ONVIF Audio Backchannel Client Test Specification

Version 16.01 January, 2016

Table of Contents

Introduction

The goal of the ONVIF Test Specification set is to make it possible to realize fully interoperable IP physical security implementations from different vendors. This specification also acts as an input document to the development of a test tool which will be used to test the ONVIF Client implementation conformance towards ONVIF standard. This Client Test Tool analyzes network communications between ONVIF Devices and Clients being tested and determines whether a specific Client is ONVIF conformant (see ONVIF Conformance Process Specification [http://www.onvif.org/Documents/Specifications.aspx]).

This particular document defines test cases required for testing Audio Backchannel features of a Client application e.g. Audio Backchannel Streaming, Get Audio Decoder Configurations List, Get Audio Output Configurations List, Get Audio Outputs List, Get Audio Decoder Configuration, Get Audio Output Configuration, Profile Configuration for Audio Backchannel, Configure Audio Decoder Configuration, Configure Audio Output Configuration. It also describes the test framework, test setup, prerequisites, test policies needed for the execution of the described test cases.

Scope

This ONVIF Audio Backchannel Client Test Specification defines and regulates the conformance testing procedure for the ONVIF conformant Clients in the scope of Audio Backchannel features. Conformance testing is meant to be black-box network traces analysis and verification. The objective of this specification is to provide the test cases to test individual requirements of ONVIF Clients in the scope of Audio Backchannel features according to ONVIF Media Service Specification and ONVIF Streaming Specification.

The principal intended purposes are:

• Provide self-assessment tool for implementations.

• Provide comprehensive test suite coverage for Audio Backchannel features.

This specification **does not** address the following:

• 3rd parties Client use cases

• Non-functional (performance and regression) testing and analysis.

• SOAP Implementation Interoperability test i.e. Web Services Interoperability Basic Profile version 2.0 (WS-I BP2.0).

• Network protocol implementation Conformance test for HTTPS and HTTP protocols.

The following sections cover test cases needed for the verification of relevant features as mentioned in the ONVIF Profile Specifications.

Audio Backchannel Streaming

Audio Backchannel Streaming section specifies Client ability to stream audio for backchannel to Device.

Get Audio Decoder Configurations List

Get Audio Decoder Configurations List section specifies Client ability to request audio decoder configurations list from a Device.

Get Audio Output Configurations List

Get Audio Output Configurations List section specifies Client ability to request audio output configurations list from a Device.

Get Audio Outputs List

Get Audio Outputs List section specifies Client ability to request audio outputs list from a Device.

Get Audio Decoder Configuration

Get Audio Decoder Configuration section specifies Client ability to request audio decoder settings from a Device.

Get Audio Output Configuration

Get Audio Output Configuration section specifies Client ability to request audio output settings from a Device.

Profile Configuration for Audio Backchannel

Profile Configuration for Audio Backchannel section specifies Client ability to configure media profile for audio backchannel streaming on a Device.

Configure Audio Decoder Configuration

Configure Audio Decoder Configuration section specifies Client ability to change audio decoder configuration on a Device.

Configure Audio Output Configuration

Configure Audio Output Configuration section specifies Client ability to change audio output configuration on a Device.

Normative references

ONVIF Conformance Process Specification:

http://www.onvif.org/Documents/Specifications.aspx

ONVIF Profile Policy:

http://www.onvif.org/Documents/Specifications.aspx

ONVIF Core Specifications:

http://www.onvif.org/Documents/Specifications.aspx

ONVIF Core Client Test Specification:

http://www.onvif.org/Documents/Specifications.aspx

ONVIF Media Service Specification:

http://www.onvif.org/Documents/Specifications.aspx

ONVIF Streaming Specification:

http://www.onvif.org/Documents/Specifications.aspx

ISO/IEC Directives, Part 2:

http://www.iso.org/directives

ISO 16484-5:2014-09 Annex P:

https://www.iso.org/obp/ui/#!iso:std:63753:en

WS-BaseNotification:

http://docs.oasis-open.org/wsn/wsn-ws_base_notification-1.3-spec-os.pdf

W3C SOAP 1.2, Part 1, Messaging Framework:

http://www.w3.org/TR/soap12-part1/

W3C XML Schema Part 1: Structures Second Edition:

http://www.w3.org/TR/xmlschema-1/

W3C XML Schema Part 2: Datatypes Second Edition:

"http://www.w3.org/TR/xmlschema-2/ [http://www.w3.org/TR/xmlschema-2/]

W3C Web Services Addressing 1.0 – Core:

http://www.w3.org/TR/ws-addr-core/

IETF RFC 2326, Real Time Streaming Protocol (RTSP):

http://www.ietf.org/rfc/rfc2326.txt

Rules for the structure and drafting of International Standards, Annex H: Verbal forms for the expression of provisions.

Terms and Definitions

Conventions

The key words "shall", "shall not", "should", "should not", "may", "need not", "can", "cannot" in this specification are to be interpreted as described in [ISO/IEC Directives Part 2].

Definitions

This section describes terms and definitions used in this document.

Profile   See ONVIF Profile Policy.

ONVIF Device   Computer appliance or software program that exposes one or multiple ONVIF Web Services.

ONVIF Client	Computer appliance or software program that uses ONVIF Web Services.
Conversation	A Conversation is all exchanges between two MAC addresses that contains SOAP request and response.
Network	A network is an interconnected group of devices communicating using the Internet protocol.
Network Trace Capture file	Data file created by a network protocol analyzer software (such as Wireshark). Contains network packets data recorded during a live network communications.
SOAP	SOAP is a lightweight protocol intended for exchanging structured information in a decentralized, distributed environment. It uses XML technologies to define an extensible messaging framework providing a message construct that can be exchanged over a variety of underlying protocols.
Client Test Tool	ONVIF Client Test Tool that tests ONVIF Client implementation towards the ONVIF Test Specification set.
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	Maps a video or an audio source or an audio output to a video or an audio encoder, an audio decoder configuration and PTZ and analytics configuration
Valid Device Response	Device has responded to specific request with code HTTP or RTSP 200 OK and SOAP fault message has not appeared.

Abbreviations

This section describes abbreviations used in this document.

HTTP   Hyper Text Transport Protocol.

HTTPS Hyper Text Transport Protocol over Secure Socket Layer.

URI   Uniform Resource Identifier.

WSDL   Web Services Description Language.

XML   eXtensible Markup Language.

RTSP   Real Time Streaming Protocol.

RTP   Realtime Transport Protocol.

TCP   Transmission Control Protocol.

UDP   User Datagram Protocol.

SDP   Session Description Protocol.

AAC   Advanced Audio Coding.

Namespaces

Prefix and namespaces used in this test specification are listed in Table 1. **These prefixes are not part of the standard and an implementation can use any prefix.**

Table 1. Defined namespaces in this specification

Prefix	Namespace URI	Description
soapenv	http://www.w3.org/2003/05/soap-envelope	Envelope namespace as defined by SOAP 1.2 [SOAP 1.2, Part 1]
xs	http://www.w3.org/2001/XMLSchema	Instance namespace as defined by XS [XML-Schema, Part1] and [XMLSchema,Part 2]
xsi	http://www.w3.org/2001/XMLSchema-instance	XML schema instance namespace
tns1	http://www.onvif.org/ver10/topics	The namespace for the ONVIF topic namespace
tt	http://www.onvif.org/ver10/schema	ONVIF XML schema descriptions
trt	http://www.onvif.org/ver10/media/wsdl	The namespace for the WSDL media service
tev	http://www.onvif.org/ver10/events/wsdl	The namespace for the WSDL event service
wsnt	http://docs.oasis-open.org/wsn/b-2	Schema namespace of the [WS-BaseNotification] specification.

