RemoveAudioSourceConfigurationResponse message

Execute steps 9-21

If there is no profile with "fixed" attribute value "false" in profiles list, end test.



- 8. If the DUT sends CreateProfileResponse message, validate CreateProfileResponse message from the DUT.
- 9. ONVIF Client invokes AddAudioSourceConfigurationRequest (ConfigurationToken = first audio source from the list on step 4 (ASC1), ProfileToken = New Profile) message to add AudioSourceConfiguration to the new profile.
- 10. The DUT sends AddAudioSourceConfigurationResponse message.
- 11. ONVIF Client invokes GetAudioSourceConfigurationRequest (**ConfigurationToken = ASC1**) message to retrieve audio source configuration parameters.
- 12. The DUT sends GetAudioSourceConfigurationResponse message.
- 13. Check the UseCount value in GetAudioSourceConfigurationResponse (UseCount = usecount1+1, usecount1 is value of UseCount for ASC1 from the list on step 4). If test step 7 is executed, the usecount1 value could be reduced by 1.
- 14. ONVIF Client invokes AddAudioSourceConfigurationRequest (**ProfileToken = New Profile, ConfigurationToken = ASC2**, where ASC2 is other ASC from list on step4) message to replace audio source configuration in profile.
- 15. The DUT sends AddAudioSourceConfigurationResponse message.
- 16. ONVIF Client invokes GetAudioSourceConfigurationRequest (**ConfigurationToken = ASC1**) message to retrieve audio source configuration parameters.
- 17. The DUT sends GetAudioSourceConfigurationResponse message.
- 18. Check that **UseCount = usecount1**, in GetAudioSourceConfigurationResponse.
- 19. ONVIF Client invokes GetAudioSourceConfigurationRequest (**ConfigurationToken = ASC2**) message to retrieve audio source configuration parameters.
- 20. The DUT sends GetAudioSourceConfigurationResponse message.
- 21. Check that **UseCount = usecount2+1**, in GetAudioSourceConfigurationResponse where usecount2 is UseCount value for ASC2 in the list on step 4.
- 22. ONVIF Client invokes DeleteProfileRequest (**ProfileToken = New Profile**) message to remove profile with audio source configuration.
- 23. The DUT sends DeleteProfileResponse message.

Test Result:

PASS –

DUT passes all assertions.

FAIL –

The DUT did not send GetAudioSourceConfigurationsResponse message.

The DUT did not send valid GetAudioSourceConfigurationsResponse message.

The DUT did not send AddAudioSourceConfigurationResponse message.

The DUT did not send valid AddAudioSourceConfigurationResponse message.

The DUT did not send GetAudioSourceConfigurationResponse message.

The DUT did not send valid GetAudioSourceConfigurationResponse message.

UseCount value is not increased by 1 after adding of the AudioSourceConfiguration to one more profile.

UseCount value is not decreased by 1 after replacing of the AudioSourceConfiguration in profile for removed AudioSourceConfiguration and UseCount value is not increased by 1 after replacing of the AudioSourceConfiguration in profile for new AudioSourceConfiguration.

Note: See Annex A.5 for Name and Token Parameters Length limitations.

4.6.9 AUDIO SOURCE CONFIGURATION USE COUNT (REMOVE AUDIO SOURCE CONFIGURATION)

Test Label: Media Service DUT Audio Source Configuration Use Count Validation after Removing of Audio Source Configuration from Profile

Test Case ID: MEDIA-3-2-14

ONVIF Core Specification Coverage: Remove audio source configuration from a profile, Get audio source configurations, Get audio source configuration

Command Under Test: GetProfiles, GetAudioSourceConfigurations, GetAudioSourceConfiguration, RemoveAudioSourceConfiguration

WSDL Reference: media.wsdl

Test Propose: To check Audio Source Configuration use count after removing audio source configuration from profile.

Pre-Requisite: Media is supported by DUT. Audio is supported by DUT. ONVIF Client gets the Media Service entry point by GetCapabilities command. List of media profiles are received by GetProfiles command.

Test Configuration: ONVIF Client and DUT

ONVIF	Client	DUT
Start ONVIF Client	GetAudioSourceConfigurationsRequest	Start DUT
Retrieve list of ASCs	GetAudioSourceConfigurationsResponse (list of existing AudioSourceConfigurations)	e Send list of existing AudioSource Configurations
validate ASCs	CreateProfileRequest (ProfileToken = New Profile)	
Receive and validate response	CreateProfileResponse (empty)	Send CreateProfile Response
Add AudioSource Configuration to profile	AddAudioSourceConfigurationRequest (ProfileToken = New Profile, ConfigurationToken = ASC1)	Add audio source
Receive and validate response	(empty)	profile
Remove ASC from profile	Request (ProfileToken = New Profile) RemoveAudioSourceConfiguration	→ Remove audio source
Receive and validate response	Response (empty) GetAudioSourceConfigurationRequest	profile
Retrieve parameters of ASC1	GetAudioSourceConfigurationResponse	Send parameters
Receive and validate ASC1	DeleteProfileRequest (ProfileToken = New Profile)	
Delete Profile		
Receive and validate response	DeleteProfileResponse (empty) ◀	Delete profile
	I	I

Test Procedure:

- 1. Start an ONVIF Client.
- 2. Start the DUT.
- 3. ONVIF Client invokes GetAudioSourceConfigurationsRequest message to retrieve list of audio source configurations from device.

-168-

- 4. The DUT sends GetAudioSourceConfigurationsResponse message.
- 5. ONVIF Client invokes CreateProfileRequest (**Name = NewName, Token = New Profile**) message to create profile.
- 6. If the DUT will generate a SOAP 1.2 fault message (Action/MaxNVTProfiles),

If there is profile with "fixed" attribute value "false" in profiles list (**profile1**)

ONVIF Client invokes RemoveAudioEncoderConfigurationRequest and RemoveAudioSourceConfigurationRequest (**ProfileToken = profile1**) message to remove AudioSourceConfiguration from profile1

The DUT sends RemoveAudioEncoderConfigurationResponse and RemoveAudioSourceConfigurationResponse message

Execute steps 8-14

If there is no profile with "fixed" attribute value "false" in profiles list, end test.

- 7. If the DUT sends CreateProfileResponse message, validate CreateProfileResponse message from the DUT.
- 8. ONVIF Client invokes AddAudioSourceConfigurationRequest (ConfigurationToken = first audio source from the list on step 4 (ASC1), ProfileToken = New Profile) message to add AudioSourceConfiguration to the new profile.
- 9. The DUT sends AddAudioSourceConfigurationResponse message.
- 10. ONVIF Client invokes RemoveAudioSourceConfigurationRequest (**ProfileToken = New Profile**) message to remove audio source configuration from profile1.
- 11. The DUT sends RemoveAudioSourceConfigurationResponse message.
- 12. ONVIF Client invokes GetAudioSourceConfigurationRequest (**ConfigurationToken = ASC1**) message to retrieve audio source configuration parameters.
- 13. The DUT sends GetAudioSourceConfigurationResponse message.
- 14. Check that **UseCount = usecount1**, in GetAudioSourceConfigurationResponse **message** where usecount1 is UseCount value for ASC1 in the list on step 4. If test step 6 is executed, the usecount1 value could be reduced by 1.
- 15. ONVIF Client invokes DeleteProfileRequest (**ProfileToken = New Profile**) message to remove profile with audio source configuration.
- 16. The DUT sends DeleteProfileResponse message.

Test Result:

PASS –

ONVIF

DUT passes all assertions.

FAIL -

The DUT did not send GetAudioSourceConfigurationsResponse message.

The DUT did not send valid GetAudioSourceConfigurationsResponse message.

The DUT did not send AddAudioSourceConfigurationResponse message.

The DUT did not send valid AddAudioSourceConfigurationResponse message.

The DUT did not send RemoveAudioSourceConfigurationResponse message.

The DUT did not send valid RemoveAudioSourceConfigurationResponse message.

The DUT did not send GetAudioSourceConfigurationResponse message.

The DUT did not send valid GetAudioSourceConfigurationResponse message.

UseCount value is not decreased by 1 after removing of the AudioSourceConfiguration from profile.

Note: See Annex A.5 for Name and Token Parameters Length limitations.

4.6.10 AUDIO SOURCE CONFIGURATION USE COUNT (PROFILE DELETION WITH AUDIO SOURCE CONFIGURATION)

Test Label: Media Service DUT Audio Source Configuration Use Count Validation after Deletion of Profile with Audio Source Configuration in it.

Test Case ID: MEDIA-3-2-15

ONVIF Core Specification Coverage: Get media profiles, Get audio source configurations, Get audio source configuration. Delete media profile

GetAudioSourceConfigurations, Command Under Test: GetProfiles, GetProfiles, GetAudioSourceConfiguration, SetAudioSourceConfiguration, DeleteProfile

WSDL Reference: media.wsdl

Test Propose: To check Audio Source Configuration use count after deletion of profile with audio source configuration in it.

Pre-Requisite: Media is supported by DUT. Audio is supported by DUT. ONVIF Client gets the Media Service entry point by GetCapabilities command. List of profiles is received by GetProfiles command/ List of profiles is received by GetProfiles command.

Test Configuration: ONVIF Client and DUT

ONVIF	Client	DUT	
Retrieve list of	GetAudioSourceConfigurationsRequest		Start DUT
ASCs	GetAudioSourceConfigurationsResponse (list of existing AudioSourceConfigurations)	→ ;	Send list of existing AudioSource Configurations
Receive and validate ASCs	CreateProfileRequest (ProfileToken = New Profile)		Comgulatione
Create new profile		-	
Receive and	CreateProfileResponse (empty)		Create profile
Add AudioSource	AddAudioSourceConfigurationRequest (ProfileToken = New Profile, ConfigurationToken = ASC1)		
Profile Receive and	AddAudioSourceConfigurationResponse (empty)	→	Add audio source configuration to profile
validate response	DeleteProfileRequest (ProfileToken = New Profile)		
Delete profile		→	
Receive and	DeleteProfileResponse (empty)		Delete profile
Retrieve	GetAudioSourceConfigurationRequest (ConfigurationToken = ASC1)		
ASC1	GetAudioSourceConfigurationResponse (parameters for ASC1)	-	Send parameters for ASC1
Receive and validate response	≪		

Test Procedure:

- 1. Start an ONVIF Client.
- 2. Start the DUT.
- 3. ONVIF Client invokes GetAudioSourceConfigurationsRequest message to retrieve list of audio source configurations from device.
- 4. The DUT sends GetAudioSourceConfigurationsResponse message.



- 5. ONVIF Client invokes CreateProfileRequest (**Name = NewName, Token = New Profile**) message to create profile.
- 6. If the DUT will generate a SOAP 1.2 fault message (Action/MaxNVTProfiles),

If there is profile with "fixed" attribute value "false" in profiles list (**profile1**)

ONVIF Client invokes RemoveAudioEncoderConfigurationRequest and RemoveAudioSourceConfigurationRequest (**ProfileToken = profile1**) message to remove AudioSourceConfiguration from profile1

The DUT sends RemoveAudioEncoderConfigurationResponse and RemoveAudioSourceConfigurationResponse message

Execute steps 8-14

If there is no profile with "fixed" attribute value "false" in profiles list, end test.

- 7. If the DUT sends CreateProfileResponse message, validate CreateProfileResponse message from the DUT.
- 8. ONVIF Client invokes AddAudioSourceConfigurationRequest (ConfigurationToken = first audio source from the list on step 4 (ASC1), ProfileToken = New Profile) message to add AudioSourceConfiguration to the new profile.
- 9. The DUT sends AddAudioSourceConfigurationResponse message.
- 10. ONVIF Client invokes DeleteProfileRequest (**ProfileToken = New Profile**) message to remove profile with audio source configuration.
- 11. The DUT sends DeleteProfileResponse message.
- 12. ONVIF Client invokes GetAudioSourceConfigurationRequest (**ConfigurationToken = ASC1**) message to retrieve audio source configuration parameters.
- 13. The DUT sends GetAudioSourceConfigurationResponse message.
- 14. Check that **UseCount = usecount1**, in GetAudioSourceConfigurationResponse where **usecount1** is UseCount value for ASC1 in the list on step 4. If test step 6 is executed, the usecount1 value could be reduced by 1.

Test Result:

PASS –

DUT passes all assertions.

FAIL -

The DUT did not send GetAudioSourceConfigurationsResponse message.

The DUT did not send valid GetAudioSourceConfigurationsResponse message.

The DUT did not send AddAudioSourceConfigurationResponse message.

The DUT did not send valid AddAudioSourceConfigurationResponse message.

The DUT did not send DeleteProfileResponse message.

The DUT did not send valid DeleteProfileResponse message.



The DUT did not send GetAudioSourceConfigurationResponse message.

The DUT did not send valid GetAudioSourceConfigurationResponse message.

UseCount value is not decreased by 1 after deletion of the profile with the AudioSourceConfiguration.

Note: See Annex A.5 for Name and Token Parameters Length limitations.

4.6.11 AUDIO SOURCE CONFIGURATION USE COUNT (SET AUDIO SOURCE CONFIGURATION)

Test Label: Media Service DUT Audio Source Configuration Use Count Validation after Changing of Audio Source Configuration.

Test Case ID: MEDIA-3-2-16

ONVIF Core Specification Coverage: Get audio source configurations, Get audio source configuration, Modify an audio source configuration.

Command Under Test: GetAudioSourceConfigurations, GetAudioSourceConfiguration, SetAudioSourceConfiguration

WSDL Reference: media.wsdl

Test Propose: To check Audio Source Configuration use count after changing of audio source configuration parameters.

Pre-Requisite: Media is supported by DUT. Audio is supported by DUT. ONVIF Client gets the Media Service entry point by GetCapabilities command.

Test Configuration: ONVIF Client and DUT

ONVIF (Client [DUT
Retrieve list of	GetAudioSourceConfigurationsRequest	Start DUT
Receive and	GetAudioSourceConfigurationsResponse (list of existing AudioSourceConfigurations)	Send list of existing AudioSource Configurations
Set new UseCount	 SetAudioSourceConfigurationRequest (AudioSourceConfiguration = ASC1, UseCount = NewUseCount) 	
for ASC1 Receive and	SetAudioSourceConfigurationResponse (empty)	Set audio source configuration parameters
Retrieve parameters of	GetAudioSourceConfigurationRequest (ConfigurationToken = ASC1)	
ASC1 Receive and validate ASC1	GetAudioSourceConfigurationResponse (parameters for ASC1) ◀	Send parameters for ASC1

Test Procedure:

- 1. Start an ONVIF Client.
- 2. Start the DUT.
- 3. ONVIF Client invokes GetAudioSourceConfigurationsRequest message to retrieve list of audio source configurations from device.
- 4. The DUT sends GetAudioSourceConfigurationsResponse message.
- ONVIF Client invokes SetAudioSourceConfigurationRequest (token = the first Audio Source Configuration from the list on step 4(ASC1), UseCount = NewUseCount) message to set parameters for audio source configuration.
- 6. The DUT sends SetAudioSourceConfigurationResponse message.
- 7. ONVIF Client invokes GetAudioSourceConfigurationRequest (**ConfigurationToken = ASC1**) message to retrieve audio source configuration parameters.
- 8. The DUT sends GetAudioSourceConfigurationResponse message.
- 9. Check that **UseCount = usecount1**, in GetAudioSourceConfigurationResponse where usecount1 is UseCount value for ASC1 in the list on step 4.

Test Result:

-174-

PASS –

DUT passes all assertions.

FAIL -

The DUT did not send GetAudioSourceConfigurationsResponse message.

The DUT did not send valid GetAudioSourceConfigurationsResponse message.

The DUT did not send SetAudioSourceConfigurationResponse message.

The DUT did not send valid SetAudioSourceConfigurationResponse message.

The DUT did not send GetAudioSourceConfigurationResponse message.

The DUT did not send valid GetAudioSourceConfigurationResponse message.

UseCount value was changed after trying to set UseCount value.

Note: See Annex A.5 for Name and Token Parameters Length limitations.

4.7 Audio Encoder Configuration

4.7.1 AUDIO ENCODER CONFIGURATIONS AND PROFILES CONSISTENCY

Test Label: Media Service DUT Audio Encoder Configurations and Profiles Consistency Validation.

-175-

Test Case ID: MEDIA-3-3-1

ONVIF Core Specification Coverage: Get media profiles, Get audio encoder configurations

Command Under Test: GetProfiles, GetAudioEncoderConfigurations

WSDL Reference: media.wsdl

Test Propose: To check that Audio Encoder Configurations and Audio Encoder Profiles are consistent.

Pre-Requisite: Audio is supported by DUT. ONVIF Client gets the Media Service entry point by GetCapabilities command.

Test Configuration: ONVIF Client and DUT

Test Sequence:



Test Procedure:

- 1. Start an ONVIF Client.
- 2. Start the DUT.
- 3. ONVIF Client invokes GetProfilesRequest message to retrieve list of profiles and their audio encoder configurations.
- 4. The DUT sends GetProfilesResponse message.



- 5. ONVIF Client invokes GetAudioEncoderConfigurationsRequest message to retrieve list of Audio Source Configurations from device.
- 6. The DUT sends GetAudioEncoderConfigurationsResponse message.
- 7. Check each GetAudioEncoderConfiguration from that GetAudioEncoderConfigurationsResponse message has unique token.
- 8. Check that each AudioEncoderConfiguration from the GetProfilesResponse message are included in the GetAudioEncoderConfigurationsResponse message.
- 9. Check that AudioEncoderConfiguration parameters are same in the GetProfilesResponse message and in the GetAudioEncoderConfigurationsResponse message for corresponding AudioEncoderConfiguration.

Test Result:

PASS -

DUT passes all assertions.

FAIL -

The DUT did not send GetProfilesResponse message.

The DUT did not send valid GetProfilesResponse message.

The DUT did not send GetAudioEncoderConfigurationsResponse message.

The DUT did not send valid GetAudioEncoderConfigurationsResponse message.

The DUT return AudioEncoderConfigurations two or more in GetAudioEncoderConfigurationsResponse message with the same ConfigurationToken.

The DUT returned the GetProfilesResponse message with AudioEncoderConfigurations that were not included in the GetAudioEncoderConfigurationsResponse message.

The DUT returned different parameters list and parameters values in the GetAudioEncoderConfigurationsResponse message and in the GetProfilesResponse message for the same AudioEncoderConfiguration.

AUDIO ENCODER CONFIGURATIONS AND AUDIO ENCODER CONFIGURATION 4.7.2 CONSISTENCY

Test Label: Media Service DUT GetAudioEncoderConfigurations Command and GetAudioEncoderConfiguration Command Consistency Validation.

Test Case ID: MEDIA-3-3-2

ONVIF Core Specification Coverage: Get audio encoder configurations, Get audio encoder configuration

Command Under Test: GetAudioEncoderConfigurations, GetAudioEncoderConfiguration

WSDL Reference: media.wsdl

GetAudioEncoderConfigurations Test Propose: То check that command and GetAudioEncoderConfiguration command are consistent.

