ONVIF®

Other Features Client Test Specification

Version 19.12

December 2019

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

Vers.	Date	Description
19.12	Oct 04, 2019	Note about not found GetStreamUri was added in the following test cases according to #339: AUDIOBACKCHANNELSTREAMING-2 G.711 AUDIO BACKCHANNEL STREAMING AUDIOBACKCHANNELSTREAMING-3 G.726 AUDIO BACKCHANNEL STREAMING AUDIOBACKCHANNELSTREAMING-4 AAC AUDIO BACKCHANNEL STREAMING
19.12	Sep 18, 2019	The following was done according to #325: Scope\Supplementary Features and Test Cases sections was added. Supplementary Features and Test Cases sections was added.
19.12	Aug 14, 2019	Initial version.

Table of Contents

1 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).

This particular document defines test cases required for Client features that are out of any profiles. It also describes the test framework, test setup, prerequisites, test policies needed for the execution of the described test cases.

1.1 Scope

This ONVIF Other Features Client Test Specification defines and regulates the conformance testing procedure for the ONVIF conformant Clients in the scope of features which are out of any profile. 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 ONVIF Network Specification.

The principal intended purposes are:

- Provide self-assessment tool for implementations.
- Provide comprehensive test suite coverage for Audio Backchannel for Media features.
- Provide comprehensive test suite coverage for some Imaging features.
- Provide comprehensive test suite coverage for OSD features for Media.
- Provide comprehensive test suite coverage for TLS Enabled Version configuration.

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.

1.2 Audio Backchannel for Media Features

1.2.1 Audio Backchannel Streaming

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

1.2.2 Get Audio Decoder Configurations List

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

1.2.3 Get Audio Output Configurations List

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

1.2.4 Get Audio Outputs List

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

1.2.5 Get Audio Decoder Configuration

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

1.2.6 Get Audio Output Configuration

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

1.2.7 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.

1.2.8 Configure Audio Decoder Configuration

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

1.2.9 Configure Audio Output Configuration

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

1.3 Imaging Features

1.3.1 Get Imaging Capabilities

Get Imaging Capabilities section specifies Client ability to request imaging capabilities from Device.

1.4 OSD for Media Features

1.4.1 Get OSD Configuration

Get OSD Configuration section specifies Client ability to request OSD configuration from Device.

1.4.2 Get OSD List

Get OSD List section specifies Client ability to request OSD list from Device.

1.4.3 OSD Configuration

OSD Configuration section specifies Client ability to change OSD settings on Device.

1.5 Security Configuration Features

1.5.1 Enabled TLS Versions Configuration

Enabled TLS Versions Configuration section specifies Client ability to configure enabled TLS versions on Device.

2 Normative references

- ONVIF Conformance Process Specification:

  https://www.onvif.org/profiles/conformance/

- ONVIF Profile Policy:

  https://www.onvif.org/profiles/

- ONVIF Network Interface Specifications:

  https://www.onvif.org/profiles/specifications/

- ISO/IEC Directives, Part 2, Annex H:

  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/

- ONVIF Media Service Specification:

  https://www.onvif.org/profiles/specifications/

- ONVIF Streaming Specification:

  https://www.onvif.org/profiles/specifications/

- ONVIF Imaging Service Specification:

  https://www.onvif.org/profiles/specifications/

- ONVIF Security Configuration Specification:

  https://www.onvif.org/profiles/specifications/

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

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

3 Terms and Definitions

3.1 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].

3.2 Definitions

This section describes terms and definitions used in this document.

Address An address refers to a URI.

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.

Valid Device Response Device has responded to specific request with code HTTP or RTSP 200 OK and SOAP fault message has not appeared.

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

3.3 Abbreviations

This section describes abbreviations used in this document.

HTTP	Hyper Text Transport Protocol.
HTTPS	Hyper Text Transport Protocol over Secure Socket Layer.
IP	Internet Protocol.
IPv4	Internet Protocol version 4.
TCP	Transport Control Protocol.
UDP	User Datagram Protocol.
URI	Uniform Resource Identifier.
WSDL	Web Services Description Language.
XML	eXtensible Markup Language.
RTSP	Real Time Streaming Protocol.
RTP	Realtime Transport Protocol.
SDP	Session Description Protocol.
AAC	Advanced Audio Coding.
OSD	On-Screen Display.

3.4 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 3.1. 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
tds	http://www.onvif.org/ver10/device/wsdl	The namespace for the WSDL device service
tev	http://www.onvif.org/ver10/events/wsdl	The namespace for the WSDL event service
ter	http://www.onvif.org/ver10/error	The namespace for ONVIF defined faults
wsnt	http://docs.oasis-open.org/wsn/b-2	Schema namespace of the [WS-BaseNotification] specification.