Test Overview

This section provides information for the test setup procedure and required prerequisites that should be followed during test case execution.

An ONVIF client with audio backchannel features support can provide audio backchannel configuration and streaming.

An ONVIF Profile is described by a fixed set of functionalities through a number of services that are provided by the ONVIF standard. A number of services and functionalities are mandatory for each type of ONVIF Profile. An ONVIF Device and ONVIF Client may support any combination of Profiles and other optional services and functionalities.

General

Test Cases are grouped depending on features. Each Test Cases group provides description of feature requirement level for Profiles, expected scenario under test and related test cases:

• Feature Level Requirement

• Expected Scenarios Under Test

• List of Test Cases

Feature Level Requirement

Feature Level Requirement item contains a feature ID and feature requirement level for the Profiles, which will be used for Profiles conformance.

If Feature Level Requirement is defined as Mandatory for some Profile, Client shall pass Expected Scenario Under Test for each Device with this Profile support to claim this Profile Conformance.

If Feature Level Requirement is defined as Conditional, Optional for some Profile, Client shall pass Expected Scenario Under Test for at least one Device with this Profile support to claim feature as supported.

Expected Scenarios Under Test

Expected Scenarios Under Test item contains expected scenario under test, conditions when the feature will be defined as supported and as not supported.

Test Cases

Test Case items contain list of test cases which are related to feature. Test cases provide exact procedure of testing feature support conditions.

Each Test Case contains the following parts:

- Test Label - Unique label for each test

- Test Case ID - Unique ID for each test

- Profile Normative References - Requirement level for the feature under test is defined in Profile Specification. This reference is informative and will not be used in conformance procedure.

- Feature Under Test - Feature which is under current test. Typically a particular command or an event.

- Test Purpose - The purpose of current test case.

- Pre-Requisite - The pre-requisite defines when the test should be performed. In case if pre-requisite does not match, the test result will be NOT DETECTED.

- Test Procedure - scenario expected to be reflected in network trace file.

- Test Result - Passed and failed criteria of the test case. Depending on these criteria test result will be defined as PASSED or FAILED.

- Validated Feature List - list of features ID related to this test case.

Test Setup

Collect Network Traces files required by the test cases.

Collect Feature List XML files for Devices detected in the Network Trace files.

Client shall support all mandatory and conditional features listed in the Device Feature List XML file supplied for the Profiles supported by the Client.

For ONVIF compatibility, the ONVIF Client shall follow the requirements of the conformance process. For details, please, see the latest ONVIF Conformance Process Specification.

Prerequisites

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

The Device shall be configured with an IPv4 address.

The Device shall be able to be discovered by the Client.

Audio Backchannel Streaming Test Cases

Feature Level Requirement:

Validated Feature: audio_backchannel_streaming

Profile A Requirement: None

Profile C Requirement: None

Profile G Requirement: None

Profile Q Requirement: None

Profile S Requirement: None

Expected Scenarios Under Test:

1. Client connects to Device to stream audio for backchannel.

2. Client is considered as supporting Audio Backchannel Streaming if the following conditions are met:

   • Client is able to get audio decoder configuration options to check supported audio backchannel streaming parameters using **GetAudioOutputConfigurationOptions** operation AND

   • Client is able to stream audio for backchannel using **AAC** OR **G.711** OR **G.726**.

3. Client is considered as NOT supporting Audio Backchannel Streaming if ANY of the following is TRUE:

   • No valid responses for **GetAudioOutputConfigurationOptions** request

   • No Audio Backchannel Streaming attempts were found OR

   • Detected AAC Audio Backchannel Streaming attempts have failed OR

   • Detected G.711 Audio Backchannel Streaming attempts have failed OR

   • Detected G.726 Audio Backchannel Streaming attempts have failed.

GET AUDIO DECODER CONFIGURATION OPTIONS

Test Label: Audio Backchannel Streaming - Get Audio Decoder Configuration Options

Test Case ID: AUDIOBACKCHANNELSTREAMING-1

Profile A Reference: None

Profile C Reference: None

Profile G Reference: None

Profile Q Reference: None

Profile S Reference: None

Feature Under Test: Get Audio Decoder Configuration Options

Test Purpose: To verify that Client is able to get audio decoder configuration options provided by Device using the **GetAudioDecoderConfigurationOptions** operation.

Pre-Requisite:

• The Network Trace Capture files contains at least one Conversation between Client and Device with **GetAudioDecoderConfigurationOptions** operation present.

• Device supports Audio Outputs.

Test Procedure (expected to be reflected in network trace file):

1. Client invokes **GetAudioDecoderConfigurationOptions** request message to retrieve audio decoder configuration options for the Device.

2. Device responses with code HTTP 200 OK and **GetAudioDecoderConfigurationOptionsResponse** message.

Test Result:

PASS -

• Client **GetAudioDecoderConfigurationOptions** request messages are valid according to XML Schemas listed in Namespaces AND

• Client **GetAudioDecoderConfigurationOptions** request in Test Procedure fulfills the following requirements:

  • [S1] **soapenv:Body** element has child element **trt:GetAudioDecoderConfigurationOptions** AND

• Device response to the **GetAudioDecoderConfigurationOptions** request fulfills the following requirements:

  • [S2] It has HTTP 200 response code AND

  • [S3] **soapenv:Body** element has child element **trt:GetAudioDecoderConfigurationOptionsResponse**.

FAIL -

• The Client failed PASS criteria.

Validated Feature List: audio_backchannel_streaming.get_audio_decoder_configuration_options

G.711 AUDIO BACKCHANNEL STREAMING

Test Label: Audio Backchannel Streaming - G.711

Test Case ID: AUDIOBACKCHANNELSTREAMING-2

Profile A Reference: None

Profile C Reference: None

Profile G Reference: None

Profile Q Reference: None

Profile S Reference: None

Feature Under Test: Audio Backchannel Streaming

Test Purpose: To verify that audio backchannel streaming to Device was successfully started by Client.

Pre-Requisite:

• The Network Trace Capture files contains at least one Conversation between Client and Device with audio backchannel streaming with G.711 encoding.

• Device supports G.711 encoding for Audio Outputs.

Test Procedure (expected to be reflected in network trace file):

1. Client invokes **GetStreamUri** request message for media profile that contains Audio Output Configuration and Audio Decoder Configuration with RTP-Unicast/UDP OR RTP-Multicast/UDP OR RTP/RTSP/TCP OR RTP-Unicast/RTSP/HTTP/TCP transport.

2. Device responses with code HTTP 200 OK and **GetStreamUriResponse** message.

3. Client invokes **RTSP DESCRIBE** request to retrieve media stream description with **Require** tag in RTSP header that contains "www.onvif.org/ver20/backchannel".

4. Device responses with code RTSP 200 OK with SDP that contains media type "audio" with session attribute "sendonly".

5. Client invokes **RTSP SETUP** request with transport parameter element to set media session parameters for audio backchannel with **Require** tag in RTSP header that contains "www.onvif.org/ver20/backchannel".

6. Device responses with code RTSP 200 OK.

7. Client invokes **RTSP PLAY** request to start media stream with **Require** tag in RTSP header that contains "www.onvif.org/ver20/backchannel".

8. Device responses with code RTSP 200 OK.

9. Client invokes **RTSP TEARDOWN** request to terminate the RTSP session with **Require** tag in RTSP header that contains "www.onvif.org/ver20/backchannel".

10.Device responses with code RTSP 200 OK.

Note: RTSP requests and RTSP response could be tunneled in HTTP if RTP-Unicast/RTSP/HTTP/TCP transport is used.