Pre-Requisite: Audio is supported by DUT. ONVIF Client gets the Media Service entry point by GetCapabilities command.

-177-

Test Configuration: ONVIF Client and DUT

Test Sequence:

ONVIF	Client	DUT	
Retrieve list of	GetAudioEncoderConfigurationsReques	t	Start DUT
Receive and validate response	GetAudioEncoderConfigurations Response (list of existing AudioEncoderConfigurations)		Send list of existing AudioEncoder Configurations
Retrieve parameters for	GetAudioEncoderConfigurationRequest (ConfigurationToken)		
configuration	GetAudioEncoderConfigurationRespons (parameters of audio encoder	e	Send parameters of audio encoder configuration
Receive and validate response	<		

Test Procedure:

- 1. Start an ONVIF Client.
- 2. Start the DUT.
- 3. ONVIF Client invokes GetAudioEncoderConfigurationsRequest message to retrieve list of Audio Source Configurations from device.
- 4. The DUT sends GetAudioEncoderConfigurationsResponse message.
- 5. ONVIF Client invokes GetAudioEncoderConfigurationRequest (**ConfigurationToken**) message to retrieve parameters of audio encoder configuration from the DUT.
- 6. The DUT sends GetAudioEncoderConfigurationResponse message.
- 7. Verify that all parameters for audio encoder configuration from the GetAudioEncoderConfigurationsResponse message and from the GetAudioEncoderConfigurationResponse message are the same.
- 8. Repeat steps 5-7 for each Audio Encoder Configurations from the GetAudioEncoderConfigurationsResponse message.

Test Result:

PASS -

DUT passes all assertions.

FAIL -

The DUT did not send GetAudioEncoderConfigurationsResponse message.

The DUT did not send valid GetAudioEncoderConfigurationsResponse message.

The DUT did not send GetAudioEncoderConfigurationResponse message.

The DUT did not send valid GetAudioEncoderConfigurationResponse message.

The DUT did not send equal parameters for one or more AudioEncoderConfiguration in the GetAudioEncoderConfigurationResponse message and in the GetAudioEncoderConfigurationsResponse message.

4.7.3 AUDIO ENCODER CONFIGURATIONS AND AUDIO ENCODER CONFIGURATION OPTIONS CONSISTENCY

Test Label: Media Service DUT GetAudioEncoderConfigurations Command and GetAudioEncoderConfigurationOptions Command Consistency Validation.

Test Case ID: MEDIA-3-3-3

ONVIF Core Specification Coverage: Get audio encoder configurations, Get audio encoder configuration options

Command Under Test: GetAudioEncoderConfigurations, GetAudioEncoderConfigurationOptions

WSDL Reference: media.wsdl

Test Propose: To check that GetAudioEncoderConfiguration command and GetAudioEncoderConfigurationOptions command are consistent.

Pre-Requisite: Audio is supported by DUT. ONVIF Client gets the Media Service entry point by GetCapabilities command.

Test Configuration: ONVIF Client and DUT



Test Procedure:

- 1. Start an ONVIF Client.
- 2. Start the DUT.
- 3. ONVIF Client invokes GetAudioEncoderConfigurationsRequest message to retrieve list of Audio Encoder Configurations from device.
- 4. The DUT sends GetAudioEncoderConfigurationsResponse message.
- 5. ONVIF Client invokes GetAudioEncoderConfigurationOptionsRequest (**ConfigurationToken**) message to retrieve available audio encoder configuration options from device.
- 6. The DUT sends GetAudioEncoderConfigurationOptionsResponse message.
- 7. Check that Encoding value for audio encoder configuration exists in Encoding parameter of one of the Option sections from list in the GetAudioEncoderConfigurationOptionsResponse message, and Bitrate and SampleRate exist in BitrateList and SampleRateList for this encoding in the GetAudioEncoderConfigurationOptionsResponse message.
- 8. Repeat steps 5-7 for each audio encoder configuration from the GetAudioEncoderConfigurationsResponse message.

Test Result:

PASS –

DUT passes all assertions.

FAIL -

The DUT did not send GetAudioEncoderConfigurationsResponse message.



The DUT did not send valid GetAudioEncoderConfigurationsResponse message.

The DUT did not send GetAudioEncoderConfigurationOptionsResponse message.

The DUT did not send valid GetAudioEncoderConfigurationOptionsResponse message.

The DUT sent non-available Encoding value for one or more AudioSourceConfigurations

The DUT sent not available combination of Encoding and Bitrate or Encoding and SampleRate for one or more AudioEncoderConfigurations.

4.7.4 PROFILES AND AUDIO ENCODER CONFIGURATION OPTIONS CONSISTENCY

Test Label: Media Service DUT GetProfiles Command and GetAudioEncoderConfigurationOptions Consistency Validation.

Test Case ID: MEDIA-3-3-4

ONVIF Core Specification Coverage: Get media profile, Get audio encoder configuration options

Command Under Test: GetProfiles, GetAudioEncoderConfigurationOptions

WSDL Reference: media.wsdl

Test Propose: To check that GetProfiles command and GetAudioEncoderConfigurationOptions command are consistency.

Pre-Requisite: Audio is supported by DUT. ONVIF Client gets the Media Service entry point by GetCapabilities command.

Test Configuration: ONVIF Client and DUT

Client	DUT	
GetProfilesRequest		Start DUT
GetProfilesResponse (list of existing profiles)		Send list of existing profiles
 GetAudioEncoderConfigurationOptions Request (ProfileToken, AudioEncoderConfigurationToken) 		
GetAudioEncoderConfigurationOptions Response (options)		Send options for AudioEncoder Configuration
	Client GetProfilesRequest GetProfilesResponse (list of existing profiles) GetAudioEncoderConfigurationOptions Request (ProfileToken, AudioEncoderConfigurationToken) GetAudioEncoderConfigurationOptions Response (options)	Client DUT GetProfilesRequest GetProfilesResponse (list of existing profiles) GetAudioEncoderConfigurationOptions Request (ProfileToken, AudioEncoderConfigurationToken) GetAudioEncoderConfigurationOptions Response (options)
-181-

Onvir Driving IP-based physical security through global standardization

Test Procedure:

- 1. Start an ONVIF Client.
- 2. Start the DUT.
- 3. ONVIF Client invokes GetProfilesRequest message to retrieve list of profiles from device.
- 4. The DUT sends GetProfilesResponse message.
- 5. ONVIF Client invokes GetAudioEncoderConfigurationOptionsRequest (**ProfileToken**, **ConfigurationToken**) message to retrieve AudioEncoderConfigurationOptions for audio encoder configuration and profile from the DUT.
- 6. The DUT sends GetAudioEncoderConfigurationOptionsResponse message.
- 7. Check that Encoding value for audio encoder configuration exists in Encoding parameter of one of the Option sections from list in the GetAudioEncoderConfigurationOptionsResponse message, and Bitrate and SampleRate exist in BitrateList and SampleRateList for this encoding in the GetAudioEncoderConfigurationOptionsResponse message.
- 8. Repeat steps 5-7 for each audio encoder configuration from the GetProfilesResponse message.

Test Result:

PASS –

DUT passes all assertions.

FAIL –

The DUT did not send GetProfilesResponse message.

The DUT did not send valid GetProfilesResponse message.

The DUT did not send GetAudioEncoderConfigurationOptionsResponse message.

The DUT did not send valid GetAudioEncoderConfigurationOptionsResponse message.

The DUT sent non-available Encoding value for one or more AudioSourceConfigurations

The DUT sent not available combination of Encoding and Bitrate or Encoding and SampleRate for one or more AudioEncoderConfigurations.

4.7.5 AUDIO ENCODER CONFIGURATION USE COUNT (CURRENT STATE)

Test Label: Media Service DUT Audio Encoder Configuration Use Count Validation.

Test Case ID: MEDIA-3-3-5

ONVIF Core Specification Coverage: Get media profiles, Get audio encoder configurations, Get audio encoder configuration

Command	Under	Test:	GetProfiles,	GetAudioEncoderConfigurations,
GetAudioEncoder	Configuration			

WSDL Reference: media.wsdl

Test Propose: To check Audio Encoder Configuration use count.

Pre-Requisite: Media is supported by DUT. Audio is supported by DUT. ONVIF Client gets the Media Service entry point by GetCapabilities command.

Test Configuration: ONVIF Client and DUT

Test Sequence:

ONVIF Client		DUT	
Retrieve list of	GetAudioEncoderConfigurationsReques	st	Start DUT
AECs Receive and validate AECs	GetAudioEncoderConfigurations Response (list of existing AudioEncoderConfigurations)	→	Send list of existing AudioEncoder Configurations
Retrieve list of	GetProfileRequest		Send list of
Receive and	GetProfilesResponse (list of existing profiles)		existing profiles
Retrieve AudioEncoder	GetAudioEncoderConfigurationRequest (AudioEncoderConfiguration)		
Configuration Receive and validate parameters	GetAudioEncoderConfigurationRespons (parameters for AudioEncoderConfiguration)	ie	Send parameters of AudioEncoder Configuration

Test Procedure:

- 1. Start an ONVIF Client.
- 2. Start the DUT.
- 3. ONVIF Client invokes GetAudioEncoderConfigurationsRequest message to retrieve list of audio encoder configurations from device.
- 4. The DUT sends GetAudioEncoderConfigurationsResponse message.
- 5. ONVIF Client invokes GetProfilesRequest message to retrieve list of profiles and their audio encoder configurations from device.
- 6. The DUT sends GetProfilesResponse message.
- 7. Check the **UseCount = usecount1** value for the first AudioEncoderConfiguration (AEC1) in the list on step 3.



- 8. Check that there are not more than **usecount1** profiles with this AudioEncoderConfiguration in the list from step 6.
- 9. Check that **UseCount** value in GetProfilesResponse message for every occurrence of this AudioEncoderConfiguration is **usecount1**.
- 10. ONVIF Client invokes GetAudioEncoderConfigurationRequest (**ConfigurationToken = AEC1**) message to retrieve audio encoder configuration parameters.
- 11. The DUT sends GetAudioEncoderConfigurationResponse message.
- 12. Check the UseCount value in GetAudioEncoderConfigurationResponse (UseCount = usecount1) message.
- 13. Repeat steps 7-10 for all other AudioEncoderConfigurations from the list on step 4.

Test Result:

PASS –

DUT passes all assertions.

FAIL -

The DUT did not send GetProfilesResponse message.

The DUT did not send valid GetProfilesResponse message.

The DUT did not send GetAudioEncoderConfigurationsResponse message.

The DUT did not send valid GetAudioEncoderConfigurationsResponse message.

The DUT did not send GetAudioEncoderConfigurationResponse message.

The DUT did not send valid GetAudioEncoderConfigurationResponse message.

The DUT sent UseCount value which is less than amount of profiles with AudioEncoderConfiguration.

The DUT sent different UseCount values in GetProfilesResponse, GetAudioEncoderConfigurationsResponse and GetAudioEncoderConfigurationResponse messages.

4.7.6 AUDIO ENCODER CONFIGURATION OPTIONS AND AUDIO ENCODER CONFIGURATIONS CONSISTENCY (BITRATE AND SAMPLERATE)

Test Label: Validation range of bitrate and samplerate in AudioEncoderConfigurationOptions

Test Case ID: MEDIA-3-3-11

ONVIF Core Specification Coverage: Get audio encoder configuration options

Command Under Test: GetAudioEncoderConfigurationOptions

WSDL Reference: media.wsdl

Test Purpose: To that verify bitrate and samplerate values (the output bitrate [kbps], the output sample rate [kHz]) in the GetAudioEncoderConfigurations command are consistent with options and real settings.

Pre-Requisite: Media is supported by DUT. Audio is supported by DUT. Media Service was received from the DUT.

Test Configuration: ONVIF Client and DUT

Test Sequence:

ONVIF Client DUT Start DUT GetAudioEncoderConfiguration OptionsRequest Send list of Audio GetAudioEncoderConfiguration Receive and Validate **Encoder Configuration** OptionsResponse GetAudioEncoder Options (Options.BitrateList.Items, ConfigurationOptions Options.SampleRateList.Items) Response message GetAudioEncoderConfigurations Request Send list of Audio GetAudioEncoderConfigurationsRe **Encoder Configurations** sponse Receive and Validate GetAudioEncoder Configurations GetAudioEncoderConfiguration Response message OptionsRequest (audio encoder configuration token) GetAudioEncoderConfiguration Send audio encoder OptionsResponse configuration options for Receive and Validate (Options.BitrateList.Item, specified AEC GetAudioEncoder Options.SampleRateList.Item) ConfigurationOptions Response message

Test Procedure:

- 1. Start an ONVIF Client
- 2. Start the DUT
- 3. ONVIF Client invokes GetAudioEncoderConfigurationOptionsRequest message to retrieve list of Audio Encoder Configuration Options from device.
- 4. Verify the GetAudioEncoderConfigurationOptionsResponse message from the DUT. Check that all values of Options.BitrateList.Items are less than 10000. Check that all values of Options.SampleRateList.Items are less than 1000.
- 5. ONVIF Client invokes GetAudioEncoderConfigurationsRequest message to retrieve list of Audio Encoder Configurations from device.



- 6. Verify the GetAudioEncoderConfigurationsResponse message from the DUT.
- 7. ONVIF Client invokes GetAudioEncoderConfigurationOptionsRequest message (ConfigurationToken = 'token1' where "Token1" is a first audio encoder configuration token from GetAudioEncoderConfigurationsResponse message) to retrieve list of Audio Encoder Configuration Options for specified Audio Encoder Configuration from device.
- 8. Verify the GetAudioEncoderConfigurationOptionsResponse message from the DUT. Check that all values of Options.BitrateList.Items are less than 10000 and listed in Options.BitrateList of GetAudioEncoderConfigurationOptionsResponse message from step 4. Check that all values of Options.SampleRateList.Items are less than 1000 listed in Options.SampleRateList of GetAudioEncoderConfigurationOptionsResponse message from step 4.
- 9. Repeat steps 7-8 for the rest Audio Encoder Configurations supported by the DUT.

Test Result:

PASS –

The DUT passes all assertions.

FAIL –

The DUT did not send valid GetAudioEncoderConfigurationOptionsResponse message.

The DUT did not send valid GetAudioEncoderConfigurationsResponse message.

The DUT sent at least one Options.BitrateList.Item that is greater than 10000.

The DUT sent at least one Options.SampleRateList.Item that is greater than 1000.

The DUT sent at least one Options.BitrateList.Item in GetAudioEncoderConfigurationOptionsResponse message for specific Audio Encoder Configuration that is not listed in general GetAudioEncoderConfigurationOptionsResponse message.

The DUT sent at least one Options.SampleRateList.Item in GetAudioEncoderConfigurationOptionsResponse message for specific Audio Encoder Configuration that is not listed in general GetAudioEncoderConfigurationOptionsResponse message.

Note: See Annex A.5 for Name and Token Parameters Length limitations.

Note: Bitrate shall be specified in kbps, so it shall not exceed 10000.

Note: SampleRate shall be specified in kHz, so it shall not exceed 1000.

4.7.7 AUDIO ENCODER CONFIGURATION USE COUNT (ADD SAME AUDIO ENCODER CONFIGURATION TO PROFILE TWICE)

Test Label: Media Service DUT Audio Encoder Configuration Use Count Validation after Adding of Same Audio Encoder Configuration to Profile Twice.

Test Case ID: MEDIA-3-3-12

ONVIF Core Specification Coverage: Get media profiles, Get media profile, Add audio encoder configuration to a profile, Get audio encoder configurations, Get audio encoder configuration



CommandUnderTest:GetProfiles,GetAudioEncoderConfigurations,GetAudioEncoderConfiguration, AddAudioEncoderConfigurationGetAudioEncoderConfigurationGetAudioEncoderConfiguration

WSDL Reference: media.wsdl

Test Propose: To check Audio Encoder Configuration use count after adding same audio encoder configuration to profile twice.

Pre-Requisite: Media is supported by DUT. Audio is supported by DUT. ONVIF Client gets the Media Service entry point by GetCapabilities command. List of media profiles is received by GetProfiles command.

Test Configuration: ONVIF Client and DUT

С	NVI	FΟ	Clie	nt	
		1			

DUT

	GetAudioEncoderConfigurationsRequest	Start DUT
Retrieve list of AECs	GetAudioEncoderConfigurations	Send list of
Receive and validate AECs	Response (list of existing AudioEncoderConfigurations)	AudioEncoder Configurations
Retrieve list of	GetAudioSourceConfigurationsRequest	
ASCs Receive and validate ASCs	GetAudioSourceConfigurationsResponse (list of AudioSourceConfigurations)	Send list of existing AudioSource
	CreateProfileRequest (ProfileToken = New Profile)	Configurations
Create new profile		
Receive and	CreateProfileResponse (empty)	Send CreateProfile Response
Get compatible audio source	GetCompatibleAudioSource ConfigurationsRequest (ProfileToken = New Profile)	Send Compatible
Receive and validate response	GetCompatibleAudioSource ConfigurationsResponse (empty)	AudioSource Configurations Response
Add ASC1 to New Profile	AddAudioSourceConfigurationRequest (ProfileToken = New Profile, ConfigurationToken = ASC1)	Sond
Receive and validate response	AddAudioSourceConfigurationResponse (empty)	AddAudioEncoder Configuration Response
Get compatible audio encoder	GetCompatibleAudioEncoder ConfigurationsRequest (ProfileToken = New Profile)	Send Compatible
Receive and validate response	GetCompatibleAudioEncoder ConfigurationsResponse (empty)	AudioEncoder Configurations Response

ONVIF Client		DUT
Add AudioEncoder Configuration to profile	AddAudioEncoderConfigurationRequest (ProfileToken = New Profile, ConfigurationToken = AEC1)	Send
Receive and validate response	AddAudioEncoderConfigurationRespons (empty)	e AddAudioEncoder Configuration Response
Retrieve parameters of	GetAudioEncoderConfigurationRequest (ConfigurationToken = AEC1)	-
AEC1 Receive and validate AEC1	GetAudioEncoderConfigurationRespons (parameters for AEC1)	e Send parameters for AEC1
Add AudioEncoder Configuration to	AddAudioEncoderConfigurationRequest (ProfileToken = New Profile, ConfigurationToken = AEC1)	Send
profile Receive and	AddAudioEncoderConfigurationRespons (empty)	e AddAudioEncoder Configuration Response
Retrieve parameters of	GetAudioEncoderConfigurationRequest (ConfigurationToken = AEC1)	-
AEC1 Receive and validate AEC1	GetAudioEncoderConfigurationRespons (parameters for AEC1)	e Send parameters for AEC1

Test Procedure:

- 1. Start an ONVIF Client.
- 2. Start the DUT.
- 3. ONVIF Client invokes GetAudioEncoderConfigurationsRequest message to retrieve list of audio encoder configurations from device.
- 4. The DUT sends GetAudioEncoderConfigurationsResponse message.
- 5. ONVIF Client invokes GetAudioSourceConfigurationsRequest message to retrieve list of audio encoder configurations from device.
- 6. The DUT sends GetAudioSourceConfigurationsResponse message.