Prefix	Namespace URI	Description
wsa	http://www.w3.org/2005/08/addressing	Device addressing namespace as defined by [WS-Addressing].
trt	http://www.onvif.org/ver10/media/wsdl	The namespace for the WSDL media service
timg	http://www.onvif.org/ver20/imaging/wsd	The namespace for the WSDL imaging service
tas	http://www.onvif.org/ver10/advancedsecurity/wsdl	The namespace for the WSDL Security Configuration service

4 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 with Media Service.

An ONVIF Client with Imaging features support can provide retrieve of Imaging capabilities.

An ONVIF Client with OSD features support can provide OSD configuration with Media Service.

An ONVIF Client with securitu configuration features support can provide TLS Enabled Versions Configuration configuration.

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.

4.1 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

4.1.1 Feature Level Requirement

Feature Level Requirement item contains a feature ID, check condition based on Device features, required number of Devices and feature requirement level for the Profiles, which will be used for Profiles conformance.

To claim this Feature as supported Client shall pass Expected Scenario Under Test:

- for each Device, which supports Device Features defined in Check Condition Based on Device Features
- for at least with number of Devices specified in Required Number of Devices

If Feature Level Requirement is defined as Mandatory for some Profile, Client shall support this Feature to claim this Profile Conformance.

4.1.2 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.

4.1.3 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-reqiusite 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.

4.2 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.

4.3 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.

5 Test Cases for Audio Backchannel for Media

5.1 Audio Backchannel Streaming Test Cases

5.1.1 Feature Level Requirement:

Validated Feature: Audio Backchannel Streaming (AudioBackchannelStreaming)

Check Condition based on Device Features: Audio Output (Media Service) is supported by Device.

Required Number of Devices: 1

Profile A Requirement: None

Profile C Requirement: None

Profile G Requirement: None

Profile Q Requirement: None

Profile S Requirement: None

5.1.2 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.

5.1.3 GET AUDIO DECODER CONFIGURATION OPTIONS

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

Test Case ID: AUDIOBACKCHANNELSTREAMING-1

Feature Under Test: Get Audio Decoder Configuration Options (AudioBackchannelStreaming_GetAudioDecoderConfigurationOptions)

Profile A Normative Reference: None

Profile C Normative Reference: None

Profile G Normative Reference: None

Profile Q Normative Reference: None

Profile S Normative Reference: None

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 responds 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.

5.1.4 G.711 AUDIO BACKCHANNEL STREAMING

Test Label: Audio Backchannel Streaming - G.711

Test Case ID: AUDIOBACKCHANNELSTREAMING-2

Feature Under Test: G.711 Audio Backchannel Streaming (AudioBackchannelStreaming_G711AudioBackchannelStreaming)

Profile A Normative Reference: None

Profile C Normative Reference: None

Profile G Normative Reference: None

Profile Q Normative Reference: None

Profile S Normative Reference: None

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 responds 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 responds 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 responds 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 responds 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. If Device sends response to RTSP TEARDOWN, it has 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:

Note: If no **GetStreamUri** (Media Service) corresponding to detected RTSP session found, the test will be assumed as NOT DETECTED.

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 Client **RTSP SETUP** request in Test Procedure that fulfills the following requirements:
    - [S5] It is invoked for the same Device as the response for **RTSP DESCRIBE** request AND
    - [S6] It is invoked after the Client **RTSP DESCRIBE** request AND
    - [S7] 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
    - [S8] **Require** tag in RTSP header contains "www.onvif.org/ver20/backchannel" AND
- Device response to the **RTSP SETUP** request fulfills the following requirements:
    - [S9] It has RTSP 200 response code AND
- There is a Device response to the **GetStreamUri** request in Test Procedure that fulfills the following requirements:
    - [S10] It has HTTP 200 response code AND
    - [S11] It is received from the same Device as the response for **RTSP DESCRIBE** request AND
    - [S12] It is received before the Client **RTSP DESCRIBE** request AND
    - [S13] 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 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
- If there is Device response on the **RTSP TEARDOWN** request then it fulfills the following requirements:
  - [S23] It has RTSP 200 response code.

FAIL -

- The Client failed PASS criteria.