Test Result:

PASS -

• Client **RTSP DESCRIBE** request in Test Procedure fulfills the following requirements:

  • [S1] **Require** tag in RTSP header contains "www.onvif.org/ver20/backchannel" AND

- Device response to the **RTSP DESCRIBE** request fulfills the following requirements:

  - [S3] It has RTSP 200 response code AND

  - [S4] SDP packet contains media type "audio" (m=audio) with session attribute "sendonly" (a=sendonly) and sessions attribute "rtpmap" with encoding name "PCMU" AND

- There is a Device response to the **GetStreamUri** request in Test Procedure that fulfills the following requirements:

  - [S5] It has HTTP 200 response code AND

  - [S6] It is received from the same Device as the response for **RTSP DESCRIBE** request AND

  - [S7] It is received before the Client **RTSP DESCRIBE** request AND

  - [S8] It contains **trt:MediaUri\tt:Uri** element which value is equal to RTSP address that was used to send the **RTSP DESCRIBE** request AND

- There is Client **RTSP SETUP** request in Test Procedure that fulfills the following requirements:

  - [S9] It is invoked for the same Device as the response for **RTSP DESCRIBE** request AND

  - [S10] It is invoked after the Client **RTSP DESCRIBE** request AND

  - [S11] RTSP address that was used to send **RTSP SETUP** is corresponds to media type "audio" with session attribute "sendonly" depending on media session attribute, general session attribute and address that was used for the **RTSP DESCRIBE** request (see [RFC 2326]) AND

  - [S12] **Require** tag in RTSP header contains "www.onvif.org/ver20/backchannel" AND

- Device response to the **RTSP SETUP** request fulfills the following requirements:

  - [S13] It has RTSP 200 response code AND

- There is Client **RTSP PLAY** request in Test Procedure that fulfills the following requirements:

  - [S14] It is invoked for the same Device as the response for **RTSP SETUP** request AND

  - [S15] It is invoked after the Client **RTSP SETUP** request AND

  - [S16] RTSP address that was used to send it should be equal to address that was used for the **RTSP DESCRIBE** request AND

  - [S17] **Require** tag in RTSP header contains "www.onvif.org/ver20/backchannel" AND

- Device response to the **RTSP PLAY** request fulfills the following requirements:

  - [S18] It has RTSP 200 response code AND

- There is Client **RTSP TEARDOWN** request in Test Procedure that fulfills the following requirements:

  - [S19] It is invoked for the same Device as the response for **RTSP SETUP** request AND

  - [S20] It is invoked after the Client **RTSP PLAY** request AND

  - [S21] RTSP address that was used to send it should be equal to address that was used for the **RTSP DESCRIBE** request AND

- [S22] **Require** tag in RTSP header contains "www.onvif.org/ver20/backchannel" AND

• Device response to the **RTSP TEARDOWN** request fulfills the following requirements:

- [S23] It has RTSP 200 response code.

FAIL -

• The Client failed PASS criteria.

Validated Feature List: audio_backchannel_streaming.g711

G.726 AUDIO BACKCHANNEL STREAMING

Test Label: Audio Backchannel Streaming - G.726

Test Case ID: AUDIOBACKCHANNELSTREAMING-3

Profile A Reference: None

Profile C Reference: None

Profile G Reference: None

Profile Q Reference: None

Profile S Reference: None

Feature Under Test: Audio Backchannel Streaming

Test Purpose: To verify that audio backchannel streaming to Device was successfully started by Client.

Pre-Requisite:

• The Network Trace Capture files contains at least one Conversation between Client and Device with audio backchannel streaming with G.726 encoding.

• Device supports G.726 encoding for Audio Outputs.

Test Procedure (expected to be reflected in network trace file):

1. Client invokes **GetStreamUri** request message for media profile that contains Audio Output Configuration and Audio Decoder Configuration with RTP-Unicast/UDP OR RTP-Multicast/UDP OR RTP/RTSP/TCP OR RTP-Unicast/RTSP/HTTP/TCP transport.

2. Device responses with code HTTP 200 OK and **GetStreamUriResponse** message.

3. Client invokes **RTSP DESCRIBE** request to retrieve media stream description with **Require** tag in RTSP header that contains "www.onvif.org/ver20/backchannel".

4. Device responses with code RTSP 200 OK with SDP that contains media type "audio" with session attribute "sendonly".

5. Client invokes **RTSP SETUP** request with transport parameter element to set media session parameters for audio backchannel with **Require** tag in RTSP header that contains "www.onvif.org/ver20/backchannel".

6. Device responses with code RTSP 200 OK.

7. Client invokes **RTSP PLAY** request to start media stream with **Require** tag in RTSP header that contains "www.onvif.org/ver20/backchannel".

8. Device responses with code RTSP 200 OK.

9. Client invokes **RTSP TEARDOWN** request to terminate the RTSP session with **Require** tag in RTSP header that contains "www.onvif.org/ver20/backchannel".

10. Device responses with code RTSP 200 OK.

Note: RTSP requests and RTSP response could be tunneled in HTTP if RTP-Unicast/RTSP/HTTP/TCP transport is used.

Test Result:

PASS -

- Client **RTSP DESCRIBE** request in Test Procedure fulfills the following requirements:

  - [S1] **Require** tag in RTSP header contains "www.onvif.org/ver20/backchannel" AND

- Device response to the **RTSP DESCRIBE** request fulfills the following requirements:

  - [S3] It has RTSP 200 response code AND

  - [S4] SDP packet contains media type "audio" (m=audio) with session attribute "sendonly" (a=sendonly) and sessions attribute "rtpmap" with encoding name "G726-*" AND

- There is a Device response to the **GetStreamUri** request in Test Procedure that fulfills the following requirements:

  - [S5] It has HTTP 200 response code AND

  - [S6] It is received from the same Device the response for **RTSP DESCRIBE** request AND

  - [S7] It is received before the Client **RTSP DESCRIBE** request AND

  - [S8] It contains **trt:MediaUri\tt:Uri** element which value is equal to RTSP address that was used to send the **RTSP DESCRIBE** request AND

- There is Client **RTSP SETUP** request in Test Procedure that fulfills the following requirements:

  - [S9] It is invoked for the same Device as the response for **RTSP DESCRIBE** request AND

  - [S10] It is invoked after the Client **RTSP DESCRIBE** request AND

  - [S11] RTSP address that was used to send **RTSP SETUP** is corresponds to media type "audio" with session attribute "sendonly" depending on media session attribute, general session attribute and address that was used for the **RTSP DESCRIBE** request (see [RFC 2326]) AND

  - [S12] **Require** tag in RTSP header contains "www.onvif.org/ver20/backchannel" AND

- Device response to the **RTSP SETUP** request fulfills the following requirements:

  - [S13] It has RTSP 200 response code AND

- There is Client **RTSP PLAY** request in Test Procedure that fulfills the following requirements:

    - [S14] It is invoked for the same Device as the response for **RTSP SETUP** request AND

    - [S15] It is invoked after the Client **RTSP SETUP** request AND

    - [S16] RTSP address that was used to send it should be equal to address that was used for the **RTSP DESCRIBE** request AND

    - [S17] **Require** tag in RTSP header contains "www.onvif.org/ver20/backchannel" AND

- Device response to the **RTSP PLAY** request fulfills the following requirements:

    - [S18] It has RTSP 200 response code AND

- There is Client **RTSP TEARDOWN** request in Test Procedure that fulfills the following requirements:

    - [S19] It is invoked for the same Device as the response for **RTSP SETUP** request AND

    - [S20] It is invoked after the Client **RTSP PLAY** request AND

    - [S21] RTSP address that was used to send it should be equal to address that was used for the **RTSP DESCRIBE** request AND

    - [S22] **Require** tag in RTSP header contains "www.onvif.org/ver20/backchannel" AND

- Device response to the **RTSP TEARDOWN** request fulfills the following requirements:

    - [S23] It has RTSP 200 response code.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: audio_backchannel_streaming.g726

AAC AUDIO BACKCHANNEL STREAMING

Test Label: Audio Backchannel Streaming - AAC

Test Case ID: AUDIOBACKCHANNELSTREAMING-4

Profile A Reference: None

Profile C Reference: None

Profile G Reference: None

Profile Q Reference: None

Profile S Reference: None

Feature Under Test: Audio Backchannel Streaming

Test Purpose: To verify that audio backchannel streaming to Device was successfully started by Client.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with audio backchannel streaming with AAC encoding.