- 7. ONVIF Client invokes CreateProfileRequest (Name = NewName, Token = New Profile) message to create profile.
- 8. If the DUT will generate a SOAP 1.2 fault message (Action/MaxNVTProfiles),

If there is profile with "fixed" attribute value "false" in profiles list (**profile1**)

ONVIF Client invokes RemoveAudioEncoderConfigurationRequest (**ProfileToken = profile1**) message to remove AudioEncoderConfiguration from profile1

The DUT sends RemoveAudioEncoderConfigurationResponse message

Execute steps 10-26

If there is no profile with "fixed" attribute value "false" in profiles list, end test.

- 9. If the DUT sends CreateProfileResponse message, validate CreateProfileResponse message from the DUT.
- 10. ONVIF Client invokes GetCompatibleAudioSourceConfigurationsRequest (**ProfileToken = New Profile**) message to get compatible AudioSourceConfigurations for the new profile.
- 11. The DUT sends GetCompatibleAudioSourceConfigurationsResponse message.
- ONVIF Client invokes AddAudioSourceConfigurationRequest (ConfigurationToken = first audio source configuration from the list on step 11 (ASC1), ProfileToken = New Profile) message to add AudioSourceConfiguration to the new profile.
- 13. The DUT sends AddAudioSourceConfigurationResponse message.
- 14. ONVIF Client invokes GetCompatibleAudioEncoderConfigurationsRequest (ProfileToken = New Profile) message to get compatible AudioEncoderConfigurations for the new profile with Audio Source Configuration.
- 15. The DUT sends GetCompatibleAudioEncoderConfigurationsResponse message. If response does not contain any AudioEncoderConfiguration, repeat steps 12-15 with next audio source configuration from step 11. If there are no other audio source configuration, skip other steps and go to the next test.
- 16. ONVIF Client invokes AddAudioEncoderConfigurationRequest (ConfigurationToken = first audio encoder from the list on step 15 (AEC1), ProfileToken = New Profile) message to add AudioEncoderConfiguration to the new profile.
- 17. The DUT sends AddAudioEncoderConfigurationResponse message.
- 18. ONVIF Client invokes GetAudioEncoderConfigurationRequest (**ConfigurationToken = AEC1**) message to retrieve audio encoder configuration parameters.
- 19. The DUT sends GetAudioEncoderConfigurationResponse message.
- 20. Check the UseCount value in GetAudioEncoderConfigurationResponse message (UseCount = usecount1+1, usecount1 is value of UseCount for AEC1 from the list on step 4). If test step 8 is executed, the usecount1 value could be reduced by 1.
- 21. ONVIF Client invokes AddAudioEncoderConfigurationRequest (**ProfileToken = New Profile**, **ConfigurationToken = AEC1**) message to replace audio encoder configuration in profile.
- 22. The DUT sends AddAudioEncoderConfigurationResponse message.

- 23. ONVIF Client invokes GetAudioEncoderConfigurationRequest (**ConfigurationToken = AEC1**) message to retrieve audio encoder configuration parameters.
- 24. The DUT sends GetAudioEncoderConfigurationResponse message.
- 25. Check that **UseCount = usecount1+1**, in GetAudioEncoderConfigurationResponse message.
- 26. ONVIF Client restores DUT configuration.

Test Result:

PASS –

DUT passes all assertions.

FAIL –

The DUT did not send valid GetAudioEncoderConfigurationsResponse message.

 $\label{eq:compatibleAudioEncoderConfigurationsResponse} The DUT did not send valid GetCompatibleAudioEncoderConfigurationsResponse message.$

The DUT did not send valid GetCompatibleAudioSourceConfigurationsResponse message.

The DUT did not send valid AddAudioEncoderConfigurationResponse message.

The DUT did not send valid GetAudioEncoderConfigurationResponse message.

UseCount value is not increased by 1 after adding of the AudioEncoderConfiguration to one more profile.

UseCount value is changed after repeated adding of the AudioEncoderConfiguration in profile.

Note: See Annex A.5 for Name and Token Parameters Length limitations.

4.7.8 AUDIO ENCODER CONFIGURATION USE COUNT (ADD DIFFERENT AUDIO ENCODER CONFIGURATIONS IN PROFILE)

Test Label: Media Service DUT Audio Encoder Configuration Use Count Validation after Adding Different Audio Encoder Configurations to the Profile.

Test Case ID: MEDIA-3-3-13

ONVIF Core Specification Coverage: Get media profiles, Get media profile, Add audio encoder configuration to a profile, Get audio encoder configurations, Get audio encoder configuration

CommandUnderTest:GetProfiles,GetAudioEncoderConfigurations,GetAudioEncoderConfiguration, AddAudioEncoderConfigurationGetAudioEncoderConfigurationGetAudioEncoderConfiguration

WSDL Reference: media.wsdl

Test Propose: To check Audio Encoder Configuration use count after adding different audio encoder configurations to profile.

Pre-Requisite: Media is supported by DUT. Audio is supported by DUT. There are at least two AudioEncoderConfigurations. ONVIF Client gets the Media Service entry point by GetCapabilities command. List of media profiles is received by GetProfiles command

Test Configuration: ONVIF Client and DUT

DUT

	CotAudioEncodorConfigurationsPoguest	Start DUT
Retrieve list of AECs		Send list of
Receive and validate AECs	GetAudioEncoderConfigurations Response (list of existing AudioEncoderConfigurations)	existing AudioEncoder Configurations
Retrieve list of	GetAudioSourceConfigurationsRequest	
Receive and validate ASCs	GetAudioSourceConfigurationsResponse (list of AudioSourceConfigurations)	Send list of existing AudioSource
	CreateProfileRequest (ProfileToken = New Profile)	Configurations
Create new profile		
Receive and validate response	CreateProfileResponse (empty)	Send CreateProfile Response
Get compatible audio source configurations	GetCompatibleAudioSource ConfigurationsRequest (ProfileToken = New Profile)	Send Compatible
Receive and validate response	GetCompatibleAudioSource ConfigurationsResponse (empty)	AudioSource Configurations Response
Add ASC1 to New Profile	AddAudioSourceConfigurationRequest (ProfileToken = New Profile, ConfigurationToken = ASC1)	Send
Receive and validate response	AddAudioSourceConfigurationResponse (empty)	AddAudioEncoder Configuration Response
Get compatible audio encoder configurations	GetCompatibleAudioEncoder ConfigurationsRequest (ProfileToken = New Profile)	Send Compatible
Receive and validate response	GetCompatibleAudioEncoder ConfigurationsResponse (empty)	AudioEncoder Configurations Response

ONVIF	DUT	
Add AudioEncoder	AddAudioEncoderConfigurationRequest (ProfileToken = New Profile, ConfigurationToken = AEC1)	
profile Receive and	AddAudioEncoderConfigurationRespons (empty)	e Add AudioEncoderConf iguration to profile
validate response	GetAudioEncoderConfigurationRequest (ConfigurationToken = AEC1)	
Retrieve parameters of AEC1	GetAudioEncoderConfigurationResponse	Send parameters
Receive and validate AEC1	(parameters for AEC1)	for AEC1
	AddAudioEncoderConfigurationRequest (ProfileToken = Profile1, ConfigurationToken = AEC2)	
Add AudioEncoder Configuration to profile		→
Receive and	AddAudioEncoderConfigurationRespons (empty)	e AddAudioEncoder Configuration to profile
validate response	GetAudioEncoderConfigurationRequest (ConfigurationToken = AEC1)	_
parameters of AEC1	GetAudioEncoderConfigurationRespons	→ e Send parameters
Receive and validate AEC1	(parameters for AEC1)	for AEC1
Retrieve	GetAudioEncoderConfigurationRequest (ConfigurationToken = AEC2)	
AEC2		→
Receive and	GetAudioEncoderConfigurationRespons (parameters for AEC2)	e Send parameters for AEC2
validate AEC2	•	

Test Procedure:

- 1. Start an ONVIF Client.
- 2. Start the DUT.



- 3. ONVIF Client invokes GetAudioEncoderConfigurationsRequest message to retrieve list of audio encoder configurations from device.
- 4. The DUT sends GetAudioEncoderConfigurationsResponse message.
- 5. If there is only one AudioEncoderConfiguration in the GetAudioEncoderConfigurationsResponse message go to the next test.
- 6. ONVIF Client invokes GetAudioSourceConfigurationsRequest message to retrieve list of audio source configurations from device.
- 7. The DUT sends GetAudioSourceConfigurationsResponse message.
- 8. ONVIF Client invokes CreateProfileRequest (Name = NewName, Token = New Profile) message to create profile.
- 9. If the DUT will generate a SOAP 1.2 fault message (Action/MaxNVTProfiles),

If there is profile with "fixed" attribute value "false" in profiles list (**profile1**)

ONVIF Client invokes RemoveAudioEncoderConfigurationRequest (**ProfileToken = profile1**) message to remove AudioEncoderConfiguration from profile1

The DUT sends RemoveAudioEncoderConfigurationResponse message

Execute steps 11-30

If there is no profile with "fixed" attribute value "false" in profiles list, end test.

- 10. If the DUT sends CreateProfileResponse message, validate CreateProfileResponse message from the DUT.
- 11. ONVIF Client invokes GetCompatibleAudioSourceConfigurationsRequest (**ProfileToken = New Profile**) message to get compatible AudioSourceConfigurations for the new profile.
- 12. The DUT sends GetCompatibleAudioSourceConfigurationsResponse message.
- ONVIF Client invokes AddAudioSourceConfigurationRequest (ConfigurationToken = first audio source configuration from the list on step 11 (ASC1), ProfileToken = New Profile) message to add AudioSourceConfiguration to the new profile.
- 14. The DUT sends AddAudioSourceConfigurationResponse message.
- 15. ONVIF Client invokes GetCompatibleAudioEncoderConfigurationsRequest (ProfileToken = New Profile) message to get compatible AudioEncoderConfigurations for the new profile with Audio Source Configuration.
- 16. The DUT sends GetCompatibleAudioEncoderConfigurationsResponse message. If response does not contain at least two AudioEncoderConfiguration, repeat steps 12-15 with next audio source configuration from step 11. If there are no other audio source configuration, skip other steps and go to the next test.
- 17. ONVIF Client invokes AddAudioEncoderConfigurationRequest (ConfigurationToken = first audio encoder from the list on step 15 (AEC1), ProfileToken = New Profile) message to add AudioEncoderConfiguration to the new profile.
- 18. The DUT sends AddAudioEncoderConfigurationResponse message.
- ONVIF Client invokes GetAudioEncoderConfigurationRequest (ConfigurationToken = AEC1) message to retrieve audio encoder configuration parameters.



- 20. The DUT sends GetAudioEncoderConfigurationResponse message.
- 21. Check the UseCount value in GetAudioEncoderConfigurationResponse (UseCount = usecount1+1, usecount1 is value of UseCount for AEC1 from the list on step 4). If test step 9 is executed, the usecount1 value could be reduced by 1.
- 22. ONVIF Client invokes AddAudioEncoderConfigurationRequest (**ProfileToken = New Profile**, **ConfigurationToken = AEC2**, where AEC2 is other AEC from list on step4) message to replace audio encoder configuration in profile.
- 23. The DUT sends AddAudioEncoderConfigurationResponse message.
- 24. ONVIF Client invokes GetAudioEncoderConfigurationRequest (**ConfigurationToken = AEC1**) message to retrieve audio encoder configuration parameters.
- 25. The DUT sends GetAudioEncoderConfigurationResponse message.
- 26. Check that **UseCount = usecount1**, in GetAudioEncoderConfigurationResponse.
- 27. ONVIF Client invokes GetAudioEncoderConfigurationRequest (**ConfigurationToken = AEC2**) message to retrieve audio encoder configuration parameters.
- 28. The DUT sends GetAudioEncoderConfigurationResponse message.
- 29. Check that **UseCount = usecount2+1**, in GetAudioEncoderConfigurationResponse where usecount2 is UseCount value for AEC2 in the list on step 4.
- 30. ONVIF Client restores DUT configuration.

Test Result:

PASS –

DUT passes all assertions.

FAIL -

The DUT did not send valid GetAudioEncoderConfigurationsResponse message.

The DUT did not send valid AddAudioEncoderConfigurationResponse message.

The DUT did not send valid GetAudioEncoderConfigurationResponse message.

The DUT did not send valid GetCompatibleAudioEncoderConfigurationsResponse message.

The DUT did not send valid GetCompatibleAudioSourceConfigurationsResponse message.

UseCount value is not increased by 1 after adding of the AudioEncoderConfiguration to one more profile.

UseCount value is not decreased by 1 after replacing of the AudioEncoderConfiguration in profile for removed AudioEncoderConfiguration and UseCount value is not increased by 1 after replacing of the AudioEncoderConfiguration in profile for new AudioEncoderConfiguration.

Note: See Annex A.5 for Name and Token Parameters Length limitations.



4.7.9 AUDIO ENCODER CONFIGURATION USE COUNT (REMOVE AUDIO ENCODER CONFIGURATION)

Test Label: Media Service DUT Audio Encoder Configuration Use Count Validation after Removing of Audio Encoder Configuration from Profile.

-196-

Test Case ID: MEDIA-3-3-14

ONVIF Core Specification Coverage: Get media profiles, Remove audio encoder configuration from a profile, Get audio encoder configurations, Get audio encoder configuration

Command Under Test: GetProfiles, GetAudioEncoderConfigurations, GetAudioEncoderConfiguration, RemoveAudioEncoderConfiguration

WSDL Reference: media.wsdl

Test Propose: To check Audio Encoder Configuration use count after removing audio encoder configuration from profile.

Pre-Requisite: Media is supported by DUT. Audio is supported by DUT. ONVIF Client gets the Media Service entry point by GetCapabilities command. List of media profiles is received by GetProfiles command.

Test Configuration: ONVIF Client and DUT

DUT

		Start DUT
Retrieve list of	GetAudioEncoderConfigurationsRequest	Send list of
AECS	GetAudioEncoderConfigurations Response (list of existing	existing
Receive and validate AECs	AudioEncoderConfigurations)	Configurations
Retrieve list of	GetAudioSourceConfigurationsRequest	
Receive and validate ASCs	GetAudioSourceConfigurationsResponse (list of AudioSourceConfigurations)	Send list of existing AudioSource
	CreateProfileRequest (ProfileToken = New Profile)	Configurations
Create new profile		
Receive and validate response	CreateProfileResponse (empty)	Send CreateProfile Response
Get compatible audio source	GetCompatibleAudioSource ConfigurationsRequest (ProfileToken = New Profile)	Send Compatible
Receive and validate response	GetCompatibleAudioSource ConfigurationsResponse (empty)	AudioSource Configurations Response
Add ASC1 to New Profile	AddAudioSourceConfigurationRequest (ProfileToken = New Profile, ConfigurationToken = ASC1)	Send
Receive and validate response	AddAudioSourceConfigurationResponse (empty)	AddAudioEncoder Configuration Response
Get compatible audio encoder configurations	GetCompatibleAudioEncoder ConfigurationsRequest (ProfileToken = New Profile)	Send Compatible
Receive and validate response	GetCompatibleAudioEncoderConfiguratio nsResponse (empty)	AudioEncoder Configurations Response

ONVIF	Client
-------	--------

DUT

Add AudioEncoder Configuration to	AddAudioEncoderConfigurationRequest (ProfileToken = New Profile, ConfigurationToken = AEC1)	
profile	AddAudioEncoderConfigurationResponse (empty)	Add audio encoder configuration to
Receive and		profile
	RemoveAudioEncoderConfiguration Request (ProfileToken = New Profile)	
Remove AEC from		Remove audio
Receive and	RemoveAudioEncoderConfiguration Response (empty)	encoder configuration form
validate response	•	prome
Retrieve parameters of	GetAudioEncoderConfigurationRequest (ConfigurationToken = AEC1)	
AEC1	GetAudioEncoderConfigurationResponse	Send parameters for AEC1
Receive and validate AEC1	<	
	1	

Test Procedure:

- 1. Start an ONVIF Client.
- 2. Start the DUT.
- 3. ONVIF Client invokes GetAudioEncoderConfigurationsRequest message to retrieve list of audio encoder configurations from device.
- 4. The DUT sends GetAudioEncoderConfigurationsResponse message.
- 5. ONVIF Client invokes GetAudioSourceConfigurationsRequest message to retrieve list of audio source configurations from device.
- 6. The DUT sends GetAudioSourceConfigurationsResponse message.
- 7. ONVIF Client invokes CreateProfileRequest (Name = NewName, Token = New Profile) message to create profile.
- 8. If the DUT will generate a SOAP 1.2 fault message (Action/MaxNVTProfiles),

If there is profile with "fixed" attribute value "false" in profiles list (profile1)

ONVIF Client invokes RemoveAudioEncoderConfigurationRequest (**ProfileToken = profile1**) message to remove AudioEncoderConfiguration from profile1

The DUT sends RemoveAudioEncoderConfigurationResponse message

Execute steps 10-22

If there is no profile with "fixed" attribute value "false" in profiles list, end test.

- 9. If the DUT sends CreateProfileResponse message, validate CreateProfileResponse message from the DUT.
- 10. ONVIF Client invokes GetCompatibleAudioSourceConfigurationsRequest (**ProfileToken = New Profile**) message to get compatible AudioSourceConfigurations for the new profile.
- 11. The DUT sends GetCompatibleAudioSourceConfigurationsResponse message.
- ONVIF Client invokes AddAudioSourceConfigurationRequest (ConfigurationToken = first audio source configuration from the list on step 11 (ASC1), ProfileToken = New Profile) message to add AudioSourceConfiguration to the new profile.
- 13. The DUT sends AddAudioSourceConfigurationResponse message.
- 14. ONVIF Client invokes GetCompatibleAudioEncoderConfigurationsRequest (ProfileToken = New Profile) message to get compatible AudioEncoderConfigurations for the new profile with Audio Source Configuration.
- 15. The DUT sends GetCompatibleAudioEncoderConfigurationsResponse message. If response does not contain any AudioEncoderConfiguration, repeat steps 12-15 with next audio source configuration from step 11. If there are no other audio source configuration, skip other steps and go to the next test.
- 16. ONVIF Client invokes AddAudioEncoderConfigurationRequest (ConfigurationToken = first audio encoder from the list on step 15 (AEC1), ProfileToken = New Profile) message to add AudioEncoderConfiguration to the new profile.
- 17. ONVIF Client invokes RemoveAudioEncoderConfigurationRequest (**ProfileToken = New Profile**) message to remove audio encoder configuration from profile1.
- 18. The DUT sends RemoveAudioEncoderConfigurationResponse message.
- 19. ONVIF Client invokes GetAudioEncoderConfigurationRequest (**ConfigurationToken = AEC1**) message to retrieve audio encoder configuration parameters.
- 20. The DUT sends GetAudioEncoderConfigurationResponse message.
- 21. Check that **UseCount = usecount1**, in GetAudioEncoderConfigurationResponse where usecount1 is UseCount value for AEC1 in the list on step 4. If test step 8 is executed, the usecount1 value could be reduced by 1.
- 22. ONVIF Client restores DUT configuration.