5.1.5 G.726 AUDIO BACKCHANNEL STREAMING

Test Label: Audio Backchannel Streaming - G.726

Test Case ID: AUDIOBACKCHANNELSTREAMING-3

Feature Under Test: G.726 Audio Backchannel Streaming (AudioBackchannelStreaming_G726AudioBackchannelStreaming)

Profile A Normative Reference: None

Profile C Normative Reference: None

Profile G Normative Reference: None

Profile Q Normative Reference: None

Profile S Normative Reference: None

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 responds 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 responds 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 responds 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 responds 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. If Device sends response to RTSP TEARDOWN, it has 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:

Note: If no **GetStreamUri** (Media Service) corresponding to detected RTSP session found, the test will be assumed as NOT DETECTED.

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 Client **RTSP SETUP** request in Test Procedure that fulfills the following requirements:
  - [S5] It is invoked for the same Device as the response for **RTSP DESCRIBE** request AND
  - [S6] It is invoked after the Client **RTSP DESCRIBE** request AND
  - [S7] 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
  - [S8] **Require** tag in RTSP header contains "www.onvif.org/ver20/backchannel" AND
- Device response to the **RTSP SETUP** request fulfills the following requirements:
  - [S9] It has RTSP 200 response code AND
- There is a Device response to the **GetStreamUri** request in Test Procedure that fulfills the following requirements:
  - [S10] It has HTTP 200 response code AND
  - [S11] It is received from the same Device the response for **RTSP DESCRIBE** request AND
  - [S12] It is received before the Client **RTSP DESCRIBE** request AND
  - [S13] 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 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
- If there is Device response on the **RTSP TEARDOWN** request then it fulfills the following requirements:
  - [S23] It has RTSP 200 response code.

FAIL -

- The Client failed PASS criteria.

5.1.6 AAC AUDIO BACKCHANNEL STREAMING

Test Label: Audio Backchannel Streaming - AAC

Test Case ID: AUDIOBACKCHANNELSTREAMING-4

Feature Under Test: AAC Audio Backchannel Streaming (AudioBackchannelStreaming_AACAudioBackchannelStreaming)

Profile A Normative Reference: None

Profile C Normative Reference: None

Profile G Normative Reference: None

Profile Q Normative Reference: None

Profile S Normative Reference: None

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 responds 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 responds 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 responds 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 responds 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. If Device sends response to RTSP TEARDOWN, it has 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:

Note: If no **GetStreamUri** (Media Service) corresponding to detected RTSP session found, the test will be assumed as NOT DETECTED.

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" or "MP4A-LATM" AND
- There is Client **RTSP SETUP** request in Test Procedure that fulfills the following requirements:
  - [S5] It is invoked for the same Device as the response for **RTSP DESCRIBE** request AND
  - [S6] It is invoked after the Client **RTSP DESCRIBE** request AND
  - [S7] 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
  - [S8] **Require** tag in RTSP header contains "www.onvif.org/ver20/backchannel" AND
- Device response to the **RTSP SETUP** request fulfills the following requirements:
  - [S9] It has RTSP 200 response code AND
- There is a Device response to the **GetStreamUri** request in Test Procedure that fulfills the following requirements:
  - [S10] It has HTTP 200 response code AND
  - [S11] It is received from the same Device the response for **RTSP DESCRIBE** request AND
  - [S12] It is received before the Client **RTSP DESCRIBE** request AND
  - [S13] 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 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
- If there is Device response on the **RTSP TEARDOWN** request then it fulfills the following requirements:
  - [S23] It has RTSP 200 response code.

FAIL -

- The Client failed PASS criteria.

5.2 Get Audio Decoder Configurations List Test Cases

5.2.1 Feature Level Requirement:

Validated Feature: Get Audio Decoder Configurations (GetAudioDecoderConfigurationsList)

Check Condition based on Device Features: Audio Output (Media Service) is supported by Device.

Required Number of Devices: 1

Profile A Requirement: None

Profile C Requirement: None

Profile G Requirement: None

Profile Q Requirement: None

Profile S Requirement: None

5.2.2 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.

5.2.3 GET AUDIO DECODER CONFIGURATIONS

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

Test Case ID: GETAUDIODECODERCONFIGURATIONSLIST-1

Feature Under Test: Get Audio Decoder Configurations (GetAudioDecoderConfigurationsList_GetAudioDecoderConfigurations)

Profile A Normative Reference: None

Profile C Normative Reference: None

Profile G Normative Reference: None

Profile Q Normative Reference: None

Profile S Normative Reference: None

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 responds 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.

5.3 Get Audio Output Configurations List Test Cases

5.3.1 Feature Level Requirement:

Validated Feature: Get Audio Output Configurations (GetAudioOutputConfigurationsList)

Check Condition based on Device Features: Audio Output (Media Service) is supported by Device.