- Device supports AAC encoding for Audio Outputs.

Test Procedure (expected to be reflected in network trace file):

1. Client invokes **GetStreamUri** request message for media profile that contains Audio Output Configuration and Audio Decoder Configuration with RTP-Unicast/UDP OR RTP-Multicast/UDP OR RTP/RTSP/TCP OR RTP-Unicast/RTSP/HTTP/TCP transport.

2. Device responses with code HTTP 200 OK and **GetStreamUriResponse** message.

3. Client invokes **RTSP DESCRIBE** request to retrieve media stream description with **Require** tag in RTSP header that contains "www.onvif.org/ver20/backchannel".

4. Device responses with code RTSP 200 OK with SDP that contains media type "audio" with session attribute "sendonly".

5. Client invokes **RTSP SETUP** request with transport parameter element to set media session parameters for audio backchannel with **Require** tag in RTSP header that contains "www.onvif.org/ver20/backchannel".

6. Device responses with code RTSP 200 OK.

7. Client invokes **RTSP PLAY** request to start media stream with **Require** tag in RTSP header that contains "www.onvif.org/ver20/backchannel".

8. Device responses with code RTSP 200 OK.

9. Client invokes **RTSP TEARDOWN** request to terminate the RTSP session with **Require** tag in RTSP header that contains "www.onvif.org/ver20/backchannel".

10.Device responses with code RTSP 200 OK.

Note: RTSP requests and RTSP response could be tunneled in HTTP if RTP-Unicast/RTSP/HTTP/TCP transport is used.

Test Result:

PASS -

- Client **RTSP DESCRIBE** request in Test Procedure fulfills the following requirements:

  - [S1] **Require** tag in RTSP header contains "www.onvif.org/ver20/backchannel" AND

- Device response to the **RTSP DESCRIBE** request fulfills the following requirements:

  - [S3] It has RTSP 200 response code AND

  - [S4] SDP packet contains media type "audio" (m=audio) with session attribute "sendonly" (a=sendonly) and sessions attribute "rtpmap" with encoding name "mpeg4-generic" AND

- There is a Device response to the **GetStreamUri** request in Test Procedure that fulfills the following requirements:

  - [S5] It has HTTP 200 response code AND

- [S6] It is received from the same Device the response for **RTSP DESCRIBE** request AND

- [S7] It is received before the Client **RTSP DESCRIBE** request AND

- [S8] It contains **trt:MediaUri\tt:Uri** element which value is equal to RTSP address that was used to send the **RTSP DESCRIBE** request AND

- There is Client **RTSP SETUP** request in Test Procedure that fulfills the following requirements:

  - [S9] It is invoked for the same Device as the response for **RTSP DESCRIBE** request AND

  - [S10] It is invoked after the Client **RTSP DESCRIBE** request AND

  - [S11] RTSP address that was used to send **RTSP SETUP** is corresponds to media type "audio" with session attribute "sendonly" depending on media session attribute, general session attribute and address that was used for the **RTSP DESCRIBE** request (see [RFC 2326]) AND

  - [S12] **Require** tag in RTSP header contains "www.onvif.org/ver20/backchannel" AND

- Device response to the **RTSP SETUP** request fulfills the following requirements:

  - [S13] It has RTSP 200 response code AND

- There is Client **RTSP PLAY** request in Test Procedure that fulfills the following requirements:

  - [S14] It is invoked for the same Device as the response for **RTSP SETUP** request AND

  - [S15] It is invoked after the Client **RTSP SETUP** request AND

  - [S16] RTSP address that was used to send it should be equal to address that was used for the **RTSP DESCRIBE** request AND

  - [S17] **Require** tag in RTSP header contains "www.onvif.org/ver20/backchannel" AND

- Device response to the **RTSP PLAY** request fulfills the following requirements:

  - [S18] It has RTSP 200 response code AND

- There is Client **RTSP TEARDOWN** request in Test Procedure that fulfills the following requirements:

  - [S19] It is invoked for the same Device the response for **RTSP SETUP** request AND

  - [S20] It is invoked after the Client **RTSP PLAY** request AND

  - [S21] RTSP address that was used to send it should be equal to address that was used for the **RTSP DESCRIBE** request AND

  - [S22] **Require** tag in RTSP header contains "www.onvif.org/ver20/backchannel" AND

- Device response to the **RTSP TEARDOWN** request fulfills the following requirements:

  - [S23] It has RTSP 200 response code.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: audio_backchannel_streaming.aac

Get Audio Decoder Configurations List Test Cases

Feature Level Requirement:

Validated Feature: get_audio_decoder_configurations_list

Profile A Requirement: None

Profile C Requirement: None

Profile G Requirement: None

Profile Q Requirement: None

Profile S Requirement: None

Expected Scenarios Under Test:

1. Client connects to Device to retrieve a complete list of Audio Decoders.

2. Client is considered as supporting Get Audio Decoder Configurations List if the following conditions are met:

   • Client is able to list available Get Audio Decoder Configurations List using **GetAudioDecoderConfigurations** operation.

3. Client is considered as NOT supporting Get Audio Decoder Configurations List if ANY of the following is TRUE:

   • No valid responses for **GetAudioDecoderConfigurations** request.

GET AUDIO DECODER CONFIGURATIONS

Test Label: Get Audio Decoder Configurations List - Get Audio Decoder Configurations

Test Case ID: GETAUDIODECODERCONFIGURATIONS-1

Feature Under Test: Get Audio Decoder Configurations

Test Purpose: To verify that list of all audio decoder configurations items provided by Device is received by Client using the **GetAudioDecoderConfigurations** operation.

Pre-Requisite:

• The Network Trace Capture files contains at least one Conversation between Client and Device with **GetAudioDecoderConfigurations** operation present.

• Device supports Audio Outputs.

Test Procedure (expected to be reflected in network trace file):

1. Client invokes **GetAudioDecoderConfigurations** request message to retrieve a list of all audio decoder configurations from the Device.

2. Device responses with code HTTP 200 OK and **GetAudioDecoderConfigurationsResponse** message.

Test Result:

PASS -

- Client **GetAudioDecoderConfigurations** request messages are valid according to XML Schemas listed in Namespaces AND

- Client **GetAudioDecoderConfigurations** request in Test Procedure fulfills the following requirements:

  - [S1] **soapenv:Body** element has child element **trt:GetAudioDecoderConfigurations** AND

- Device response to the **GetAudioDecoderConfigurations** request fulfills the following requirements:

  - [S2] It has HTTP 200 response code AND

  - [S3] **soapenv:Body** element has child element **trt:GetAudioDecoderConfigurationsResponse**.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: get_audio_decoder_configurations_list.get_audio_decoder_configurations

Get Audio Output Configurations List Test Cases

Feature Level Requirement:

Validated Feature: get_audio_output_configurations_list

Profile A Requirement: None

Profile C Requirement: None

Profile G Requirement: None

Profile Q Requirement: None

Profile S Requirement: None

Expected Scenarios Under Test:

1. Client connects to Device to retrieve a complete list of Audio Outputs.

2. Client is considered as supporting Get Audio Output Configurations List if the following conditions are met:

   - Client is able to list available Get Audio Output Configurations List using **GetAudioOutputConfigurations** operation.