Test Result:

PASS –

DUT passes all assertions.

FAIL -

The DUT did not send valid GetAudioEncoderConfigurationsResponse message.

The DUT did not send valid AddAudioEncoderConfigurationResponse message.



The DUT did not send valid RemoveAudioEncoderConfigurationResponse message.

The DUT did not send valid GetAudioEncoderConfigurationResponse message.

The DUT did not send valid GetCompatibleAudioEncoderConfigurationsResponse message.

The DUT did not send valid GetCompatibleAudioSourceConfigurationsResponse message.

UseCount value is not decreased by 1 after removing of the AudioEncoderConfiguration from profile.

Note: See Annex A.5 for Name and Token Parameters Length limitations.

4.7.10 AUDIO ENCODER CONFIGURATION USE COUNT (DELETION PROFILE WITH AUDIO ENCODER CONFIGURATION)

Test Label: Media Service DUT Audio Encoder Configuration Use Count Validation after Deletion of Profile with Audio Encoder Configuration in it.

Test Case ID: MEDIA-3-3-15

ONVIF Core Specification Coverage: Get audio encoder configurations, Get audio encoder configuration, Delete media profile

Command Under Test: GetAudioEncoderConfigurations, GetAudioEncoderConfiguration, DeleteProfile.

WSDL Reference: media.wsdl

Test Propose: To check Audio Encoder Configuration use count after deletion profile with audio encoder configuration in it.

Pre-Requisite: Media is supported by DUT. Audio is supported by DUT. ONVIF Client gets the Media Service entry point by GetCapabilities command. List of media profiles is received by GetProfiles command.

Test Configuration: ONVIF Client and DUT

DUT

		Start DUT	
Retrieve list of	GetAudioEncoderConfigurationsRequest	Send list of	
Receive and	GetAudioEncoderConfigurations Response (list of existing AudioEncoderConfigurations)	existing AudioEncoder Configurations	
validate AECs	- / Marcentosadi Comigarationo)	,	
Retrieve list of ASCs	GetAudioSourceConfigurationsRequest ►		
Receive and validate ASCs	GetAudioSourceConfigurationsResponse (list of AudioSourceConfigurations)	Send list of existing AudioSource	
	CreateProfileRequest (ProfileToken = New Profile)	Configurations	
Create new profile			
Receive and validate response	CreateProfileResponse (empty)	Send CreateProfile Response	
Get compatible audio source configurations	GetCompatibleAudioSource ConfigurationsRequest (ProfileToken = New Profile)	Send Compatible	
Receive and validate response	GetCompatibleAudioSource ConfigurationsResponse (empty)	AudioSource Configurations Response	
Add ASC1 to New Profile	AddAudioSourceConfigurationRequest (ProfileToken = New Profile, ConfigurationToken = ASC1)	Send	
Receive and validate response	AddAudioSourceConfigurationResponse (empty)	AddAudioEncoder Configuration Response	
Get compatible audio encoder configurations	GetCompatibleAudioEncoder ConfigurationsRequest (ProfileToken = New Profile)	Send Compatible	
Receive and validate response	GetCompatibleAudioEncoder ConfigurationsResponse (empty)	AudioEncoder Configurations Response	

Г

ONVIF	Client	DUT	
Add AudioEncoder Configuration to	AddAudioEncoderConfigurationRequest (ProfileToken = New Profile, ConfigurationToken = AEC1)		Add audio encoder
Receive and validate response	AddAudioEncoderConfigurationRespons (empty)	se	configuration to profile
Delete Profile	DeleteProfileRequest (ProfileToken = New Profile)		Delete profile
Receive and validate response	DeleteProfileResponse (empty)		Delete profile
Retrieve parameters of AEC1	GetAudioEncoderConfigurationRequest (ConfigurationToken = AEC1)		
Receive and validate AEC1	GetAudioEncoderConfigurationRespons (parameters for AEC1)	e	Send parameters for AEC1

Test Procedure:

- 1. Start an ONVIF Client.
- 2. Start the DUT.
- 3. ONVIF Client invokes GetAudioEncoderConfigurationsRequest message to retrieve list of audio encoder configurations from device.
- 4. The DUT sends GetAudioEncoderConfigurationsResponse message.
- 5. ONVIF Client invokes GetAudioSourceConfigurationsRequest message to retrieve list of audio source configurations from device.
- 6. The DUT sends GetAudioSourceConfigurationsResponse message.
- 7. ONVIF Client invokes CreateProfileRequest (Name = NewName, Token = New Profile) message to create profile.
- 8. If the DUT will generate a SOAP 1.2 fault message (Action/MaxNVTProfiles),

If there is profile with "fixed" attribute value "false" in profiles list (profile1)

ONVIF Client invokes RemoveAudioEncoderConfigurationRequest (**ProfileToken = profile1**) message to remove AudioEncoderConfiguration from profile1

The DUT sends RemoveAudioEncoderConfigurationResponse message

Execute steps 10-22



If there is no profile with "fixed" attribute value "false" in profiles list, end test.

- 9. If the DUT sends CreateProfileResponse message, validate CreateProfileResponse message from the DUT.
- 10. ONVIF Client invokes GetCompatibleAudioSourceConfigurationsRequest (**ProfileToken = New Profile**) message to get compatible AudioSourceConfigurations for the new profile.
- 11. The DUT sends GetCompatibleAudioSourceConfigurationsResponse message.
- 12. ONVIF Client invokes AddAudioSourceConfigurationRequest (ConfigurationToken = first audio source configuration from the list on step 11 (ASC1), ProfileToken = New Profile) message to add AudioSourceConfiguration to the new profile.
- 13. The DUT sends AddAudioSourceConfigurationResponse message.
- 14. ONVIF Client invokes GetCompatibleAudioEncoderConfigurationsRequest (ProfileToken = New Profile) message to get compatible AudioEncoderConfigurations for the new profile with Audio Source Configuration.
- 15. The DUT sends GetCompatibleAudioEncoderConfigurationsResponse message. If response does not contain any AudioEncoderConfiguration, repeat steps 12-15 with next audio source configuration from step 11. If there are no other audio source configuration, skip other steps and go to the next test.
- 16. ONVIF Client invokes AddAudioEncoderConfigurationRequest (ConfigurationToken = first audio encoder from the list on step 15 (AEC1), ProfileToken = New Profile) message to add AudioEncoderConfiguration to the new profile.
- 17. ONVIF Client invokes DeleteProfileRequest (**ProfileToken = New Profile**) message to remove profile with audio encoder configuration.
- 18. The DUT sends DeleteProfileResponse message.
- 19. ONVIF Client invokes GetAudioEncoderConfigurationRequest (**ConfigurationToken = AEC1**) message to retrieve audio encoder configuration parameters.
- 20. The DUT sends GetAudioEncoderConfigurationResponse message.
- 21. Check that **UseCount = usecount1**, in GetAudioEncoderConfigurationResponse where usecount1 is UseCount value for AEC1 in the list on step 4.If test step 8 is executed, the usecount1 value could be reduced by 1.
- 22. ONVIF Client restores DUT configuration.

Test Result:

PASS –

DUT passes all assertions.

FAIL –

The DUT did not send valid GetAudioEncoderConfigurationsResponse message.

The DUT did not send valid AddAudioEncoderConfigurationResponse message.

The DUT did not send valid DeleteProfileResponse message.

The DUT did not send valid GetAudioEncoderConfigurationResponse message.



The DUT did not send valid GetCompatibleAudioEncoderConfigurationsResponse message.

The DUT did not send valid GetCompatibleAudioSourceConfigurationsResponse message.

UseCount value is not decreased by 1 after deletion of the profile with the AudioEncoderConfiguration.

Note: See Annex A.5 for Name and Token Parameters Length limitations.

4.7.11 AUDIO ENCODER CONFIGURATION USE COUNT (SET AUDIO ENCODER CONFIGURATION)

Test Label: Media Audio Encoder Configuration Use Count Validation after Changing of Audio Encoder Configuration.

Test Case ID: MEDIA-3-3-16

ONVIF Core Specification Coverage: Get media profiles, Get audio encoder configurations, Get audio encoder configuration, Modify audio encoder configurations.

Command Under Test: GetAudioEncoderConfigurations, GetAudioEncoderConfiguration, SetAudioEncoderConfiguration

WSDL Reference: media.wsdl

Test Propose: To check Audio Encoder Configuration use count after setting audio encoder configuration parameters.

Pre-Requisite: Media is supported by DUT. Audio is supported by DUT. ONVIF Client gets the Media Service entry point by GetCapabilities command.

Test Configuration: ONVIF Client and DUT

ONVIF Client		DUT	
Retrieve list of AECs	GetAudioEncoderConfigurationsRequest	Start DUT	
Receive and	GetAudioEncoderConfigurations Response (list of existing AudioEncoderConfigurations)	Send list of existing AudioEncoder Configurations	
Set new UseCount	SetAudioEncoderConfigurationRequest (AudioEncoderConfiguration = AEC1, UseCount = NewUseCount)	Quad	
Receive and	SetAudioEncoderConfigurationResponse (empty)	Send SetAudioEncoder Configuration Response	
Retrieve parameters of AEC1 Receive and validate AEC1	GetAudioEncoderConfigurationRequest (ConfigurationToken = AEC1)		
	GetAudioEncoderConfigurationResponse (parameters for AEC1) ◀	Send parameters for AEC1	

Test Procedure:

- 1. Start an ONVIF Client.
- 2. Start the DUT.
- 3. ONVIF Client invokes GetAudioEncoderConfigurationsRequest message to retrieve list of audio encoder configurations from device.
- 4. The DUT sends GetAudioEncoderConfigurationsResponse message.
- 5. ONVIF Client invokes SetAudioEncoderConfigurationRequest (token = the first Audio Encoder Configuration from the list on step 4(AEC1), UseCount = NewUseCount) message to set parameters for audio encoder configuration.
- 6. The DUT sends SetAudioEncoderConfigurationResponse message.
- 7. ONVIF Client invokes GetAudioEncoderConfigurationRequest (ConfigurationToken = AEC1) message to retrieve audio encoder configuration parameters.
- 8. The DUT sends GetAudioEncoderConfigurationResponse message.
- 9. Check that UseCount = usecount1, in GetAudioEncoderConfigurationResponse where usecount1 is UseCount value for AEC1 in the list on step 4.

Test Result:

PASS –

DUT passes all assertions.

FAIL -

The DUT did not send GetAudioEncoderConfigurationsResponse message.

The DUT did not send valid GetAudioEncoderConfigurationsResponse message.

The DUT did not send GetAudioEncoderConfigurationResponse message.

The DUT did not send valid GetAudioEncoderConfigurationResponse message.

The DUT did not send SetAudioEncoderConfigurationResponse message.

The DUT did not send valid SetAudioEncoderConfigurationResponse message.

UseCount value changed after trying to set UseCount value.

Note: See Annex A.5 for Name and Token Parameters Length limitations.

4.8 Audio Output Configuration

4.8.1 SET AUDIO OUTPUT CONFIGURATION

Test Label: Media Service DUT SetAudioOutputConfiguration Command Validation.

Test Case ID: MEDIA-3-4-4

ONVIF Core Specification Coverage: Get audio output configurations, Get audio output configuration, Get audio output configuration options, Modify audio output configuration

Command Under Test: GetAudioOutputConfigurations, GetAudioOutputConfiguration, GetAudioOutputConfigurationOptions, SetAudioOutputConfiguration

WSDL Reference: media.wsdl

Test Propose: To check SetAudioOutputConfiguration behavior.

Pre-Requisite: Media is supported by DUT. Audio output is supported by DUT. ONVIF Client gets the Media Service entry point by GetCapabilities command.

Test Configuration: ONVIF Client and DUT

ONVIF	Client E	DUT
Retrieve list of	GetAudioOutputConfigurationsRequest	Start DUT
AOCs	GetAudioOutputConfigurationsResponse (list of existing AudioOutputConfigurations)	 Send list of existing AudioOutput Configurations
Receive and validate AOCs		
Retrieve options	Request (ConfigurationToken = AOC1)	•
Receive and	GetAudioOutputConfigurationOptions Response (options)	Send options for AOC1.
AOC1	←	_
Set new settings	SetAudioOutputConfiguration (token = AOC1, Name, OutputToken, SendPrivacy OutputLevel, ForcePersistence)	,
for AOC1	SetVideoEncoderConfigurationResponse	Set audio output
Receive and validate response	(empty)	configuration
Retrieve	GetAudioOutputConfigurationRequest (ConfigurationToken = AOC1)	
AOC1	GetAudioOutputConfigurationResponse (parameters for AOC1)	Send parameters for AOC1
Receive and validate AOC1	<	-

Test Procedure:

- 1. Start an ONVIF Client.
- 2. Start the DUT.
- 3. ONVIF Client invokes GetAudioOutputConfigurationsRequest message to retrieve list of audio output configurations from device.
- 4. The DUT sends GetAudioOutputConfigurationsResponse message.
- 5. ONVIF Client invokes GetAudioOutputConfigurationOptionsRequest (ConfigurationToken = AOC1, where AOC1 is the first configuration from the list on step 4) message to retrieve options for AOC1 from device.
- 6. The DUT sends GetAudioOutputConfigurationOptionsResponse message.



- 7. ONVIF Client invokes SetAudioOutputConfigurationRequest (Name = NewName, token = AOC1, OutputToken = OT1, where OT1 is one of OutputTokensAvailable from GetAudioOutputConfigurationResponse, SendPrimacy = SendPrimacyMode1, where SendPrimacyMode1 is one of modes from SendPrimacyOptions from Response on step 6, OutputLevel = OL1, where OL1 is between OutputRange.min and OutputRange.max from response on step 6, ForcePersistence = false) message to set parameters for audio output configuration.
- 8. The DUT sends SetAudioOutputConfigurationResponse message.
- 9. Validate SetAudioOutputConfigurationResponse
- 10. ONVIF Client invokes GetAudioOutputConfigurationRequest (ConfigurationToken = AOC1) message to retrieve settings of AOC1 from device.
- 11. The DUT sends GetAudioOutputConfigurationResponse message.
- 12. Validate AOC1 settings. Check, that all settings are as were set on step 7.
- 13. Repeat steps 5-12 for other Audio Output Configurations from step 4.

Test Result:

PASS –

DUT passes all assertions.

FAIL –

The DUT did not send GetAudioOutputConfigurationsResponse message.

The DUT did not send valid GetAudioOutputConfigurationsResponse message.

The DUT did not send GetAudioOutputConfigurationOptionsResponse message.

The DUT did not send valid GetAudioOutputConfigurationOptionsResponse message.

The DUT did not send GetAudioOutputConfigurationResponse message.

The DUT did not send valid GetAudioOutputConfigurationResponse message.

The DUT did not send SetAudioOutputConfigurationResponse message.

The DUT did not send valid SetAudioOutputConfigurationResponse message.

At least for one Audio Output Configuration after modifying settings, there is at least one of following items:

- Name is not equal to NewName;
- OutputToken is not equal to OT1;
- SendPrimacy is not equal to SendPrimacyMode1;
- OutputLevel is not equal OL1;
- ForcePersistence is not false.

Note: See Annex A.5 for Name and Token Parameters Length limitations.

4.8.2 SET AUDIO OUTPUT CONFIGURATION – INVALID OUTPUTTOKEN

Test Label: Media Service DUT SetAudioOutputConfiguration Command Validation (Invalid OUTPUTTOKEN).

Test Case ID: MEDIA-3-4-6

ONVIF Core Specification Coverage: Modify audio output configuration

Command Under Test: SetAudioOutputConfiguration

WSDL Reference: media.wsdl

Test Propose: To check SetAudioOutputConfiguration command behavior in case of incorrect configuration token.

Pre-Requisite: Media is supported by DUT. Audio output is supported by DUT. ONVIF Client gets the Media Service entry point by GetCapabilities command.

Test Configuration: ONVIF Client and DUT

ONVIF	Client	DUT
Retrieve list of	GetAudioOutputConfigurationsRequest	Start DUT
Receive and	GetAudioOutputConfigurationsResponse (list of existing AudioOutputConfigurations)	Send parameters for AOC1
Retrieve options	GetAudioOutputConfigurationOptions Request (ConfigurationToken = AOC1)	
Receive and validate options for AOC1	GetAudioOutputConfigurationOptions Response (options)	Send options for AOC1.
Set new settings	SetAudioOutputConfiguration (token = AOC1, Name, InvalidOutputToken, InvalidSendPrivacy, InvalidOutputLevel)	
Receive and validate response	SOAP 1.2 fault response (InvalidArgVal/ConfigModify)	Set audio output configuration
Retrieve parameters of	GetAudioOutputConfigurationRequest (ConfigurationToken = AOC1)	_
Receive and	GetAudioOutputConfigurationResponse (parameters for AOC1)	Send parameters for AOC1
validate AOC1		

Test Procedure:

- 1. Start an ONVIF Client.
- 2. Start the DUT.
- 3. ONVIF Client invokes GetAudioOutputConfigurationsRequest message to retrieve list of audio output configurations from device.
- 4. The DUT sends GetAudioOutputConfigurationsResponse message.
- 5. ONVIF Client invokes GetAudioOutputConfigurationOptionsRequest (ConfigurationToken = AOC1, where AOC1 is the first configuration from the list on step 4) message to retrieve options for AOC1 from device.
- 6. The DUT sends GetAudioOutputConfigurationOptionsResponse message.



- 7. ONVIF Client invokes SetAudioOutputConfigurationRequest (Name = NewName, token = AOC1, OutputToken = InvalidOT1, where InvalidOT1 is not one of OutputTokensAvailable from GetAudioOutputConfigurationResponse, SendPrimacy = InvalidSendPrimacyMode1, where InvalidSendPrimacyMode1 is not one of modes from SendPrimacyOptions from Response on step 6, OutputLevel = InvalidOL1, where OL1 is not between OutputRange.min and OutputRange.max from response on step 6) message with incorrect parameters.
- 8. The DUT will generate a SOAP 1.2 fault message (Sender/InvalidArgVal/ConfigModify).

-212-

- 9. ONVIF Client invokes GetAudioOutputConfigurationRequest (ConfigurationToken = AOC1) message to retrieve settings of AOC1 from device.
- 10. The DUT sends GetAudioOutputConfigurationResponse message.
- 11. Validate AOC1 settings. Check, that all settings are as settings of AOC1 in the list on step 4.
- 12. Repeat steps 5-12 for other Audio Output Configurations from step 4.

Test Result:

PASS –

DUT passes all assertions.

FAIL –

The DUT did not send GetAudioOutputConfigurationsResponse message.

The DUT did not send valid GetAudioOutputConfigurationsResponse message.

The DUT did not send GetAudioOutputConfigurationOptionsResponse message.

The DUT did not send valid GetAudioOutputConfigurationOptionsResponse message.

The DUT did not send GetAudioOutputConfigurationResponse message.