Required Number of Devices: 1

Profile A Requirement: None

Profile C Requirement: None

Profile G Requirement: None

Profile Q Requirement: None

Profile S Requirement: None

5.3.2 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.

5.3.3 GET AUDIO OUTPUT CONFIGURATIONS

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

Test Case ID: GETAUDIOOUTPUTCONFIGURATIONSLIST-1

Feature Under Test: Get Audio Output Configurations (GetAudioOutputConfigurationsList_GetAudioOutputConfigurations)

Profile A Normative Reference: None

Profile C Normative Reference: None

Profile G Normative Reference: None

Profile Q Normative Reference: None

Profile S Normative Reference: None

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 responds 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.

5.4 Get Audio Outputs List Test Cases

5.4.1 Feature Level Requirement:

Validated Feature: Get Audio Outputs (GetAudioOutputsList)

Check Condition based on Device Features: Audio Output (Media Service) is supported by Device.

Required Number of Devices: 1

Profile A Requirement: None

Profile C Requirement: None

Profile G Requirement: None

Profile Q Requirement: None

Profile S Requirement: None

5.4.2 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).

5.4.3 GET AUDIO OUTPUTS

Test Label: Get Audio Outputs List - Get Audio Outputs

Test Case ID: GETAUDIOOUTPUTSLIST-1

Feature Under Test: Get Audio Outputs (GetAudioOutputsList_GetAudioOutputs)

Profile A Normative Reference: None

Profile C Normative Reference: None

Profile G Normative Reference: None

Profile Q Normative Reference: None

Profile S Normative Reference: None

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 responds 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.

5.5 Get Audio Decoder Configuration Test Cases

5.5.1 Feature Level Requirement:

Validated Feature: Get Audio Decoder Configuration (GetAudioDecoderConfiguration)

Check Condition based on Device Features: Audio Output (Media Service) is supported by Device.

Required Number of Devices: 1

Profile A Requirement: None

Profile C Requirement: None

Profile G Requirement: None

Profile Q Requirement: None

Profile S Requirement: None

5.5.2 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.

5.5.3 GET AUDIO DECODER CONFIGURATION

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

Test Case ID: GETAUDIODECODERCONFIGURATION-1

Feature Under Test: Get Audio Decoder Configuration (GetAudioDecoderConfiguration_GetAudioDecoderConfigurationFeature)

Profile A Normative Reference: None

Profile C Normative Reference: None

Profile G Normative Reference: None

Profile Q Normative Reference: None

Profile S Normative Reference: None

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 responds 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.

5.6 Get Audio Output Configuration Test Cases

5.6.1 Feature Level Requirement:

Validated Feature: Get Audio Output Configuration (GetAudioOutputConfiguration)

Check Condition based on Device Features: Audio Output (Media Service) is supported by Device.

Required Number of Devices: 1

Profile A Requirement: None

Profile C Requirement: None

Profile G Requirement: None

Profile Q Requirement: None

Profile S Requirement: None

5.6.2 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.

5.6.3 GET AUDIO OUTPUT CONFIGURATION

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

Test Case ID: GETAUDIOOUTPUTCONFIGURATION-1

Feature Under Test: Get Audio Output Configuration (GetAudioOutputConfiguration_GetAudioOutputConfigurationFeature)

Profile A Normative Reference: None

Profile C Normative Reference: None

Profile G Normative Reference: None

Profile Q Normative Reference: None

Profile S Normative Reference: None

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 responds 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.

5.7 Profile Configuration for Audio Backchannel Test Cases

5.7.1 Feature Level Requirement:

Validated Feature: Profile Configuration for Audio Backchannel (ProfileConfigurationForAudioBackchannel)

Check Condition based on Device Features: Audio Output (Media Service) is supported by Device.

Required Number of Devices: 1

Profile A Requirement: None

Profile C Requirement: None

Profile G Requirement: None

Profile Q Requirement: None

Profile S Requirement: None

5.7.2 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.

5.7.3 GET COMPATIBLE AUDIO OUTPUT CONFIGURATIONS

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

Test Case ID: PROFILECONFIGURATIONFORAUDIOBACKCHANNEL-1

Feature Under Test: Get Compatible Audio Output Configurations (ProfileConfigurationForAudioBackchannel_GetCompatibleAudioOutputConfigurations)

Profile A Normative Reference: None

Profile C Normative Reference: None

Profile G Normative Reference: None

Profile Q Normative Reference: None

Profile S Normative Reference: None

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 responds 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.