3. Client is considered as NOT supporting Get Audio Output Configurations List if ANY of the following is TRUE:

   - No valid responses for **GetAudioOutputConfigurations** request.

GET AUDIO OUTPUT CONFIGURATIONS

Test Label: Get Audio Output Configurations List - Get Audio Output Configurations

Test Case ID: GETAUDIOOUTPUTCONFIGURATIONS-1

Feature Under Test: Get Audio Output Configurations

Test Purpose: To verify that list of all audio output configurations items provided by Device is received by Client using the **GetAudioOutputConfigurations** operation.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with **GetAudioOutputConfigurations** operation present.

- Device supports Audio Outputs.

Test Procedure (expected to be reflected in network trace file):

1. Client invokes **GetAudioOutputConfigurations** request message to retrieve a list of all audio output configurations from the Device.

2. Device responses with code HTTP 200 OK and **GetAudioOutputConfigurationsResponse** message.

Test Result:

PASS -

- Client **GetAudioOutputConfigurations** request messages are valid according to XML Schemas listed in Namespaces AND

- Client **GetAudioOutputConfigurations** request in Test Procedure fulfills the following requirements:

  - [S1] **soapenv:Body** element has child element **trt:GetAudioOutputConfigurations** AND

- Device response to the **GetAudioOutputConfigurations** request fulfills the following requirements:

  - [S2] It has HTTP 200 response code AND

  - [S3] **soapenv:Body** element has child element **trt:GetAudioOutputConfigurationsResponse**.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: get_audio_output_configurations_list.get_audio_output_configurations

Get Audio Outputs List Test Cases

Feature Level Requirement:

Validated Feature: get_audio_outputs_list

Profile A Requirement: None

Profile C Requirement: None

Profile G Requirement: None

Profile Q Requirement: None

Profile S Requirement: None

Expected Scenarios Under Test:

1. Client connects to Device to retrieve a complete list of Audio Outputs.

2. Client is considered as supporting Get Audio Outputs List if the following conditions are met:

   • Client is able to list available Get Audio Outputs List using **GetAudioOutputs** operation (Media Service or Device IO Service).

3. Client is considered as NOT supporting Get Audio Outputs List if ANY of the following is TRUE:

   • No valid responses for **GetAudioOutputs** request (Media Service or Device IO Service).

GET AUDIO OUTPUTS

Test Label: Get Audio Outputs List - Get Audio Outputs

Test Case ID: GETAUDIOOUTPUTS-1

Feature Under Test: Get Audio Outputs

Test Purpose: To verify that list of all audio outputs items provided by Device is received by Client using the **GetAudioOutputs** operation (Media Service or Device IO Service).

Pre-Requisite:

• The Network Trace Capture files contains at least one Conversation between Client and Device with **GetAudioOutputs** operation (Media Service or Device IO Service) present.

• Device supports Audio Outputs.

Test Procedure (expected to be reflected in network trace file):

1. Client invokes **GetAudioOutputs** request message (Media Service or Device IO Service) to retrieve a list of all audio outputs from the Device.

2. Device responses with code HTTP 200 OK and **GetAudioOutputsResponse** message.

Test Result:

PASS -

• Client **GetAudioOutputs** request messages are valid according to XML Schemas listed in Namespaces AND

• Client **GetAudioOutputs** request in Test Procedure fulfills the following requirements:

   • [S1] **soapenv:Body** element has child element **trt:GetAudioOutputs** AND

• Device response to the **GetAudioOutputs** request fulfills the following requirements:

   • [S2] It has HTTP 200 response code AND

- [S3] **soapenv:Body** element has child element **trt:GetAudioOutputsResponse**.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: get_audio_outputs_list.get_audio_outputs

Get Audio Decoder Configuration Test Cases

Feature Level Requirement:

Validated Feature: get_audio_decoder_configuration

Profile A Requirement: None

Profile C Requirement: None

Profile G Requirement: None

Profile Q Requirement: None

Profile S Requirement: None

Expected Scenarios Under Test:

1. Client connects to Device to retrieve an Audio Decoder Configuration.

2. Client is considered as supporting Get Audio Decoder Configuration if the following conditions are met:

   - Client is able to get Audio Decoder Configuration using **GetAudioDecoderConfiguration** operation OR Client supports get_audio_decoder_configurations_list.get_audio_decoder_configurations feature.

3. Client is considered as NOT supporting Get Audio Decoder Configuration if ANY of the following is TRUE:

   - No valid responses for **GetAudioDecoderConfiguration** request.

GET AUDIO DECODER CONFIGURATION

Test Label: Get Audio Decoder Configuration - Get Audio Decoder Configuration

Test Case ID: GETAUDIODECODERCONFIGURATION-1

Profile A Reference: None

Profile C Reference: None

Profile G Reference: None

Profile Q Reference: None

Profile S Reference: None

Feature Under Test: Get Audio Decoder Configuration

Test Purpose: To verify that audio decoder configuration provided by Device is received by Client using the **GetAudioDecoderConfiguration** operation.

Pre-Requisite:

• The Network Trace Capture files contains at least one Conversation between Client and Device with **GetAudioDecoderConfiguration** operation present.

• Device supports Audio Outputs.

Test Procedure (expected to be reflected in network trace file):

1. Client invokes **GetAudioDecoderConfiguration** request message to retrieve audio decoder configuration for specified audio decoder configuration from the Device.

2. Device responses with code HTTP 200 OK and **GetAudioDecoderConfigurationResponse** message.

Test Result:

PASS -

• Client **GetAudioDecoderConfiguration** request messages are valid according to XML Schemas listed in Namespaces AND

• Client **GetAudioDecoderConfiguration** request in Test Procedure fulfills the following requirements:

  • [S1] **soapenv:Body** element has child element **trt:GetAudioDecoderConfiguration** AND

• Device response to the **GetAudioDecoderConfiguration** request fulfills the following requirements:

  • [S2] It has HTTP 200 response code AND

  • [S3] **soapenv:Body** element has child element **trt:GetAudioDecoderConfigurationResponse**.

FAIL -

• The Client failed PASS criteria.

Validated Feature List: get_audio_decoder_configuration.get_audio_decoder_configuration

Get Audio Output Configuration Test Cases

Feature Level Requirement:

Validated Feature: get_audio_output_configuration

Profile A Requirement: None

Profile C Requirement: None

Profile G Requirement: None

Profile Q Requirement: None

Profile S Requirement: None

Expected Scenarios Under Test:

1. Client connects to Device to retrieve an Audio Output Configuration.

2. Client is considered as supporting Get Audio Output Configuration if the following conditions are met:

   • Client is able to get Audio Output Configuration using **GetAudioOutputConfiguration** operation (Media Service OR Device IO Service) OR Client supports get_audio_output_configurations_list.get_audio_output_configurations feature.

3. Client is considered as NOT supporting Get Audio Output Configuration if ANY of the following is TRUE:

   • No valid responses for **GetAudioOutputConfiguration** request.

GET AUDIO OUTPUT CONFIGURATION

Test Label: Get Audio Output Configuration - Get Audio Output Configuration

Test Case ID: GETAUDIOOUTPUTCONFIGURATION-1

Profile A Reference: None

Profile C Reference: None

Profile G Reference: None

Profile Q Reference: None

Profile S Reference: None

Feature Under Test: Get Audio Output Configuration

Test Purpose: To verify that audio output configuration provided by Device is received by Client using the **GetAudioOutputConfiguration** operation.

Pre-Requisite:

• The Network Trace Capture files contains at least one Conversation between Client and Device with **GetAudioOutputConfiguration** operation present.