The DUT did not send valid GetAudioOutputConfigurationResponse message.

The DUT did not send SOAP 1.2 fault message.

The DUT sent incorrect SOAP 1.2 fault message (fault code, namespace, etc.).

Settings are changed after incorrect SetAudioOutputConfigurationRequest.

Note: Other faults than specified in the test are acceptable but specified is preferable.

Note: See Annex A.5 for Name and Token Parameters Length limitations.

4.8.3 SET AUDIO OUTPUT CONFIGURATION – INVALID CONFIGURATION

Test Label: Media Service DUT SetAudioOutputConfiguration Command Validation (Invalid Configuration).

Test Case ID: MEDIA-3-4-7

ONVIF Core Specification Coverage: 11.12.5 Modify audio output configuration

Command Under Test: SetAudioOutputConfiguration

WSDL Reference: media.wsdl

Test Propose: To check SetAudioOutputConfiguration command behavior in case of incorrect configuration token.

Pre-Requisite: Media is supported by the DUT. Audio output is supported by the DUT. ONVIF Client gets the Media Service entry point by GetCapabilities command.

Test Configuration: ONVIF Client and DUT

Test Sequence:

ONVIF	Client	DUT	
Retrieve list of	SetAudioOutputConfigurationRequest (ConfigurationToken = InvalidAudioOutput)		Start DUT
Receive and validate SOAP 1.2 fault	SOAP 1.2 fault response (InvalidArgVal/NoConfig) ◀		Send SOAP 1.2 fault

Test Procedure:

- 1. Start an ONVIF Client.
- 2. Start the DUT.
- 3. ONVIF Client invokes SetAudioOutputConfigurationRequest (**token = InvalidAudioOutput**) message with incorrect ConfigurationToken.
- 4. The DUT will generate a SOAP 1.2 fault message (Sender/InvalidArgVal/NoConfig).

Test Result:

PASS –

The DUT passed all assertions.

FAIL -

The DUT did not send SOAP 1.2 fault message.

The DUT sent an incorrect SOAP 1.2 fault message (fault code, namespace, etc.).

Note: Other faults than specified in the test are acceptable though the specified are preferable.

Note: See Annex A.5 for Name and Token Parameters Length limitations.

4.8.4 GET AUDIO OUTPUT CONFIGURATION OPTIONS

Test Label: Get Audio Output Configuration Options command verification.

Test Case ID: MEDIA-3-4-8

ONVIF Core Specification Coverage: Audio Output Configuration (ONVIF Media Service ONVIF www.onvif.org info@onvif.org



Specification)

Command Under Test: GetAudioOutputConfigurations, GetAudioOutputConfiguration, GetAudioOutputConfigurationOptions, SetAudioOutputConfiguration

WSDL Reference: media.wsdl

Test Propose: To check GetAudioOutputConfigurationOptions behavior.

Pre-Requisite: Media service is supported by the DUT. Audio output is supported by the DUT. Media Service is received from the DUT.

Test Configuration: ONVIF Client and DUT

ONVIF	Client D	UT
Retrieve list of profiles	GetProfilesRequest	Start DUT
Receive and validate list of	GetProfilesResponse	Send list of existing profiles.
profiles Retrieve list of AudioOutput	GetAudioOutputConfigurationsRequest	
Configurations Receive and	GetAudioOutputConfigurationsResponse (list of AudioSourceConfigurations)	Send list of AudioOutput Configurations
validate response Retrieve AudioOutput	 GetAudioOutputConfigurationOptions Request (ConfigurationToken) 	
Configuration options Receive and	GetAudioOutputConfigurationOptions Response	 Send GetAudioOutput Configuration options
validate response Retrieve	 GetAudioOutputConfigurationOptions Request (ProfileToken) 	
Configuration options	GetAudioOutputConfigurationOptions Response	→ Send GetAudioOutput Configuration options
Receive and validate response Retrieve	GetAudioOutputConfigurationOptions Request (ProfileToken, ConfigurationToken)	
Configuration options Receive and validate response Retrieve AudioOutput Configuration options	GetAudioOutputConfigurationOptions Response	→ Send GetAudioOutput Configuration options
	 GetAudioOutputConfigurationOptions Request (empty) 	
	GetAudioOutputConfigurationOptions Response	→ Send GetAudioOutput Configuration
Receive and validate response	<	

Test Procedure:

1. Start an ONVIF Client.

- 2. Start the DUT.
- 3. ONVIF Client invokes GetProfiles to get list of existing profiles from the DUT.

-216-

- 4. The DUT responds with a GetProfilesResponse message with parameters
 - Profiles =: profileList
- 5. If *profileList* is empty, FAIL the test and skip other steps.
- 6. ONVIF Client invokes **GetAudioOutputConfigurations** to get list of existing audio output configurations.
- 7. The DUT responds with GetAudioOutputConfigurations message with parameters
 - Configurations =: configurationsList
- 8. If *configurationsList* is empty, FAIL the test and skip other steps.
- 9. Set the following:
 - configurationToken := configurationsList.token[0]
 - profileToken := profileList.token[0]
- 10. ONVIF Client invokes GetAudioOutputConfigurationOptions with parameters
 - ConfigurationToken := configurationToken
 - ProfileToken := *profileToken*
- 11. The DUT responds with **GetAudioOutputConfigurationOptionsResponse** message with parameters
 - Options
- 12. ONVIF Client invokes GetAudioOutputConfigurationOptions with parameters
 - ConfigurationToken skipped
 - ProfileToken := profileToken
- 13. The DUT responds with **GetAudioOutputConfigurationOptionsResponse** message with parameters
 - Options
- 14. ONVIF Client invokes GetAudioOutputConfigurationOptions with parameters
 - ConfigurationToken := *configurationToken*
 - ProfileToken skipped
- 15. The DUT responds with **GetAudioOutputConfigurationOptionsResponse** message with parameters
 - Options
- 16. ONVIF Client invokes GetAudioOutputConfigurationOptions with parameters
- ConfigurationToken skipped
- ProfileToken skipped
- 17. The DUT responds with **GetAudioOutputConfigurationOptionsResponse** message with parameters
 - Options

Test Result:

PASS –

DUT passes all assertions.

FAIL –

The DUT did not send GetProfilesResponse message.

The DUT did not send GetAudioOutputConfigurationsResponse message.

The DUT did not send **GetAudioOutputConfigurationOptionsResponse** message.

4.8.5 AUDIO OUTPUT CONFIGURATION

Test Label: Audio Output Configuration verification

Test Case ID: MEDIA-3-4-9

ONVIF Core Specification Coverage: Audio Output Configuration (ONVIF Media Service Specification), Add audio output configuration to a profile (ONVIF Media Specification), Remove audio output configuration from a profile (ONVIF Media Specification), Get audio outputs (ONVIF Media Specification), Get audio output configurations (ONVIF Media Specification), Get audio output configuration (ONVIF Media Specification), Get audio output configuration), Get audio output configuration, Get audio output configuration (ONVIF Media Specification), Get audio output configuration (ONVIF Media Specification), Get audio output configuration options (ONVIF Media Specification), Modify an audio output configuration (ONVIF Media Specification).

CommandUnderTest:GetAudioOutputs,GetAudioOutputConfigurations,GetAudioOutputConfiguration,GetCompatibleAudioOutputConfigurations,GetCompatibleAudioOutputConfigurations,AddAudioOutputConfiguration,GetAudioOutputConfigurationOptions,SetAudioOutputConfiguration,RemoveAudioOutputConfigurationGetAudioOutputConfiguration,SetAudioOutputConfiguration,

WSDL Reference: media.wsdl

Test Purpose: To verify Audio Output Configuration Operations

Pre-Requisite: Media service is supported by the DUT. Audio output is supported by the DUT. Media Service is received from the DUT.

Test Configuration: ONVIF Client and DUT

Test Sequence:

ONVIF Client		DUT	
Create or select empty media profile	HelperCreateEmptyProfile (in <i>profileName,</i> out <i>profileToken</i> , out <i>profileToDelete</i> , out <i>profileToRestore</i>) Annex A.6	Send empty profile token	
	GetAudioOutputsRequest		
Receive and Validate Audio Outputs	GetAudioOutputsResponse (AudioOutputs := <i>audioOutputsList</i>)	Send Audio Outputs of the device	
	GetCompatibleAudioOutputConfigurations (ProfileToken := <i>profileToken</i>)		
Receive and Validate Compatible Audio Output	GetCompatibleAudioOutputConfigurations Response (Configurations =: compatibleConfigurationsList)	Send Audio Output Configurations compatible with the received media profile	
Configurations Add audio Output configuration to new	AddAudioOutputConfiguration (ProfileToken := profileToken, ConfigurationToken := configurationToken)		
media profile Receive and Validate	AddAudioOutputConfigurationResponse	Add audio Output configuration and send response	
response	 GetAudioOutputConfigurationOptions (ProfileToken skipped, ConfigurationToken := configurationToken) 		
Receive and Validate Audio Output	GetAudioOutputConfigurationOptions Response (Options =: <i>options</i>)	Send configurable options for audio Output configuration	
configurable options			

ONVIF	Client	DUT
Set Invalid audio source cfg values	SetAudioOutputConfiguration (Configuration.Name := compatibleConfigurationsList[0].token.Name Configuration.UseCount := compatibleConfigurationsList[0].token.UseCount, Configuration.Token := configuration.OutputToken := compatibleConfigurationsList[0].token.Output Token, ConfigurationsList[0].token.Output Token, ConfigurationsList[0].token.SendF rimacy, Configuration.OutputLevel := invalidOutputLevel, ForcePersistence := false) SOAP Fault Message (InvalidArgVal/ConfigModify)	Send SOAP 1.2 fault message
Oct Audio Octout	SetAudioOutputConfiguration (Configuration.Name := compatibleConfigurationsList[0].token.Name, Configuration.UseCount := compatibleConfigurationsList[0].token.UseCount, Configuration.Token := configuration.Token, Configuration.OutputToken := compatibleConfigurationsList[0].token.Output Token, Configuration.SendPrimacy := compatibleConfigurationsList[0].token.SendF rimacy, Configuration.OutputLevel := validOutputLevel, ForcePersistence := false)	
Set Audio Output Configuration	SetAudioOutputConfigurationResponse	Modify Audio Output Configuration
Receive and Validate Audio Output Configuration	GetAudioOutputConfiguration (ConfigurationToken := <i>configurationToken</i>)	Send the modified Audio
	GetAudioOutputConfigurationResponse (Configuration =: <i>updatedConfiguration</i>) RemoveAudioOutputConfiguration (ProfileToken := <i>profileToken</i>)	Output Configuration
		*



Test Procedure:

- 1. Start an ONVIF Client.
- 2. Start the DUT.
- 3. Set the following:
 - profileName := "TestProfile"
- 4. ONVIF Client generates creation or selection of profile with name (in *profileName*) and with token (out *profileToken*) and retrieves media profile token to delete (out *profileToDelete*) or media profile to restore (out *profileToRestore*) by following the procedure mentioned in Annex A.6.
- 5. ONVIF Client invokes GetAudioOutputs to retrieve the existing audio outputs of DUT.
- 6. The DUT responds with **GetAudioOutputsResponse** with parameters:
 - AudioOutputs := audioOutputsList
- 7. If *audioOutputsList* is empty, FAIL the test and go to step 31.
- 8. ONVIF Client invokes GetCompatibleAudioOutputConfigurations with parameters:
 - ProfileToken := profileToken
- 9. The DUT responds with **GetCompatibleAudioOutputConfigurationsResponse** with parameters:
 - Configurations =: compatibleConfigurationsList
- 10. If *compatibleConfigurationsList* is empty, FAIL the test and go to step 31.
- 11. Set the following:
 - configurationToken := compatibleConfigurationsList.token[0]

12. ONVIF *Client* invokes **AddAudioOutputConfiguration** with parameters:

- ProfileToken := profileToken
- ConfigurationToken := configurationToken
- 13. The DUT responds with AddAudioOutputResponse message.
- 14. ONVIF Client invokes GetAudioOutputConfigurationOptions with parameters:

-221-

- ProfileToken skipped
- ConfigurationToken := configurationToken
- 15. The DUT responds with **GetAudioOutputConfigurationOptions** with parameters:
 - Options =: options
- 16. Set the following:
 - *invalidOutputLevel* := *options*.OutputLevelRange.Min 1.
- 17. ONVIF Client invokes SetAudioOutputConfiguration with parameters:
 - Configuration.Name := compatibleConfigurationsList[0].token.Name
 - Configuration.UseCount := compatibleConfigurationsList[0].token.UseCount
 - Configuration.Token := configurationToken
 - Configuration.OutputToken := compatibleConfigurationsList[0].token.OutputToken
 - Configuration.SendPrimacy := *compatibleConfigurationsList*[0].token.SendPrimacy
 - Configuration.OutputLevel := *invalidOutputLevel*
 - ForcePersistence := false
- 18. The DUT responds with env:Sender\ter: InvalidArgVal \ter:ConfigModify SOAP 1.2 fault.
- 19. Set the following:
 - *validOutputLevel* := *options*.OutputLevelRange.Max.
- 20. ONVIF Client invokes SetAudioOutputConfiguration with parameters:
 - Configuration.Name := compatibleConfigurationsList[0].token.Name
 - Configuration.UseCount := compatibleConfigurationsList[0].token.UseCount
 - Configuration.Token := configurationToken
 - Configuration.OutputToken := compatibleConfigurationsList[0].token.OutputToken
 - Configuration.SendPrimacy := compatibleConfigurationsList[0].token.SendPrimacy
 - Configuration.OutputLevel := validOutputLevel
 - ForcePersistence := false

- 21. The DUT responds with **SetAudioOutputConfigurationResponse** message.
- 22. ONVIF Client invokes GetAudioOutputConfiguration with parameters:
 - ConfigurationToken := configurationToken
- 23. DUT responds with **GetAudioOutputConfiguration** with parameters:
 - Configuration =: updatedConfiguration
- 24. If *updatedConfiguration*.Name does not equal to *compatibleConfigurationsList*[0].token.Name, FAIL the test and go to step 31.
- 25. If *updatedConfiguration*.Token does not equal to *configurationToken*, FAIL the test and go to step 31.
- 26. If *updatedConfiguration*.OutputToken does not equal to *compatibleConfigurationsList*[0]. token.OutputToken, FAIL the test and go to step 31.
- 27. If *updatedConfiguration*.SendPrimacy does not equal to *compatibleConfigurationsList*[0].token. SendPrimacy, FAIL the test and go to step 31.
- 28. If *updatedConfiguration*.OutputLevel does not equal to *validOutputLevel*, FAIL the test and go to step 31.
- 29. ONVIF Client invokes RemoveAudioOutputConfiguration with parameters:
 - ProfileToken := profileToken
- 30. The DUT responds with RemoveAudioOutputConfigurationResponse message.
- 31. If *profileToRestore* was returned by HelperCreateEmptyProfile, ONVIF Cliend restore media profile (in *profileToRestore*) by following the procedure mentioned in Annex A.7 to restore DUT configuration.
- 32. If *profileToDelete* was returned by HelperCreateEmptyProfile, ONVIF Cliend delete media profile (in *profileToDelete*) by following the procedure mentioned in Annex A.8 to restore DUT configuration.

Test Result:

PASS -

The DUT passed all assertions.

FAIL –

The DUT did not send GetAudioOutputsResponse message.

The DUT did not send GetCompatibleAudioOutputConfigurationsResponse message.

The DUT did not send AddAudioOutputConfigurationResponse message.

The DUT did not send GetAudioOutputConfigurationOptionsResponse message.

The DUT did not send the SOAP 1.2 fault message (InvalidArgVal/ConfigModify) for invalid **SetAudioOutputConfiguration** request.

The DUT did not send **SetAudioOutputConfigurationResponse** message.

The DUT did not send GetAudioOutputConfigurationResponse message.

The DUT did not send RemoveAudioOutputConfigurationResponse message.

4.8.6 AUDIO DECODER CONFIGURATION

Test Label: Audio Decoder Configuration verification

Test Case ID: MEDIA-3-4-10

ONVIF Core Specification Coverage: Audio Decoder Configuration (ONVIF Media Service Specification), Get compatible audio decoder configurations (ONVIF Media Specification), Add audio decoder configuration to a profile (ONVIF Media Specification), GetProfile (ONVIF Media Specification), Remove audio decoder configuration from a profile (ONVIF Media Specification).

Command Under Test: GetAudioDecoderConfigurations, GetAudioDecoderConfiguration, GetCompatibleAudioDecoderConfigurations, AddAudioDecoderConfiguration, GetProfile, RemoveAudioDecoderConfiguration

WSDL Reference: media.wsdl

Test Purpose: To verify Add Audio Decoder Configuration and Remove Audio Decoder Configuration Operations

Pre-Requisite: Media service is supported by the DUT. Audio output is supported by the DUT. Media Service is received from the DUT.

Test Configuration: ONVIF Client and DUT

Test Sequence:

ONVI	⁻ Client	DUT
Create or select empty media profile	HelperCreateEmptyProfile (in <i>profileName,</i> out <i>profileToken</i> , out <i>profileToDelete,</i> out <i>profileToRestore</i>) Annex A.6	Send empty profile token
Receive and Validate	GetCompatibleAudioOutputConfigurations (ProfileToken := profileToken) GetCompatibleAudioOutputConfigurations Response (Configurations =: compatibleConfigurationsList)	Send Audio Output Configurations compatible with the received media profile
Compatible Audio Output Configurations Add audio Output configuration to new media profile	 AddAudioOutputConfiguration (ProfileToken := profileToken, ConfigurationToken := configurationToken) 	
Receive and Validate response	AddAudioOutputConfigurationResponse	Add audio Output configuration and send response
Receive and Validate Audio Decoder configurations	GetAudioDecoderConfigurationsResponse (Configurations =: audioDecoderConfigurationsList)	Send audio decoder configurations
Receive and Validate Compatible Audio Decored Configurations	(ProfileToken := <i>profileToken</i>) GetCompatibleAudioDecoderConfigurations Response (Configuration =: <i>compatibleAudioDecoderConfigurations</i>) ◀	Send Audio Decoder Configurations compatible with the received media profile



-225-

Test Procedure:

- 1. Start an ONVIF Client.
- 2. Start the DUT.
- 3. Set the following:
 - profileName := "TestProfile"
- 4. ONVIF Client generates creation or selection of profile with name (in *profileName*) and with token (out *profileToken*) and retrieves media profile token to delete (out *profileToDelete*) or media profile to restore (out *profileToRestore*) by following the procedure mentioned in Annex

A.6.

- 5. ONVIF Client invokes GetCompatibleAudioOutputConfigurations with parameters:
 - ProfileToken := profileToken
- 6. The DUT responds with **GetCompatibleAudioOutputConfigurationsResponse** with parameters:
 - Configurations =: compatibleAudioOutputConfigs
- 7. If *compatibleAudioOutputConfigs* is empty, FAIL the test and and go to step 30.
- 8. Set the following:
 - audioOutputConfigToken := compatibleAudioOutputConfigs[0].token
- 9. ONVIF Client invokes AddAudioOutputConfiguration with parameters:
 - ProfileToken := profileToken
 - ConfigurationToken := audioOutputConfigToken
- 10. The DUT responds with AddAudioOutputConfigurationResponse message.
- 11. ONVIF Client invokes GetAudioDecoderConfigurations.
- 12. The DUT responds with **GetAudioDecoderConfigurationsResponse** with parameters:
 - Configurations =: audioDecoderConfigsList
- 13. If *audioDecoderConfigsList* is empty, FAIL the test and go to step 30.
- 14. ONVIF Client invokes GetCompatibleAudioDecoderConfigurations with parameters:
 - ProfileToken := profileToken
- 15. The DUT responds with **GetCompatibleAudioDecoderConfigurationsResponse** with parameters:
 - Configuration =: compatibleAudioDecoderConfigs
- 16. If *compatibleAudioDecoderConfigs* is empty, FAIL the test and go to step 30.
- 17. Set the following:
 - audioDecoderConfigToken := compatibleAudioDecoderConfigs[0].token
- 18. ONVIF Client invokes AddAudioDecoderConfiguration with parameters:
 - ProfileToken := profileToken
 - ConfigurationToken := audioDecoderConfigToken
- 19. The DUT responds with AddAudioDecoderConfiguration message.
- 20. ONVIF Client invokes GetProfile with parameters:
 - ProfileToken := profileToken

- 21. The DUT responds with **GetProfileResponse** with parameters:
 - Profile =: *profile*
- 22. If *profile* does not contain Extension.AudioOutputConfiguration element, FAIL the test and go to step 30.
- 23. If *profile*.Extension.AudioOutputConfiguration.token value does not equal to *audioOutputConfigToken*, FAIL the test and go to step 30.
- 24. If *profile* does not contain Extension.AudioDecoderConfiguration element, FAIL the test and go to step 30.
- 25. If *profile*.Extension.AudioDecoderConfiguration.token value does not equal to *audioDecoderConfigToken*, FAIL the test and go to step 30.
- 26. ONVIF Client invokes RemoveAudioDecoderConfiguration with parameters:
 - ProfileToken := profileToken
- 27. The DUT responds with RemoveAudioDecoderConfigurationResponse message.
- 28. ONVIF Client invokes RemoveAudioOutputConfiguration with parameters:
 - ProfileToken := profileToken
- 29. The DUT responds with **RemoveAudioOutputConfigurationResponse** message.
- 30. If *profileToRestore* was returned by HelperCreateEmptyProfile, ONVIF Cliend restore media profile (in *profileToRestore*) by following the procedure mentioned in Annex A.7 to restore DUT configuration.
- 31. If *profileToDelete* was returned by HelperCreateEmptyProfile, ONVIF Cliend delete media profile (in *profileToDelete*) by following the procedure mentioned in Annex A.8 to restore DUT configuration.

Test Result:

PASS -

The DUT passed all assertions.

FAIL -

The DUT did not send GetCompatibleAudioOutputConfigurationsResponse message.

The DUT did not send AddAudioOutputConfigurationResponse message.

The DUT did not send GetAudioDecoderConfigurationsResponse message.

The DUT did not send GetCompatibleAudioDecoderConfigurationsResponse message.

The DUT did not send AddAudioDecoderConfiguration message.

The DUT did not send GetProfile message.

The DUT did not send RemoveAudioDecoderConfigurationResponse message.

The DUT did not send RemoveAudioOutputConfigurationResponse message.



4.8.7 AUDIO OUTPUT CONFIGURATIONS AND AUDIO OUTPUT CONFIGURATION OPTIONS CONSISTENCY

-228-

Test Label: GetAudioOutputConfigurations Command and GetAudioOutputConfigurationOptions Consistency Validation

Test Case ID: MEDIA-3-4-11

ONVIF Core Specification Coverage: Get Audio Output Configurations (ONVIF Media Specification), Get Audio Output Configuration Options (ONVIF Media Specification).

Command Under Test: GetAudioOutputConfigurations, GetAudioOutputConfigurationOptions

WSDL Reference: media.wsdl

Test Purpose: To check that GetAudioOutputConfigurations command and GetAudioOutputConfigurationOptions command are consistent

Pre-Requisite: Media service is supported by the DUT. Audio output is supported by the DUT. Media Service is received from the DUT.

Test Configuration: ONVIF Client and DUT

Test Sequence:



Test Procedure:

- 1. Start an ONVIF Client.
- 2. Start the DUT.
- 3. ONVIF Client invokes GetAudioOutputConfigurations.

-229-

Driving IP-based physical security through global standardization

- 4. The DUT responds with GetAudioOutputConfigurationsResponse with parameters:
 - Configurations list =: audioOutputConfigsList •
- 5. If audioOutputConfigsList is empty, FAIL the test and skip other steps.
- 6. For each audio output configuration (audioOutputConfig) from audioOutputConfigsList repeat the following steps :
 - 6.1. ONVIF Client invokes GetAudioOutputConfigurationOptions with parameters
 - ConfigurationToken := audioOutputConfig.token
 - ProfileToken skipped

6.2. The DUT responds with GetAudioOutputConfigurationOptionsResponse with parameters:

- Options =: audioOutputOptions
- 6.3. If audioOutputOptions.OutputTokensAvailable list contains at least two equal tokens, FAIL the test and skip other steps.
- 6.4. If audioOutputOptions.OutputTokensAvailable list does not contain audioOutputConfig.OutputToken value, FAIL the test and skip other steps.
- 6.5. If audioOutputConfig contains SendPrimacy element, check the following:
 - 6.5.1. If audioOutputOptions does not contain at least one SendPrimacyOptions element, FAIL the test and skip other steps.
 - audioOutputOptions.SendPrimacyOptions 6.5.2. If contain list does not audioOutputConfig.SendPrimacy value, FAIL the test and skip other steps.
- 6.6. If audioOutputConfig.OutputLevel than value is less audioOutputOptions.OutputLevelRange.Min, FAIL the test and skip other steps.
- 6.7. If audioOutputConfig.OutputLevel than value more is audioOutputOptions.OutputLevelRange.Max, FAIL the test and skip other steps.

Test Result:

PASS -

The DUT passed all assertions.

FAIL -

The DUT did not send GetAudioOutputConfigurationsResponse message.

The DUT did not send GetAudioOutputConfigurationOptionsResponse message.

PROFILES AND AUDIO OUTPUT CONFIGURATION OPTIONS CONSISTENCY 4.8.8

Test Label: GetProfiles Command and GetAudioOutputConfigurationOptions Consistency Validation

Test Case ID: MEDIA-3-4-12

ONVIF Core Specification Coverage: Get Profiles (ONVIF Media Specification), Get Audio Output Configuration Options (ONVIF Media Specification). ONVIF www.onvif.org

Command Under Test: GetProfiles, GetAudioOutputConfigurationOptions

WSDL Reference: media.wsdl

Test Purpose: To check that GetProfiles command and GetAudioOutputConfigurationOptions command are consistent

-230-

Pre-Requisite: Media service is supported by the DUT. Audio output is supported by the DUT. Media Service is received from the DUT.

Test Configuration: ONVIF Client and DUT

Test Sequence:



Test Procedure:

1. Start an ONVIF Client.

2. Start the DUT.

ONVIF

- 3. ONVIF Client invokes GetProfiles.
- 4. DUT responds with GetProfilesResponse with parameters:
 - Profiles =: *profilesList*
- 5. If *profilesList* is empty, FAIL the test and skip other steps.
- 6. For each profile item with Extension.AudioOutputConfiguration element (*profile*) from *profilesList* repeat the following steps:

-231-

- 6.1. ONVIF Client invokes **GetAudioOutputConfigurationOptions** with parameters
 - ConfigurationToken := *profile*.Extension.AudioOutputConfiguration.token
 - ProfileToken := profile.token
- 6.2. The DUT responds with GetAudioOutputConfigurationOptionsResponse with parameters:
- Options =: *audioOutputOptions*
- 6.3. If *audioOutputOptions*.OutputTokensAvailable list contains at least two equal tokens, FAIL the test and skip other steps.
- 6.4. If *audioOutputOptions*.OutputTokensAvailable token list does not contain *profile*.Extension.AudioOutputConfiguration.OutputToken value, FAIL the test and skip other steps.
- 6.5. If *profile*.Extension.AudioOutputConfiguration contains SendPrimacy element, check the following:
 - 6.5.1. If *audioOutputOptions* does not contain at least one SendPrimacyOptions element, FAIL the test and skip other steps.
 - 6.5.2. If *audioOutputOptions*.SendPrimacyOptions list does not contain *profile*.Extension.AudioOutputConfiguration.SendPrimacy value, FAIL the test and skip other steps.
- 6.6. If *profile*.Extension.AudioOutputConfiguration.OutputLevel value is less than *audioOutputOptions*.OutputLevelRange.Min, FAIL the test and skip other steps.
- 6.7. If *profile*.Extension.AudioOutputConfiguration.OutputLevel value is more than *audioOutputOptions*.OutputLevelRange.Max, FAIL the test and skip other steps.

Test Result:

PASS –

The DUT passed all assertions.

FAIL –

The DUT did not send GetProfilesResponse message.

The DUT did not send GetAudioOutputConfigurationOptionsResponse message.

4.8.9 PROFILES AND AUDIO OUTPUT CONFIGURATIONS CONSISTENCY

Test Label: GetProfiles Command and GetAudioOutputConfigurations Consistency Validation

-232-

Test Case ID: MEDIA-3-4-13

ONVIF Core Specification Coverage: Get media profiles (ONVIF Media Specification), Get audio output configurations (ONVIF Media Specification).

Command Under Test: GetProfiles, GetAudioOutputConfigurations

WSDL Reference: media.wsdl

Test Purpose: To check that GetAudioOutputConfigurations command and GetProfiles command are consistent

Pre-Requisite: Media service is supported by the DUT. Audio output is supported by the DUT. Media Service is received from the DUT.

Test Configuration: ONVIF Client and DUT

Test Sequence:



Test Procedure:

- 1. Start an ONVIF Client.
- 2. Start the DUT.
- 3. ONVIF Client invokes GetProfiles.
- 4. DUT responds with GetProfilesResponse with parameters:
 - Profiles =: profilesList
- 5. If *profilesList* is empty, FAIL the test and skip other steps.

6. ONVIF Client invokes GetAudioOutputConfigurations.

- 7. DUT responds with GetAudioOutputConfigurationsResponse with parameters:
 - Configurations list =: audioOutputConfigsList
- 8. If *audioOutputConfigsList* is empty, FAIL the test and skip other steps.
- 9. If *audioOutputConfigsList* contains at least two AudioOutputConfiguration items with equal token, FAIL the test and skip other steps.
- 10. For each profile item with Extension.AudioOutputConfiguration element (*profile*) from *profilesList* repeat the following steps:
 - 10.1. If *audioOutputConfigsList* does not contain AudioOutputConfiguration item with token value equals to *profile*.Extension.AudioOutputConfiguration.token, FAIL the test and skip other steps.
 - 10.2. If *profile*.Extension.AudioOutputConfiguration item does not have equal field values to *audioOutputConfigsList* [token = *profile*.Extension.AudioOutputConfiguration.token] item, FAIL the test and skip other steps.

Test Result:

PASS –

DUT passes all assertions.

FAIL –

DUT did not send **GetProfilesResponse** message.

DUT did not send GetAudioOutputConfigurationsResponse message.

Note: The following fields are comparied at step 10.2:

- AudioOutputConfiguration:
 - o Name
 - UseCount
 - o token
 - OutputToken
 - SendPrimacy
 - OutputLevel

4.8.10 AUDIO OUTPUT CONFIGURATIONS AND AUDIO OUTPUT CONFIGURATION CONSISTENCY

Test Label: GetAudioOutputConfigurations Command and GetAudioOutputConfiguration Consistency Validation

Test Case ID: MEDIA-3-4-14

ONVIF Core Specification Coverage: Get Audio Output Configurations (ONVIF Media Specification), Get Audio Output Configuration (ONVIF Media Specification).

Command Under Test: GetAudioOutputConfigurations, GetAudioOutputConfiguration

WSDL Reference: media.wsdl

ONVIF

check Test Purpose: То that GetAudioOutputConfigurations command and GetAudioOutputConfiguration command are consistent

Pre-Requisite: Media service is supported by the DUT. Audio output is supported by the DUT. Media Service is received from the DUT.

Test Configuration: ONVIF Client and DUT

Test Sequence:

ONVIF Client		DUT
Retrieve audio output	GetAudioOutputConfigurations	
configurations Receive and	GetAudioOutputConfigurationsResponse (Configurations =: <i>audioOutputConfigsList</i>)	Send list of audio output configurations
validate response	GetAudioOutputConfiguration	
Retrieve audio output	(ConfigurationToken := audioOutputConfig.token)	
configuration Receive and	GetAudioOutputConfigurationResponse (Configuration =: <i>configuration</i>)	Send audio output configuration
validate response		
	GetAudioOutputConfiguration	
Retrieve audio output	(ConfigurationToken := audioOutputConfig.token)	
configuration Receive and	GetAudioOutputConfigurationResponse (Configuration =: <i>configuration</i>)	Send audio output configuration
validate response	•	-

Test Procedure:

- 1. Start an ONVIF Client.
- 2. Start the DUT.
- 3. ONVIF Client invokes GetAudioOutputConfigurations.
- 4. DUT responds with GetAudioOutputConfigurationsResponse with parameters: ONVIF



- Configurations =: audioOutputConfigsList
- 5. If *audioOutputConfigsList* is empty, FAIL the test and skip other steps.
- 6. For each AudioOutputConfiguration item (*audioOutputConfig*) from *audioOutputConfigsList* repeat the following steps :

6.1. ONVIF Client invokes GetAudioOutputConfiguration with parameters

- ConfigurationToken := *audioOutputConfig*.token
- 6.2. DUT responds with **GetAudioOutputConfigurationResponse** with parameters:
 - Configuration =: *configuration*
- 6.3. If *audioOutputConfig* item does not have equal field values to *configuration* item, FAIL the test.

Test Result:

PASS –

DUT passes all assertions.

FAIL –

DUT did not send GetAudioOutputConfigurationsResponse message.

DUT did not send GetAudioOutputConfigurationResponse message.

Note: The following fields are comparied at step 6.3:

- AudioOutputConfiguration:
 - o Name
 - o UseCount
 - o token
 - OutputToken
 - SendPrimacy
 - OutputLevel

4.8.11 AUDIO OUTPUT CONFIGURATIONS AND AUDIO OUTPUTS CONSISTENCY

Test Label: GetAudioOutputConfigurations Command and GetAudioOutputs Consistency Validation

Test Case ID: MEDIA-3-4-15

ONVIF Core Specification Coverage: Get Audio Output Configurations (ONVIF Media Specification), Get Audio Outputs (ONVIF Media Specification).

Command Under Test: GetAudioOutputConfigurations, GetAudioOutputs

WSDL Reference: media.wsdl

Test Purpose: To check that GetAudioOutputConfigurations command and GetAudioOutputs command are consistent

Pre-Requisite: Media service is supported by the DUT. Audio output is supported by the DUT. Media Service is received from the DUT.

-236-

Test Configuration: ONVIF Client and DUT

Test Sequence:



Test Procedure:

- 1. Start an ONVIF Client.
- 2. Start the DUT.
- 3. ONVIF Client invokes GetAudioOutputConfigurations.
- 4. DUT responds with GetAudioOutputConfigurationsResponse with parameters:
 - Configurations =: audioOutputConfigsList
- 5. If *audioOutputConfigsList* is empty, FAIL the test and skip other steps.
- 6. ONVIF Client invokes GetAudioOutputs.
- 7. DUT responds with GetAudioOutputsResponse with parameters:
 - AudioOutputs list =: audioOutputsList
- 8. If *audioOutputsList* is empty, FAIL the test and skip other steps.
- 9. If *audioOutputsList* contains at least two items with equal tokens, FAIL the test and skip other steps.



10. For each AudioOutputConfiguration item (*audioOutputConfig*) from *audioOutputConfigsList* repeat the following:

-237-

10.1. If *audioOutputsList* does not contain token equals to *audioOutputConfig*.OutputToken, FAIL the test.

Test Result:

PASS –

DUT passes all assertions.

FAIL -

DUT did not send GetAudioOutputConfigurationsResponse message.

DUT did not send GetAudioOutputsResponse message.

4.8.12 PROFILES AND AUDIO DECODER CONFIGURATIONS CONSISTENCY

Test Label: GetProfiles Command and GetAudioDecoderConfigurations Consistency Validation

Test Case ID: MEDIA-3-4-16

ONVIF Core Specification Coverage: Get media profiles (ONVIF Media Specification), Get audio decoder configurations (ONVIF Media Specification).

Command Under Test: GetProfiles, GetAudioDecoderConfigurations

WSDL Reference: media.wsdl

Test Purpose: To check that GetAudioDecoderConfigurations command and GetProfiles command are consistent

Pre-Requisite: Media service is supported by the DUT. Audio output is supported by the DUT. Media Service is received from the DUT.

Test Configuration: ONVIF Client and DUT

Test Sequence:



-238-

Test Procedure:

- 1. Start an ONVIF Client.
- 2. Start the DUT.
- 3. ONVIF Client invokes GetProfiles.
- 4. DUT responds with GetProfilesResponse with parameters:
 - Profiles =: profilesList
- 5. If *profilesList* is empty, FAIL the test and skip other steps.
- 6. ONVIF Client invokes GetAudioDecoderConfigurations.
- 7. DUT responds with GetAudioDecoderConfigurationsResponse with parameters:
 - Configurations list =: audioDecoderConfigsList
- 8. If *audioDecoderConfigsList* is empty, FAIL the test and skip other steps.
- 9. If *audioDecoderConfigsList* contains at least two AudioDecoderConfiguration items with equal token, FAIL the test and skip other steps.
- 10. For each profile item with Extension.AudioDecoderConfiguration element (*profile*) from *profilesList* repeat the following steps:
 - 10.1. If *audioDecoderConfigsList* does not contain AudioDecoderConfiguration item with token value equals to *profile*.Extension.AudioDecoderConfiguration.token, FAIL the test and skip other steps.
 - 10.2. If *profile*.Extension.AudioDecoderConfiguration item does not have equal field values to *audioDecoderConfigsList* [token = *profile*.Extension.AudioDecoderConfiguration.token] item, FAIL the test and skip other steps.

Test Result:

PASS –

DUT passes all assertions.

FAIL -

DUT did not send **GetProfilesResponse** message.

DUT did not send GetAudioDecoderConfigurationsResponse message.

Note: The following fields are comparied at step 10.2:

- AudioDecoderConfiguration:
 - o Name
 - UseCount
 - o token

4.8.13 AUDIO DECODER CONFIGURATIONS AND AUDIO DECODER CONFIGURATION CONSISTENCY

Test Label: GetAudioDecoderConfigurations Command and GetAudioDecoderConfiguration Consistency Validation

Test Case ID: MEDIA-3-4-17

ONVIF Core Specification Coverage: Get Audio Decoder Configurations (ONVIF Media Specification), Get Audio Decoder Configuration (ONVIF Media Specification).