• Device supports Audio Outputs.

Test Procedure (expected to be reflected in network trace file):

1. Client invokes **GetAudioOutputConfiguration** request message to retrieve audio output configuration for specified audio output configuration from the Device.

2. Device responses with code HTTP 200 OK and **GetAudioOutputConfigurationResponse** message.

Test Result:

PASS -

• Client **GetAudioOutputConfiguration** request messages are valid according to XML Schemas listed in Namespaces AND

• Client **GetAudioOutputConfiguration** request in Test Procedure fulfills the following requirements:

- [S1] **soapenv:Body** element has child element **trt:GetAudioOutputConfiguration** AND

- Device response to the **GetAudioOutputConfiguration** request fulfills the following requirements:

  - [S2] It has HTTP 200 response code AND

  - [S3] **soapenv:Body** element has child element **trt:GetAudioOutputConfigurationResponse**.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: get_audio_output_configuration.get_audio_output_configuration

Profile Configuration for Audio Backchannel Test Cases

Feature Level Requirement:

Validated Feature: profile_configuration_audio_backchannel

Profile A Requirement: None

Profile C Requirement: None

Profile G Requirement: None

Profile Q Requirement: None

Profile S Requirement: None

Expected Scenarios Under Test:

1. Client connects to Device to configure profile for Audio Backchannel streaming.

2. Client is considered as supporting Profile Configuration for Audio Backchannel details if the following conditions are met:

   - Client is able to get compatible Audio Output Configuration using **GetCompatibleAudioOutputConfigurations** operation for specified profile AND

   - Client is able to add or replace Audio Output Configuration in media profile using **AddAudioOutputConfiguration** operation for specified audio output configuration and compatible with specified profile AND

   - Client may be able to remove Audio Output Configuration from media profile using **RemoveAudioOutputConfiguration** operation for specified profile AND

   - Client is able to get compatible Audio Decoder Configuration using **GetCompatibleAudioDecoderConfigurations** operation for specified profile AND

   - Client is able to add or replace Audio Decoder Configuration in media profile using **AddAudioDecoderConfiguration** operation for specified audio decoder configuration and compatible with specified profile AND

- Client may be able to remove Audio Decoder Configuration from media profile using **RemoveAudioDecoderConfiguration** operation for specified profile.

3. Client is considered as NOT supporting Profile Configuration for Audio Backchannel if ANY of the following is TRUE:

   - No valid responses for **GetCompatibleAudioOutputConfigurations** request OR

   - No valid responses for **AddAudioOutputConfiguration** request OR

   - Client tries to invoke **AddAudioOutputConfiguration** request without **GetCompatibleAudioOutputConfigurations** request for specified profile OR

   - Detected RemoveAudioOutputConfiguration request attempt have failed OR

   - No valid responses for **GetCompatibleAudioDecoderConfigurations** request OR

   - No valid responses for **AddAudioDecoderConfiguration** request OR

   - Client tries to invoke **AddAudioDecoderConfiguration** request without **GetCompatibleAudioDecoderConfigurations** request for specified profile OR

   - Detected RemoveAudioDecoderConfiguration request attempt has failed.

GET COMPATIBLE AUDIO OUTPUT CONFIGURATIONS

Test Label: Profile Configuration for Audio Backchannel - Get Compatible Audio Output Configurations

Test Case ID: PROFILECONFIGURATIONAUDIOBACKCHANNEL-1

Profile A Reference: None

Profile C Reference: None

Profile G Reference: None

Profile Q Reference: None

Profile S Reference: None

Feature Under Test: Get Compatible Audio Output Configurations

Test Purpose: To verify that compatible audio output configurations provided by Device for specified media profile is received by Client using the **GetCompatibleAudioOutputConfigurations** operation.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with **GetCompatibleAudioOutputConfigurations** operation present.

- Device supports Audio Outputs.

Test Procedure (expected to be reflected in network trace file):

1. Client invokes **GetCompatibleAudioOutputConfigurations** request message to retrieve compatible audio output configurations for specified media profile from the Device.

2. Device responses with code HTTP 200 OK and **GetCompatibleAudioOutputConfigurationsResponse** message.

Test Result:

PASS -

• Client **GetCompatibleAudioOutputConfigurations** request messages are valid according to XML Schemas listed in Namespaces AND

• Client **GetCompatibleAudioOutputConfigurations** request in Test Procedure fulfills the following requirements:

   • [S1] **soapenv:Body** element has child element **trt:GetCompatibleAudioOutputConfigurations** AND

• Device response to the **GetCompatibleAudioOutputConfigurations** request fulfills the following requirements:

   • [S2] It has HTTP 200 response code AND

   • [S3] **soapenv:Body** element has child element **trt:GetCompatibleAudioOutputConfigurationsResponse**.

FAIL -

• The Client failed PASS criteria.

Validated Feature List: profile_configuration_audio_backchannel.get_compatible_audio_output_configurations

ADD AUDIO OUTPUT CONFIGURATION

Test Label: Profile Configuration for Audio Backchannel - Add Audio Output Configuration

Test Case ID: PROFILECONFIGURATIONAUDIOBACKCHANNEL-2

Profile A Reference: None

Profile C Reference: None

Profile G Reference: None

Profile Q Reference: None

Profile S Reference: None

Feature Under Test: Add Audio Output Configuration

Test Purpose: To verify that Client is able to add or replace audio output configurations on a Device for specified audio output configuration and compatible with specified profile using the **AddAudioOutputConfiguration** operation.

Pre-Requisite:

• The Network Trace Capture files contains at least one Conversation between Client and Device with **AddAudioOutputConfiguration** operation present.

• Device supports Audio Outputs.

Test Procedure (expected to be reflected in network trace file):

1. Client invokes **GetCompatibleAudioOutputConfigurations** request message to retrieve compatible audio output configurations for specified media profile from the Device.

2. Device responses with code HTTP 200 OK and **GetCompatibleAudioOutputConfigurationsResponse** message.

3. Client invokes **AddAudioOutputConfiguration** request message to add or replace audio output configurations for specified media profile and with audio output configuration token that was recieved in **GetCompatibleAudioOutputConfigurationsResponse** message from the Device for the same media profile.

4. Device responses with code HTTP 200 OK and **AddAudioOutputConfigurationResponse** message.

Test Result:

PASS -

• Client **AddAudioOutputConfiguration** request messages are valid according to XML Schemas listed in Namespaces AND

• Client **AddAudioOutputConfiguration** request in Test Procedure fulfills the following requirements:

  • [S1] **soapenv:Body** element has child element **trt:AddAudioOutputConfiguration** AND

• Device response to the **AddAudioOutputConfiguration** request fulfills the following requirements:

  • [S2] It has HTTP 200 response code AND

  • [S3] **soapenv:Body** element has child element **trt:AddAudioOutputConfigurationResponse** AND

• There is Client **GetCompatibleAudioOutputConfigurations** request in Test Procedure that fulfills the following requirements:

  • [S4] It is invoked for the same Device the response for **AddAudioOutputConfiguration** request AND

  • [S5] It is invoked before the Client **AddAudioOutputConfiguration** request AND

  • [S6] **trt:ProfileToken** element value is equal to **trt:ProfileToken** element from the **AddAudioOutputConfiguration** request AND

  • [S7] It is the last **GetCompatibleAudioOutputConfigurations** request which corresponds [S4], [S5] AND [S6] AND

• Device response to the **GetCompatibleAudioOutputConfigurations** request fulfills the following requirements:

  • [S8] It has HTTP 200 response code AND

  • [S9] **soapenv:Body** element has child element **trt:GetCompatibleAudioOutputConfigurationsResponse** AND

  • [S10] It contains **trt:Configurations/@token** attribute value equal to **trt:ConfigurationToken** from the **AddAudioOutputConfiguration** request messages.