Command Under Test: GetAudioDecoderConfigurations, GetAudioDecoderConfiguration

WSDL Reference: media.wsdl

Test Purpose: To check that GetAudioDecoderConfigurations command and GetAudioDecoderConfiguration command are consistent

Pre-Requisite: Media service is supported by the DUT. Audio output is supported by the DUT. Media Service is received from the DUT.

Test Configuration: ONVIF Client and DUT

Test Sequence:

ONVIF Client		DUT	
Retrieve audio decoder	GetAudioDecoderConfigurations		
configurations Receive and	GetAudioDecoderConfigurationsResponse (Configurations =: <i>audioDecoderConfigsList</i>	t)	Send list of audio decoder configurations
validate response	GetAudioDecoderConfiguration		
Retrieve audio decoder	(ConfigurationToken := audioDecoderConfig.token)		
configuration Receive and validate response	GetAudioDecoderConfigurationResponse (Configuration =: <i>configuration</i>)		Send audio decoder configuration
	· 		
	GetAudioDecoderConfiguration		
Retrieve audio	(ConfigurationToken := audioDecoderConfig.token)		
configuration	GetAudioDecoderConfigurationResponse (Configuration =: <i>configuration</i>)	→	Send audio decoder configuration
validate response	<	—	

Test Procedure:

- 1. Start an ONVIF Client.
- 2. Start the DUT.
- 3. ONVIF Client invokes GetAudioDecoderConfigurations.
- 4. The DUT responds with GetAudioDecoderConfigurationsResponse with parameters:
 - Configurations list =: audioDecoderConfigsList
- 5. If *audioDecoderConfigsList* is empty, FAIL the test and skip other steps.
- 6. For each AudioDecoderConfiguration item (*audioDecoderConfig*) from *audioDecoderConfigsList* repeat the following steps :
 - 6.1. ONVIF Client invokes GetAudioDecoderConfiguration with parameters

- ConfigurationToken := audioDecoderConfig.token
- 6.2. The DUT responds with GetAudioDecoderConfigurationResponse with parameters:
 - Configuration =: configuration
- 6.3. If *audioDecoderConfig* item does not have equal field values to *configuration* item, FAIL the test.

Test Result:

PASS –

The DUT passed all assertions.

FAIL –

The DUT did not send GetAudioDecoderConfigurationsResponse message.

The DUT did not send GetAudioDecoderConfigurationResponse message.

Note: The following fields are comparied at step 6.3:

- AudioDecoderConfiguration:
 - o Name
 - UseCount
 - o token

4.9 PTZ Configuration

4.9.1 PTZ CONFIGURATIONS AND PROFILES CONSISTENCY

Test Label: Media Service and PTZ Service DUT GetConfigurations Command and GetProfiles Command Consistency Validation.

Test Case ID: MEDIA-4-1-2

ONVIF Core Specification Coverage: Get media profiles, GetConfigurations.

Command Under Test: GetProfiles, GetConfigurations

WSDL Reference: media.wsdl, ptz.wsdl.

Test Propose: To check that GetPTZConfigurations command and GetProfiles command are consistent.

Pre-Requisite: PTZ is supported by DUT. ONVIF Client gets the Media Service entry point and PTZ Service entry point by GetCapabilities command.

Test Configuration: ONVIF Client and DUT

Test Sequence:

-242-



Test Procedure:

- 1. Start an ONVIF Client.
- 2. Start the DUT.
- 3. ONVIF Client invokes GetProfilesRequest message to retrieve list of profiles and their PTZ configurations.
- 4. The DUT sends GetProfilesResponse message.
- 5. ONVIF Client invokes GetConfigurations message to retrieve list of PTZ Configurations from device.
- 6. The DUT sends GetConfigurationsResponse message.
- 7. Check that each PTZConfiguration from GetConfigurationsResponse message has unique token.
- 8. Check that all PTZConfigurations from the GetProfilesResponse message are included in list the GetConfigurationsResponse message.
- 9. Check that PTZConfiguration parameters are same in the GetProfilesResponse message and in the GetConfigurationsResponse message.

Test Result:

PASS -

DUT passes all assertions.

FAIL -

The DUT did not send GetProfilesResponse message.

The DUT did not send valid GetProfilesResponse message.

The DUT did not send GetConfigurationsResponse message.

The DUT did not send valid GetConfigurationsResponse message.

The DUT return two or more PTZConfigurations in GetConfigurationsResponse message with the same ConfigurationToken.

The DUT returned the GetProfilesResponse message with PTZConfigurations that were not included in the GetConfigurationsResponse message.

The DUT returned different parameters list and parameters values for the same PTZConfiguration in the GetConfigurationsResponse message and in the GetProfilesResponse message.

4.9.2 PTZ CONFIGURATION

Test Label: Media Configuration PTZ Configuration

Test Case ID: DRAFT-MEDIA-4-1-4

ONVIF Core Specification Coverage: Get media profiles, Add PTZ configuration to a profile, Remove PTZ configuration from a profile, Delete media profile, GetConfigurations.

Command Under Test: None

WSDL Reference: media.wsdl

Test Purpose: To verify DUT PTZ Configuration Operations

Pre-Requisite: PTZ is supported by the DUT

Test Configuration: ONVIF Client and DUT

Test Sequence:

ONVIF Client		DUT	
	GetConfigurations (Empty)	Start DUT	
Pacaiva PT7	GetConfigurationsResponse (PTZ Configurations)	Send list of existing PTZ Configurations	
Configurations	CreateProfileRequest (Name = "testprofileX")		
Create new media profile	CreateProfileResponse (Empty profile with token) or SOAP 1.2 fault message (Action/MaxNVTProfiles)	h Send create profile response	
	Annex A.3		
Add PTZ configuration to new media profile	AddPTZConfigurationRequest (Profile Token, PTZ cfg token)		
	AddPTZConfigurationResponse (empty message)	Add PTZ configuration and send response	
	RemovePTZConfigurationRequest (Profile Token)	Remove PTZ Cfg from	
Possive the message	RemovePTZConfigurationResponse (empty message)	→ media prome.	
Receive the message	•		
Send Delete Profile request	DeleteProfileRequest (Profile Token)	_	
	DeleteProfileResponse (Empty message)	Delete media profile.	
	←		

Test Procedure:

- 1. Start an ONVIF Client.
- 2. Start the DUT.
- 3. ONVIF Client invokes GetConfigurations request on PTZ service of DUT to retrieve the list of PTZ configurations supported by DUT.

-245-

- 4. ONVIF Client validates the GetConfigurationsResponse message sent by the DUT. At least one PTZ configuration should be present in the response.
- 5. ONVIF Client invokes CreateProfile request with ProfileToken = 'testprofileX'.
- 6. Verify CreateProfileResponse message or SOAP 1.2 fault message (Action/MaxNVTProfiles) from the DUT. If fault was received, execute Annex A.3.
- 7. ONVIF Client invokes AddPTZConfiguration request with ProfileToken as 'testprofileX' and ConfigurationToken as one of the existing PTZConfiguration tokens.
- 8. The DUT adds the PTZ configuration to the profile and sends the response.
- 9. ONVIF Client invokes RemovePTZConfiguration request with ProfileToken as 'testprofileX'.
- 10. The DUT removes the PTZ configuration from media profile and sends the response.
- 11. If profile with profile token = testprofileX was created during test execution, then ONVIF Client invokes DeleteProfile request with ProfileToken as 'testprofileX'. Otherwise, ONVIF client skip rest steps and restore profile settings.
- 12. The DUT deletes the media profile and sends the response.

Test Result:

PASS –

The DUT passed all assertions.

FAIL -

The DUT did not send valid GetConfigurationsResponse message.

The DUT does not have PTZ configuration.

The DUT did not send AddPTZConfigurationResponse message.

The DUT did not send RemovePTZConfigurationResponse message.

The DUT did not send DeleteProfileResponse message.

Note: See Annex A.5 for Name and Token Parameters Length limitations.

Note: If profile was deleted during Annex A.3 execution, ONVIF Client restores the deleted profile and profile settings.

4.10 Metadata Configuration

4.10.1 METADATA CONFIGURATION

Test Label: Media Configuration Metadata Configuration

Test Case ID: DRAFT-MEDIA-5-1-3

ONVIF Core Specification Coverage: Create media profile, Add metadata configuration to a profile, Remove metadata configuration from a profile, Delete media profile, Get metadata configurations, Get metadata configuration, Get compatible metadata configurations, Get metadata configuration options, Modify a metadata configuration.

Command Under Test: None

WSDL Reference: media.wsdl

Test Purpose: To verify DUT Metadata Configuration Operations

Pre-Requisite: None

Test Configuration: ONVIF Client and DUT

Test Sequence:

-247-

Onvir Driving IP-based physical security through global standardization

ONVIF	- Client	DUT	
Create new media	CreateProfileRequest (Name = "testprofileX")		Start DUT
prome	CreateProfileResponse (Empty profile wir token) or SOAP 1.2 fault message (Action/MaxNVTProfiles)	th	Send create profile
	Annex A.3		response
	GetMetadataConfigurationsRequest (Empty)		
Receive and Validate Metadata	GetMetadataConfigurationsResponse (Metadata Configurations)		Send all Metadata Configurations
Configurations	GetCompatibleMetadataConfigurations Request (Media Profile token)		Send Metadata
Receive and Validate Metadata Configurations	GetCompatibleMetadataConfigurations Response (Metadata Configurations)	→	Configurations compatible with the received media profile
Add Metadata	AddMetadataConfigurationRequest (Prof Token, Metadata cfg)	ile	
media profile	AddMetadataConfigurationResponse (empty message)		Add Metadata configuration and send response
	 GetMetadataConfigurationOptionsReque (Metadata Cfg) 	st	Send configurable options for Metadata
Receive and Validate Metadata configurable options	GetMetadataConfigurationOptions Response (Metadata Cfg options)		comguration

-248-

Onvir Driving IP-based physical security through global standardization

ONVIF Client		DUT
Set Invalid Metadata cfg values	SetMetadataConfigurationRequest (Invalio Metadata cfg values, FALSE)	
	SOAP Fault Message (InvalidArgVal/ConfigModify)	Send SOAP fault message
Set Metadata Configuration	SetMetadataConfigurationRequest (Metadata Cfg, FALSE)	
	SetMetadataConfigurationResponse (Empty)	Modify Metadata Configuration
	 GetMetadataConfigurationRequest (Metadata Cfg Token) 	
Receive and Validate Metadata	GetMetadataConfigurationResponse (Metadata Cfg)	Send the modified Metadata Configuration
Coniguration	RemoveMetadataConfigurationRequest (Profile Token)	Remove Metadata Cfg
Receive the message	RemoveMetadataConfigurationResponse (empty message)	
Send Delete Profile request	DeleteProfileRequest (Profile Token)	
	DeleteProfileResponse (Empty message)	Delete media profile.

-249-

Driving IP-based physical security through global standardization

Test Procedure:

- 1. Start an ONVIF Client.
- 2. Start the DUT.
- 3. ONVIF Client invokes CreateProfile request with ProfileToken = 'testprofileX'.
- 4. Verify CreateProfileResponse message or SOAP 1.2 fault message (Action/MaxNVTProfiles) from the DUT. If fault was received, execute Annex A.3.
- 5. ONVIF Client will invoke GetMetadataConfigurations request to retrieve the list of metadata configurations supported by the DUT.
- 6. ONVIF Client verifies the list of metadata configurations sent by the DUT.
- 7. ONVIF Client invokes GetCompatibleMetadataConfigurations request with 'testprofileX' as ProfileToken.
- 8. The DUT sends the list of metadata configurations compatible with the received media profile token.
- 9. ONVIF Client invokes AddMetadataConfiguration request message with ProfileToken as 'testprofileX' and ConfigurationToken as one of the metadata configuration tokens received in the GetCompatibleMetadataConfigurations response.
- 10. The DUT adds the Metadata configuration to the profile and sends the response.
- 11. ONVIF Client invokes GetMetadataConfigurationOptions request with ConfigurationToken as same token in AddMetadataConfiguration request.
- 12. The DUT sends the configurable options supported for the received metadata configuration.
- 13. ONVIF Client invokes SetMetadataConfiguration request with metadata configuration values outside the range defined in GetMetadataConfigurationOptions response and 'ForcePersistence' flag as 'FALSE'.
- 14. The DUT sends the SOAP 1.2 fault message (InvalidArgVal/ConfigModify).
- 15. ONVIF Client verifies the SOAP fault message sent by DUT.
- 16. ONVIF Client invokes SetMetadataConfiguration request with metadata configuration values within the range defined in GetMetadataConfigurationOptions response and 'ForcePersistence' flag as 'FALSE'.
- 17. The DUT modifies the metadata configuration and sends the SetMetadataConfiguration indicating success.
- 18. ONVIF Client verifies the modified Metadata configuration by invoking the GetMetadataConfigurations request.
- 19. The DUT sends the modified Metadata configuration in GetMetadataConfiguration response.
- 20. ONVIF Client invokes RemoveMetadataConfiguration request with ProfileToken as 'testprofileX'.
- 21. The DUT removes the Metadata configuration token from media profile and sends the response.



- 22. If profile with profile token = testprofileX was created during test execution, then ONVIF Client invokes DeleteProfile request with ProfileToken as 'testprofileX'. Otherwise, ONVIF client skip rest steps and restore profile settings.
- 23. The DUT deletes the media profile and sends the response.

Test Result:

PASS -

The DUT passed all assertions.

FAIL -

The DUT did not send CreateProfileResponse message.

The DUT did not send GetMetadataConfigurationsResponse message.

The DUT did not send GetCompatibleMetadataConfigurationsResponse message.

The DUT did not send AddMetadataConfigurationResponse message.

The DUT did not send GetMetadataConfigurationOptionsResponse message.

The DUT did not send the SOAP 1.2 fault message (InvalidArgVal/ConfigModify) for invalid SetMetadataConfiguration request.

The DUT did not send SetMetadataConfigurationResponse message.

The DUT did not send GetMetadataConfigurationResponse message.

The DUT did not modify metadata configuration correctly.

The DUT did not send RemoveMetadataConfigurationResponse message.

The DUT did not send DeleteProfileResponse message.

Note: See Annex A.5 for Name and Token Parameters Length limitations.

Note: If profile was deleted during Annex A.3 execution, ONVIF Client restores the deleted profile and profile settings.

4.11 Media Streaming

4.11.1 SNAPSHOT URI

Test Label: Media Configuration Snapshot URI

Test Case ID: MEDIA-6-1-1

ONVIF Core Specification Coverage: Get media profiles, Request snapshot URI.

Command Under Test: GetSnapshotUri

WSDL Reference: media.wsdl

Test Purpose: To retrieve snapshot URI of DUT for given media profile

Pre-Requisite: None

Test Configuration: ONVIF Client and DUT

Test Sequence:

ONVIF Client		DUT	DUT	
	GetProfilesRequest (Empty)		Start DUT	
	GetProfilesResponse (Media Profiles)	-	Send list of existing media profiles	
Receive and validate media profiles	<			
	GetSnapshotUriRequest (Profile Token)			
Retrieve URI for snapshot				
	GetSnapshotUriResponse (URI)			
			Send URI for snap shot	
Invoke HTTP GET request on snapshot uri	HTTP GET Request		Send the JPEG image as	
	200 OK (JPEG Image)	-	HTTP response	
	<			

www.onvif.org

Test Procedure:

- 1. Start an ONVIF Client.
- 2. Start the DUT.
- 3. ONVIF Client invokes GetProfiles request to retrieve the list of existing media profiles on DUT.
- 4. ONVIF Client validates the GetProfilesResponse message sent by the DUT. At least one media profile should be present with video source and video encoder.
- 5. ONVIF Client invokes GetSnapshotUri request with ProfileToken as one of the media profile tokens received in GetProfilesResponse message.
- 6. DUT sends HTTP URI and parameters defining the lifetime of the URI like ValidUntilConnect, ValidUntilReboot and Timeout in the GetSnapshotUriResponse message.
- 7. ONVIF Client invokes HTTP GET request on the snapshot URI sent by DUT.
- 8. DUT sends 200 OK message and the single shot JPEG image data.
- 9. ONVIF Client verifies the JPEG image sent by the DUT.

Test Result:

PASS -

DUT passes all assertions.

FAIL -

DUT did not send GetProfilesResponse message.

DUT did not send GetSnapshotUriResponse message.

DUT did not send one or more mandatory parameters in the GetSnapshotUriResponse message (mandatory parameters –Uri, ValidUntilConnect, ValidUntilReboot and Timeout).

DUT did not send 200 OK message.

DUT did not send valid JPEG image data.
4.12 Error Handling

4.12.1 SOAP FAULT MESSAGE

Test Label: Media Configuration DUT generates SOAP 1.2 fault message for Invalid GetStreamUriRequest Message.

Test Case ID: MEDIA-7-1-2

ONVIF Core Specification Coverage: Request stream URI.

Command Under Test: GetStreamUri

WSDL Reference: media.wsdl

Test Purpose: To verify that DUT generates SOAP 1.2 fault message to the invalid GetStreamUriRequest message (Invalid Transport).

Pre-Requisite: Media service is supported by DUT. Real-time Streaming is supported by DUT. Media service was received from the DUT.

Test Configuration: ONVIF Client and DUT

Test Sequence:

ONVIF Client		DUT
	GetStreamUriRequest (Profile Token, RTP- Unicast, RTP)	Start DUT
Receive and Validate SOAP 1.2 fault response	SOAP 1.2 fault response	Send SOAP 1.2 fault response

Test Procedure:

- 1. Start an ONVIF Client.
- 2. Start the DUT.
- 3. ONVIF Client will invoke GetStreamUriRequest message with invalid Transport (RTP).
- 4. DUT will generate the SOAP 1.2 fault message.

Test Result:

PASS -

DUT passes all assertions.

FAIL –

The DUT did not send SOAP 1.2 fault message.

The DUT did not send correct SOAP 1.2 fault message (fault code, namespace etc).

Note: See Annex A.2 for correct syntax for the StreamSetup element in GetStreamUri requests.

4.12.2 SOAP FAULT MESSAGE

Test Label: Media Configuration DUT generates SOAP 1.2 fault message for Invalid GetStreamUriRequest Message.

Test Case ID: MEDIA-7-1-4

ONVIF Core Specification Coverage: Request stream URI.

Command Under Test: GetStreamUri

WSDL Reference: media.wsdl

Test Purpose: To verify that DUT generates SOAP 1.2 fault message to the invalid GetStreamUriRequest message (Invalid Media Profile).

Pre-Requisite: Media service is supported by DUT. Real-time Streaming is supported by DUT. Media service was received from the DUT.

Test Configuration: ONVIF Client and DUT

Test Sequence:

ONVIF	Client	DUT
	GetStreamUriRequest (Invalid Profile , RTP-Unicast, UDP)	Start DUT
Receive and Validate SOAP 1.2 fault response	SOAP 1.2 fault response (InvalidArgVal/NoProfile)	Send SOAP 1.2 fault

Test Procedure:

- 1. Start an ONVIF Client.
- 2. Start the DUT.
- 3. ONVIF Client will invoke GetStreamUriRequest message with invalid media profile.
- 4. DUT will generate the SOAP 1.2 fault message (InvalidArgVal/NoProfile).

-255-

Driving IP-based physical security through global standardization

Test Result:

PASS -

DUT passes all assertions.

FAIL -

The DUT did not send SOAP 1.2 fault message.

The DUT did not send correct SOAP 1.2 fault message (fault code, namespace etc).

Note: See Annex A.2 for correct syntax for the StreamSetup element in GetStreamUri requests.

Note: See Annex A.5 for Name and Token Parameters Length limitations.

4.12.3 START MULTICAST - INVALID PROFILE TOKEN

Test Label: Media DUT StartMulticastStreaming Command Validation (Invalid ProfileToken)

Test Case ID: MEDIA-7-1-5

ONVIF Core Specification Coverage: Start multicast streaming

Command Under Test: StartMulticastStreaming

WSDL Reference: media.wsdl

Test Propose: To verify the behavior of StartMulticastStreaming command in case of invalid Profile Token.

Pre-Requisite: Media service is supported by DUT. Real-time Streaming is supported by DUT. Media service was received from the DUT.

Test Configuration: ONVIF Client and DUT

Test Sequence:



Test Procedure:

1. Start an ONVIF Client.

- 2. Start the DUT.
- 3. ONVIF Client will invoke StartMulticastStreamingRequest message (invalid ProfileToken).

-256-

4. The DUT will generate a SOAP 1.2 fault message (InvalidArgVal/NoProfile or ActionNotSupported (for case when multicast is not supported)).

Test Result:

PASS –

DUT passes all assertions.

FAIL –

The DUT did not send SOAP 1.2 fault message.

The DUT sent incorrect SOAP 1.2 fault message (fault code, namespace, etc.).

Note: See Annex in [ONVIF Base Test] for Invalid SOAP 1.2 fault message definition.

Note: Other faults than specified in the test are acceptable but specified are preferable.

Note: See Annex A.5 for Name and Token Parameters Length limitations.

4.13 Capabilities

4.13.1 MEDIA SERVICE CAPABILITIES

Test Label: Media Service Capabilities Verification.

Test Case ID: MEDIA-8-1-1

ONVIF Core Specification Coverage: Capability exchange

Command Under Test: GetServiceCapabilities (for Media Service)

WSDL Reference: media.wsdl

Test Purpose: To verify Media Service Capabilities of the DUT.

Pre-Requisite: Media Service was received from the DUT.

Test Configuration: ONVIF Client and DUT

Test Sequence:

ONVIF Client

DUT

	GetServiceCapabilitiesRequest (empty)	Start DUT
Receive and Validate GetServiceCapabilities Response message	GetServiceCapabilitiesResponse (media service capabilities)	Send capabilities of media service of the DUT

Test Procedure:

- 1. Start an ONVIF Client.
- 2. Start the DUT.
- 3. ONVIF Client will invoke GetServiceCapabilitiesRequest message to retrieve media service capabilities of the DUT.
- 4. Verify the GetServiceCapabilitiesResponse from the DUT.

Test Result:

PASS –

DUT passes all assertions.

FAIL -

The DUT did not send valid GetServiceCapabilitiesResponse.

4.13.2 GET SERVICES AND GET MEDIA SERVICE CAPABILITIES CONSISTENCY

Test Label: Get Services and Media Service Capabilities Consistency Verification.

Test Case ID: MEDIA-8-1-2

ONVIF Core Specification Coverage: Capability exchange

Command Under Test: GetServices, GetServiceCapabilities (for Media Service)

WSDL Reference: devicemgmt.wsdl, media.wsdl

Test Purpose: To verify Get Services and Media Service Capabilities consistency.

Pre-Requisite: None.

Test Configuration: ONVIF Client and DUT

Test Sequence:

ONVIF Client

DUT

	GetServicesRequest (IncludeCapability = true)	Start DUT
Receive and Validate GetServicesResponse message	GetServicesResponse (supported services with capabilities)	Send all services of the DUT without capabilities
	GetServiceCapabilitiesRequest (empty)	
Receive and Validate GetServiceCapabilities Response message	GetServiceCapabilitiesResponse (media service capabilities)	Send capabilities of media service of the DUT

Test Procedure:

- 1. Start an ONVIF Client.
- 2. Start the DUT.
- 3. ONVIF Client will invoke GetServicesRequest message (IncludeCapability = true) to retrieve all services of the DUT with service capabilities.
- 4. Verify the GetServicesResponse message from the DUT.
- 5. ONVIF Client will invoke GetServiceCapabilitiesRequest message to retrieve Media service capabilities of the DUT.
- 6. Verify the GetServiceCapabilitiesResponse message from the DUT.

Test Result:

PASS -

DUT passes all assertions.

FAIL -

The DUT did not send valid GetServicesResponse message.

The DUT did not send valid GetServiceCapabilitiesResponse message.

The DUT sent different Capabilities in GetServicesResponse message and in GetServiceCapabilitiesResponse message.

Note: Service will be defined as Media service if it will have Namespace element that is equal to "http://www.onvif.org/ver10/media/wsdl".

Annex A

This section describes the meaning of the following definitions. These definitions are used in the test case description.

A.1 Invalid Media Profile

Media profile token is a string of max length value of 64.

Invalid Media Profile:

- a) A string which is not formed according to the rules of RFC 952.
- b) A string which exceeds max length value of 64.

A.2 StreamSetup syntax for GetStreamUri

The following media stream setups for GetStreamUri are covered in this Test Specification:

- 1. RTP Unicast over UDP
- 2. RTP over RTSP over HTTP over TCP
- 3. RTP over RTSP over TCP

The correct syntax for the StreamSetup element for these media stream setups are as follows:

1. RTP Unicast over UDP

<StreamSetup>

<StreamType>RTP_unicast</StreamType>

<Transport>

<Protocol>UDP</Protocol>

</Transport>

</StreamSetup>

2. RTP over RTSP over HTTP over TCP

<StreamSetup>

- <StreamType>RTP_unicast</StreamType>
- <Transport>

<Protocol>HTTP</Protocol>

</Transport>

</StreamSetup>

- 3. RTP over RTSP over TCP
- <StreamSetup>
 - <StreamType>RTP_unicast</StreamType>
 - <Transport>
 - <Protocol>RTSP</Protocol>
 - </Transport>
- </StreamSetup>

A.3 Create Empty Profile

For the execution of test cases with profile configurations, ONVIF Client has to find, create or configure empty media profile.

ONVIF Client follows the following procedure to find, create or configure empty media profile.

- 1. ONVIF Client will invoke GetProfilesRequest message to retrieve complete profiles list.
- 2. Verify the GetProfilesResponse message from the DUT.

If there is profile with fixed attribute equal to false, then follows the following procedure:

- 3. ONVIF Client will invoke DeleteProfileRequest message (ProfileToken = Token1, where Token1 is the first ProfileToken from the GetProfilesResponse with fixed = false) to delete Profile.
- 4. Verify the DeleteProfileResponse message from the DUT.
- 5. Return to step with profile creation of the test procedure.

If there are no profiles with fixed = "false" remove all configurations from one fixed profile and use this profile for test. If there are no profiles skip other steps and fail test.

Note: See Annex A.5 for Name and Token Parameters Length limitations.

A.4 Get Guaranteed Number of Video Encoder Instances and Get Video Encoder Configuration Options Consistency Verification Description

The following general rules (see below) were used to verify consistency between:

- 1. GetVideoEncoderConfigurationOptionsResponse message requested without Profile token and Video Encoder Configuration token (general options for the DUT).
- 2. GetGuaranteedNumberOfVideoEncoderInstancesResponse messages for all Video Source Configurations supported by the DUT.

For **JPEG** encoding:

- If DUT **supports JPEG** encoding (i.e. GetVideoEncoderConfigurationOptionsResponse message **contains Options.JPEG** element), then the following behavior
 - will be assumed as correct:
 - For at least one Video Source Configuration GetGuaranteedNumberOfVideoEncoderInstancesResponse message contains no JPEG elements or contains a JPEG element with value > 0.
 - will be assumed as **incorrect**:
 - For all Video Source Configurations GetGuaranteedNumberOfVideoEncoderInstancesResponse message contains a JPEG element with value = 0
- If DUT does not support JPEG encoding (i.e. GetVideoEncoderConfigurationOptionsResponse message does not contain Options.JPEG element), then the following behavior
 - will be assumed as correct:
 - For all Video Source Configurations GetGuaranteedNumberOfVideoEncoderInstancesResponse message contains no JPEG elements or contains a JPEG element with value = 0.
 - will be assumed as **incorrect**:
 - For at least one Video Source Configuration GetGuaranteedNumberOfVideoEncoderInstancesResponse message contains a JPEG element with value > 0.

For **MPEG-4** encoding:

- If DUT **supports MPEG-4** encoding (i.e. GetVideoEncoderConfigurationOptionsResponse message **contains Options.MPEG** element), then the following behavior
 - will be assumed as **correct**:
 - For at least one Video Source Configuration GetGuaranteedNumberOfVideoEncoderInstancesResponse message contains no MPEG elements or contains an MPEG element with value > 0.
 - will be assumed as **incorrect**:
 - For all Video Source Configurations GetGuaranteedNumberOfVideoEncoderInstancesResponse message contains an MPEG element with value = 0
- If DUT does not support MPEG-4 encoding (i.e. GetVideoEncoderConfigurationOptionsResponse message does not contain Options.MPEG element), then the following behavior
 - will be assumed as correct:
 - For all Video Source Configurations GetGuaranteedNumberOfVideoEncoderInstancesResponse message contains no MPEG elements or contains an MPEG element with value = 0.

- will be assumed as **incorrect**:
 - For at least one Video Source Configuration GetGuaranteedNumberOfVideoEncoderInstancesResponse message contains an MPEG element with value > 0.

For **H.264** encoding:

• If DUT **supports H.264** encoding (i.e. GetVideoEncoderConfigurationOptionsResponse message **contains Options.H264** element), then the following behavior

-263-

- will be assumed as **correct**:
 - For at least one Video Source Configuration GetGuaranteedNumberOfVideoEncoderInstancesResponse message contains no H264 elements or contains an H264 element with value > 0.
- will be assumed as **incorrect**:
 - For all Video Source Configurations GetGuaranteedNumberOfVideoEncoderInstancesResponse message contains an H264 element with value = 0
- If DUT **does not support H.264** encoding (i.e. GetVideoEncoderConfigurationOptionsResponse message **does not contain Options.H264** element), then the following behavior
 - will be assumed as **correct**:
 - For all Video Source Configurations GetGuaranteedNumberOfVideoEncoderInstancesResponse message contains no H264 elements or contains an H264 element with value = 0.
 - will be assumed as **incorrect**:
 - For at least one Video Source Configuration GetGuaranteedNumberOfVideoEncoderInstancesResponse message contains an H264 element with value > 0.

A.5 Name and Token Parameters

There are following limitations on maximum length of Name and Token parameters that sell be used during tests by ONVIF Device Test Tool to prevent faults from DUT:

- 1. Name shall be less than or equal to 64 characters (only readable characters accepted).
- 2. Token shall be less than or equal to 64 characters (only readable characters accepted).

UTF-8 character set shall be used for Name and Token.

Note: these limitations will not be used, if ONVIF Device Test Tool reuses values that were received from the DUT.

A.6 Create or configure Empty Profile

Name: HelperCreateEmptyProfile

Procedure Purpose: Helper procedure to create or configure empty media profile.

Pre-requisite: Media Service is supported by the DUT.

Input: The profile name (*profileName*) of profile to be created.

Returns: The profile token (*profileToken*), Media Profile to restore (*profileToRestore*) or Media Profile token to delete (*profileToDelete*).

- 1. ONVIF Client invokes CreateProfile with with parameters
 - Name := *profileName*
 - Token skipped
- 2. If DUT returns CreateProfileResponse message with parameters
 - Profile := *newProfile*
 - 2.1. Set the following:
 - *profileToken* := *newProfile*.token
 - profileToDelete := newProfile.token
 - 2.2. Skip other steps.
- 3. If DUT returns env:Receiver\ter:Action\ter:MaxNVTProfiles SOAP 1.2 fault:
 - 3.1. If profile was deleted during HelperCreateEmptyProfile procedure, FAIL the test and skip other steps.
 - 3.2. ONVIF Client will invoke GetProfiles message.
 - 3.3. DUT respinds with **GetProfilesResponse** with parameters:
 - Profiles =: profileList
 - 3.4. If *profileList* is empty, FAIL the test and skip other steps.
 - 3.5. If *profileList* contains profile item with Profile.fixed = false:
 - 3.5.1. Set the following:
 - profileToRestore := the first profileList[0].Profile with fixed = false
 - 3.5.2. Set the following:
 - *profileToDelete* := *profileToRestore*.token
 - 3.5.3. ONVIF Client invokes **DeleteProfile** message with parameters
 - ProfileToken := profileToDelete
 - 3.5.4. DUT responds with **DeleteProfileResponse**.
 - 3.5.5. Go to the step 1.
 - 3.6. If *profileList* does not contain profile item with Profile.fixed = false

- 3.6.1. Set the following:
 - profileToConfigure := profileList[0].Profile
 - profileToRestore := profileList[0].Profile
- 3.6.2. If *profileToConfigure* contains VideoSourceConfiguration:
 - 3.6.2.1. ONVIF Client invokes **RemoveVideoSourceConfiguration** with parameters:
 - ProfileToken := profileToConfigure.token
 - 3.6.2.2. DUT responds with RemoveVideoSourceConfigurationResponse
- 3.6.3. If profileToConfigure contains AudioSourceConfiguration
 - 3.6.3.1. ONVIF Client invokes **RemoveAudioSourceConfiguration** with parameters:
 - ProfileToken := profileToConfigure.token
 - 3.6.3.2. DUT responds with RemoveAudioSourceConfigurationResponse
- 3.6.4. If profileToConfigure contains VideoEncoderConfiguration
 - 3.6.4.1. ONVIF Client invokes **RemoveVideoEncoderConfiguration** with parameters:
 - ProfileToken := profileToConfigure.token
 - 3.6.4.2. DUT responds with RemoveVideoEncoderConfigurationResponse
- 3.6.5. If profileToConfigure contains AudioEncoderConfiguration
 - 3.6.5.1. ONVIF Client invokes **RemoveAudioEncoderConfiguration** with parameters:
 - ProfileToken := profileToConfigure.token
 - 3.6.5.2. DUT responds with RemoveAudioEncoderConfigurationResponse
- 3.6.6. If profileToConfigure contains VideoAnalyticsConfiguration
 - 3.6.6.1. ONVIF Client invokes RemoveVideoAnalyticsConfiguration with parameters:
 - ProfileToken := profileToConfigure.token
 - 3.6.6.2. DUT responds with RemoveVideoAnalyticsConfigurationResponse
- 3.6.7. If profileToConfigure contains PTZConfiguration
 - 3.6.7.1. ONVIF Client invokes **RemovePTZConfiguration** with parameters:
 - ProfileToken := profileToConfigure.token
 - 3.6.7.2. DUT responds with **RemovePTZConfigurationResponse**
- 3.6.8. If profileToConfigure contains MetadataConfiguration
 - 3.6.8.1. ONVIF Client invokes **RemoveMetadataConfiguration** with parameters:
 - ProfileToken := profileToConfigure.token



- 3.6.8.2. DUT responds with **RemoveMetadataConfigurationResponse**
- 3.6.9. If *profileToConfigure* contains AudioOutputConfiguration
 - 3.6.9.1. ONVIF Client invokes **RemoveAudioOutputConfiguration** with parameters:
 - ProfileToken := profileToConfigure.token
 - 3.6.9.2. DUT responds with RemoveAudioOutputConfigurationResponse
- 3.6.10. If profileToConfigure contains AudioDecoderConfiguration
 - 3.6.10.1. ONVIF Client invokes **RemoveAudioDecoderConfiguration** with parameters:
 - ProfileToken := profileToConfigure.token
 - 3.6.10.2. DUT responds with RemoveAudioDecoderConfigurationResponse
- 4. If DUT returns any other SOAP 1.2 fault, FAIL the test and skip other steps.

PASS –

DUT passes all assertions.

FAIL -

DUT did not send CreateProfileResponse message.

DUT did not send **GetProfilesResponse** message.

DUT did not send DeleteProfileResponse message.

DUT did not send RemoveVideoSourceConfigurationResponse message.

- DUT did not send RemoveAudioSourceConfigurationResponse message.
- DUT did not send RemoveVideoEncoderConfigurationResponse message.
- DUT did not send RemoveAudioEncoderConfigurationResponse message.

DUT did not send RemoveVideoAnalyticsConfigurationResponse message.

DUT did not send RemovePTZConfigurationResponse message.

DUT did not send **RemoveMetadataConfigurationResponse** message.

A.7 Restore media profile

Name: HelperRestoreMediaProfile

Procedure Purpose: Helper procedure to restore media profile.

Pre-requisite: Media Service is received from the DUT.

Input: Media Profile (profileToRestore).

Returns: None.

<TODO> ONVIF

Procedure:

- 1. If profileToRestore was deleted during Annex A.6:
 - 1.1. ONVIF client invokes CreateProfile with parameters

Name := profileToRestore.Name

Token := profileToRestore

1.2. The DUT responds with CreateProfileResponse message with parameters

Profile

2. ONVIF clients restore all profileToRestore configurations.

PASS -

The DUT passed all assertions.

FAIL -

DUT did not send CreateProfileResponse message.

DUT did not send GetProfilesResponse message.

DUT did not send DeleteProfileResponse message.

DUT did not send RemoveVideoSourceConfigurationResponse message.

DUT did not send RemoveAudioSourceConfigurationResponse message.

DUT did not send RemoveVideoEncoderConfigurationResponse message.

DUT did not send RemoveAudioEncoderConfigurationResponse message.

DUT did not send RemoveVideoAnalyticsConfigurationResponse message.

DUT did not send RemovePTZConfigurationResponse message.

DUT did not send RemoveMetadataConfigurationResponse message.

A.8 **Delete media profile**

Name: HelperDeleteMediaProfile

Procedure Purpose: Helper procedure to delete media profile.

Pre-requisite: Media Service is received from the DUT.

Input: Media Profile token (profileToDelete).

Returns: None.

ONVIF Client DeleteProfile (ProfileToken =:



Delete media profile

DeleteProfileResponse

DUT

-268-

profileToDelete)

Receive and validate response

Send Delete profile request

Procedure:

- 1. ONVIF Client invokes DeleteProfile with parameters:
 - ProfileToken =: profileToDelete
- 2. DUT responds with **DeleteProfileResponse.**

PASS –

The DUT passed all assertions.

FAIL -

DUT did not send **DeleteProfileResponse** message.

ONVIF