FAIL -

• The Client failed PASS criteria.

Validated Feature List: profile_configuration_audio_backchannel.add_audio_output_configuration

REMOVE AUDIO OUTPUT CONFIGURATION

Test Label: Profile Configuration for Audio Backchannel - Remove Audio Output Configuration

Test Case ID: PROFILECONFIGURATIONAUDIOBACKCHANNEL-3

Profile A Reference: None

Profile C Reference: None

Profile G Reference: None

Profile Q Reference: None

Profile S Reference: None

Feature Under Test: Remove Audio Output Configuration

Test Purpose: To verify that Client is able to remove audio output configurations on a Device from specified profile using the **RemoveAudioOutputConfiguration** operation.

Pre-Requisite:

• The Network Trace Capture files contains at least one Conversation between Client and Device with **RemoveAudioOutputConfiguration** operation present.

• Device supports Audio Outputs.

Test Procedure (expected to be reflected in network trace file):

1. Client invokes **RemoveAudioOutputConfiguration** request message to remove audio output configurations from specified media profile on the Device.

2. Device responses with code HTTP 200 OK and **RemoveAudioOutputConfigurationResponse** message.

Test Result:

PASS -

• Client **RemoveAudioOutputConfiguration** request messages are valid according to XML Schemas listed in Namespaces AND

• Client **RemoveAudioOutputConfiguration** request in Test Procedure fulfills the following requirements:

  • [S1] **soapenv:Body** element has child element **trt:RemoveAudioOutputConfiguration** AND

• Device response to the **RemoveAudioOutputConfiguration** request fulfills the following requirements:

  • [S2] It has HTTP 200 response code AND

- [S3] **soapenv:Body** element has child element **trt:RemoveAudioOutputConfigurationResponse**.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: profile_configuration_audio_backchannel.remove_audio_output_configuration

GET COMPATIBLE AUDIO DECODER CONFIGURATIONS

Test Label: Profile Configuration for Audio Backchannel - Get Compatible Audio Decoder Configurations

Test Case ID: PROFILECONFIGURATIONAUDIOBACKCHANNEL-4

Profile A Reference: None

Profile C Reference: None

Profile G Reference: None

Profile Q Reference: None

Profile S Reference: None

Feature Under Test: Get Compatible Audio Decoder Configurations

Test Purpose: To verify that compatible audio decoder configurations provided by Device for specified media profile is received by Client using the **GetCompatibleAudioDecoderConfigurations** operation.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with **GetCompatibleAudioDecoderConfigurations** operation present.

- Device supports Audio Decoders.

Test Procedure (expected to be reflected in network trace file):

1. Client invokes **GetCompatibleAudioDecoderConfigurations** request message to retrieve compatible audio decoder configurations for specified media profile from the Device.

2. Device responses with code HTTP 200 OK and **GetCompatibleAudioDecoderConfigurationsResponse** message.

Test Result:

PASS -

- Client **GetCompatibleAudioDecoderConfigurations** request messages are valid according to XML Schemas listed in Namespaces AND

- Client **GetCompatibleAudioDecoderConfigurations** request in Test Procedure fulfills the following requirements:

- [S1] **soapenv:Body** element has child element **trt:GetCompatibleAudioDecoderConfigurations** AND

- Device response to the **GetCompatibleAudioDecoderConfigurations** request fulfills the following requirements:

  - [S2] It has HTTP 200 response code AND

  - [S3] **soapenv:Body** element has child element **trt:GetCompatibleAudioDecoderConfigurationsResponse**.

FAIL -

- The Client failed PASS criteria.

Validated   Feature   List: profile_configuration_audio_backchannel.get_compatible_audio_decoder_configurations

ADD AUDIO DECODER CONFIGURATION

Test Label: Profile Configuration for Audio Backchannel - Add Audio Decoder Configuration

Test Case ID: PROFILECONFIGURATIONAUDIOBACKCHANNEL-5

Profile A Reference: None

Profile C Reference: None

Profile G Reference: None

Profile Q Reference: None

Profile S Reference: None

Feature Under Test: Add Audio Decoder Configuration

Test Purpose: To verify that Client is able to add or replace audio decoder configurations on a Device for specified audio decoder configuration and compatible with specified profile using the **AddAudioDecoderConfiguration** operation.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with **AddAudioDecoderConfiguration** operation present.

- Device supports Audio Decoders.

Test Procedure (expected to be reflected in network trace file):

1. Client invokes **GetCompatibleAudioDecoderConfigurations** request message to retrieve compatible audio decoder configurations for specified media profile from the Device.

2. Device   responses   with   code   HTTP   200   OK   and **GetCompatibleAudioDecoderConfigurationsResponse** message.

3. Client invokes **AddAudioDecoderConfiguration** request message to add or replace audio decoder configurations for specified media profile and with audio decoder configuration token that was recieved

in **GetCompatibleAudioDecoderConfigurationsResponse** message from the Device for the same media profile.

4. Device responses with code HTTP 200 OK and **AddAudioDecoderConfigurationResponse** message.

Test Result:

PASS -

• Client **AddAudioDecoderConfiguration** request messages are valid according to XML Schemas listed in Namespaces AND

• Client **AddAudioDecoderConfiguration** request in Test Procedure fulfills the following requirements:

  • [S1] **soapenv:Body** element has child element **trt:AddAudioDecoderConfiguration** AND

• Device response to the **AddAudioDecoderConfiguration** request fulfills the following requirements:

  • [S2] It has HTTP 200 response code AND

  • [S3] **soapenv:Body** element has child element **trt:AddAudioDecoderConfigurationResponse** AND

• There is Client **GetCompatibleAudioDecoderConfigurations** request in Test Procedure that fulfills the following requirements:

  • [S4] It is invoked for the same Device as the response for **AddAudioDecoderConfiguration** request AND

  • [S5] It is invoked before the Client **AddAudioDecoderConfiguration** request AND

  • [S6] **trt:ProfileToken** element value is equal to **trt:ProfileToken** element from the **AddAudioDecoderConfiguration** request AND

  • [S7] It is the last **GetCompatibleAudioDecoderConfigurations** request which corresponds [S4], [S5] AND [S6] AND

• Device response to the **GetCompatibleAudioDecoderConfigurations** request fulfills the following requirements:

  • [S8] It has HTTP 200 response code AND

  • [S9] **soapenv:Body** element has child element **trt:GetCompatibleAudioDecoderConfigurationsResponse** AND

  • [S10] It contains **trt:Configurations/@token** attribute value equal to **trt:ConfigurationToken** from the **AddAudioDecoderConfiguration** request messages.

FAIL -

• The Client failed PASS criteria.

Validated Feature List: profile_configuration_audio_backchannel.add_audio_decoder_configuration

REMOVE AUDIO DECODER CONFIGURATION

Test Label: Profile Configuration for Audio Backchannel - Remove Audio Decoder Configuration

Test Case ID: PROFILECONFIGURATIONAUDIOBACKCHANNEL-6

Profile A Reference: None

Profile C Reference: None

Profile G Reference: None

Profile Q Reference: None

Profile S Reference: None

Feature Under Test: Remove Audio Decoder Configuration

Test Purpose: To verify that Client is able to remove audio decoder configurations on a Device from specified profile using the **RemoveAudioDecoderConfiguration** operation.

Pre-Requisite:

• The Network Trace Capture files contains at least one Conversation between Client and Device with **RemoveAudioDecoderConfiguration** operation present.

• Device supports Audio Decoders.

Test Procedure (expected to be reflected in network trace file):

1. Client invokes **RemoveAudioDecoderConfiguration** request message to remove audio decoder configurations from specified media profile on the Device.

2. Device responses with code HTTP 200 OK and **RemoveAudioDecoderConfigurationResponse** message.

Test Result:

PASS -

• Client **RemoveAudioDecoderConfiguration** request messages are valid according to XML Schemas listed in Namespaces AND

• Client **RemoveAudioDecoderConfiguration** request in Test Procedure fulfills the following requirements:

  • [S1] **soapenv:Body** element has child element **trt:RemoveAudioDecoderConfiguration** AND

• Device response to the **RemoveAudioDecoderConfiguration** request fulfills the following requirements:

  • [S2] It has HTTP 200 response code AND

  • [S3] **soapenv:Body** element has child element **trt:RemoveAudioDecoderConfigurationResponse**.

FAIL -

• The Client failed PASS criteria.

Validated   Feature   List:
profile_configuration_audio_backchannel.remove_audio_decoder_configuration

Configure Audio Decoder Configuration Test Cases

Feature Level Requirement:

Validated Feature: set_audio_decoder_configuration

Profile A Requirement: None

Profile C Requirement: None

Profile G Requirement: None

Profile Q Requirement: None

Profile S Requirement: None

Expected Scenarios Under Test:

1. Client connects to Device to change Audio Decoder Configuration settings.

2. Client is considered as supporting Configure Audio Decoder Configuration if the following conditions are met:

   • Client is able to change Audio Decoder Configuration settings using **SetAudioDecoderConfiguration** operation.

3. Client is considered as NOT supporting Configure Audio Decoder Configuration if ANY of the following is TRUE:

   • No valid responses for **SetAudioDecoderConfiguration** request.

SET AUDIO DECODER CONFIGURATION

Test Label: Configure Audio Decoder Configuration - Set Audio Decoder Configuration

Test Case ID: SETAUDIODECODERCONFIGURATION-1

Profile A Reference: None

Profile C Reference: None

Profile G Reference: None

Profile Q Reference: None

Profile S Reference: None

Feature Under Test: Set Audio Decoder Configuration

Test Purpose: To verify that Client is able to change audio decoder configuration provided by Device using the **SetAudioDecoderConfiguration** operation.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with **SetAudioDecoderConfiguration** operation present.

- Device supports Audio Outputs.

Test Procedure (expected to be reflected in network trace file):

1. Client invokes **SetAudioDecoderConfiguration** request message to change audio decoder configuration on the Device.

2. Device responses with code HTTP 200 OK and **SetAudioDecoderConfigurationResponse** message.

Test Result:

PASS -

- Client **SetAudioDecoderConfiguration** request messages are valid according to XML Schemas listed in Namespaces AND

- Client **SetAudioDecoderConfiguration** request in Test Procedure fulfills the following requirements:

  - [S1] **soapenv:Body** element has child element **trt:SetAudioDecoderConfiguration** AND

- Device response to the **SetAudioDecoderConfiguration** request fulfills the following requirements:

  - [S2] It has HTTP 200 response code AND

  - [S3] **soapenv:Body** element has child element **trt:SetAudioDecoderConfigurationResponse**.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: set_audio_decoder_configuration.set_audio_decoder_configuration

Configure Audio Output Configuration Test Cases

Feature Level Requirement:

Validated Feature: set_audio_output_configuration

Profile A Requirement: None

Profile C Requirement: None

Profile G Requirement: None

Profile Q Requirement: None

Profile S Requirement: None

Expected Scenarios Under Test:

1. Client connects to Device to change audio output configuration.

2. Client is considered as supporting Configure Audio Output Configuration if the following conditions are met:

- Client is able to retrieve audio output configuration options using **GetAudioOutputConfigurationOptions** operation AND

- Client is able to change audio output configuration settings using **SetAudioOutputConfiguration** operation.

3. Client is considered as NOT supporting Configure Audio Output Configuration if ANY of the following is TRUE:

- No valid responses for **GetAudioOutputConfigurationOptions** request OR

- No valid responses for **SetAudioOutputConfiguration** request.

GET AUDIO OUTPUT CONFIGURATION OPTIONS

Test Label: Configure Audio Output Configuration - Get Audio Output Configuration Options

Test Case ID: SETAUDIOOUTPUTCONFIGURATION-1

Profile A Reference: None

Profile C Reference: None

Profile G Reference: None

Profile Q Reference: None

Profile S Reference: None

Feature Under Test: Get Audio Output Configuration Options

Test Purpose: To verify that Client is able to get audio output configuration options provided by Device using the **GetAudioOutputConfigurationOptions** operation.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with **GetAudioOutputConfigurationOptions** operation present.

- Device supports Audio Outputs.

Test Procedure (expected to be reflected in network trace file):

1. Client invokes **GetAudioOutputConfigurationOptions** request message to retrieve audio output configuration options for the Device.

2. Device responses with code HTTP 200 OK and **GetAudioOutputConfigurationOptionsResponse** message.

Test Result:

PASS -

- Client **GetAudioOutputConfigurationOptions** request messages are valid according to XML Schemas listed in Namespaces AND

- Client **GetAudioOutputConfigurationOptions** request in Test Procedure fulfills the following requirements:

  - [S1] **soapenv:Body** element has child element **trt:GetAudioOutputConfigurationOptions** AND

- Device response to the **GetAudioOutputConfigurationOptions** request fulfills the following requirements:

  - [S2] It has HTTP 200 response code AND

  - [S3] **soapenv:Body** element has child element **trt:GetAudioOutputConfigurationOptionsResponse**.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: set_audio_output_configuration.get_audio_output_configuration_options

SET AUDIO OUTPUT CONFIGURATION

Test Label: Configure Audio Output Configuration - Set Audio Output Configuration

Test Case ID: SETAUDIOOUTPUTCONFIGURATION-2

Profile A Reference: None

Profile C Reference: None

Profile G Reference: None

Profile Q Reference: None

Profile S Reference: None

Feature Under Test: Set Audio Output Configuration

Test Purpose: To verify that Client is able to change audio output configuration provided by Device using the **SetAudioOutputConfiguration** operation.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with **SetAudioOutputConfiguration** operation present.

- Device supports Audio Outputs.

Test Procedure (expected to be reflected in network trace file):

1. Client invokes **SetAudioOutputConfiguration** request message to change audio output configuration on the Device.

2. Device responses with code HTTP 200 OK and **SetAudioOutputConfigurationResponse** message.

Test Result:

PASS -

- Client **SetAudioOutputConfiguration** request messages are valid according to XML Schemas listed in Namespaces AND

- Client **SetAudioOutputConfiguration** request in Test Procedure fulfills the following requirements:

  - [S1] **soapenv:Body** element has child element **trt:SetAudioOutputConfiguration** AND

- Device response to the **SetAudioOutputConfiguration** request fulfills the following requirements:

  - [S2] It has HTTP 200 response code AND

  - [S3] **soapenv:Body** element has child element **trt:SetAudioOutputConfigurationResponse**.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: set_audio_output_configuration.set_audio_output_configuration

A. Revision History

January 11, 2016 Version 16.01

- Typos were fixed.

December 18, 2015 Version 16.01

- General item (Test Overview) was added.

  Minor updates in formatting, typos, terms.

  TEARDOWN check was added to the Audio Backchannel Streaming Test Cases.

October 26, 2015 Version 15.10

- Initial version:

  General parts added

  Audio Backchannel Streaming added

  Get Audio Decoder Configurations List added

  Get Audio Output Configurations List added

  Get Audio Outputs List added

  Get Audio Decoder Configuration added

  Get Audio Output Configuration added

  Profile Configuration for Audio Backchannel added

  Configure Audio Decoder Configuration added

  Configure Audio Output Configuration added