5.7.4 ADD AUDIO OUTPUT CONFIGURATION

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

Test Case ID: PROFILECONFIGURATIONFORAUDIOBACKCHANNEL-2

Feature Under Test: Add Audio Output Configuration (ProfileConfigurationForAudioBackchannel_AddAudioOutputConfiguration)

Profile A Normative Reference: None

Profile C Normative Reference: None

Profile G Normative Reference: None

Profile Q Normative Reference: None

Profile S Normative Reference: None

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 responds 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 responds 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.

5.7.5 REMOVE AUDIO OUTPUT CONFIGURATION

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

Test Case ID: PROFILECONFIGURATIONFORAUDIOBACKCHANNEL-3

Feature Under Test: Remove Audio Output Configuration (ProfileConfigurationForAudioBackchannel_RemoveAudioOutputConfiguration)

Profile A Normative Reference: None

Profile C Normative Reference: None

Profile G Normative Reference: None

Profile Q Normative Reference: None

Profile S Normative Reference: None

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 responds 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.

5.7.6 GET COMPATIBLE AUDIO DECODER CONFIGURATIONS

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

Test Case ID: PROFILECONFIGURATIONFORAUDIOBACKCHANNEL-4

Feature Under Test: Get Compatible Audio Decoder Configurations (ProfileConfigurationForAudioBackchannel_GetCompatibleAudioDecoderConfigurations)

Profile A Normative Reference: None

Profile C Normative Reference: None

Profile G Normative Reference: None

Profile Q Normative Reference: None

Profile S Normative Reference: None

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 responds 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.

5.7.7 ADD AUDIO DECODER CONFIGURATION

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

Test Case ID: PROFILECONFIGURATIONFORAUDIOBACKCHANNEL-5

Feature Under Test: Add Audio Decoder Configuration (ProfileConfigurationForAudioBackchannel_AddAudioDecoderConfiguration)

Profile A Normative Reference: None

Profile C Normative Reference: None

Profile G Normative Reference: None

Profile Q Normative Reference: None

Profile S Normative Reference: None

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 responds 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 responds 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.

5.7.8 REMOVE AUDIO DECODER CONFIGURATION

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

Test Case ID: PROFILECONFIGURATIONFORAUDIOBACKCHANNEL-6

Feature Under Test: Remove Audio Decoder Configuration (ProfileConfigurationForAudioBackchannel_RemoveAudioDecoderConfiguration)

Profile A Normative Reference: None

Profile C Normative Reference: None

Profile G Normative Reference: None

Profile Q Normative Reference: None

Profile S Normative Reference: None

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 responds 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.

5.8 Configure Audio Decoder Configuration Test Cases

5.8.1 Feature Level Requirement:

Validated Feature: Configure Audio Decoder Configuration (SetAudioDecoderConfiguration)

Check Condition based on Device Features: Audio Output (Media Service) is supported by Device.

Required Number of Devices: 1

Profile A Requirement: None

Profile C Requirement: None

Profile G Requirement: None

Profile Q Requirement: None

Profile S Requirement: None

5.8.2 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.

5.8.3 SET AUDIO DECODER CONFIGURATION

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

Test Case ID: SETAUDIODECODERCONFIGURATION-1

Feature Under Test: Set Audio Decoder Configuration (SetAudioDecoderConfiguration_SetAudioDecoderConfigurationRequest)

Profile A Normative Reference: None

Profile C Normative Reference: None

Profile G Normative Reference: None

Profile Q Normative Reference: None

Profile S Normative Reference: None

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 responds 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.

5.9 Configure Audio Output Configuration Test Cases

5.9.1 Feature Level Requirement:

Validated Feature: Configure Audio Output Configuration (SetAudioOutputConfiguration)

Check Condition based on Device Features: Audio Output (Media Service) is supported by Device.

Required Number of Devices: 1

Profile A Requirement: None

Profile C Requirement: None

Profile G Requirement: None

Profile Q Requirement: None

Profile S Requirement: None

5.9.2 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.

5.9.3 GET AUDIO OUTPUT CONFIGURATION OPTIONS

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

Test Case ID: SETAUDIOOUTPUTCONFIGURATION-1

Feature Under Test: Get Audio Output Configuration Options (SetAudioOutputConfiguration_GetAudioOutputConfigurationOptions)

Profile A Normative Reference: None

Profile C Normative Reference: None

Profile G Normative Reference: None

Profile Q Normative Reference: None

Profile S Normative Reference: None

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 responds 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.

5.9.4 SET AUDIO OUTPUT CONFIGURATION

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

Test Case ID: SETAUDIOOUTPUTCONFIGURATION-2

Feature Under Test: Set Audio Output Configuration (SetAudioOutputConfiguration_SetAudioOutputConfigurationRequest)

Profile A Normative Reference: None

Profile C Normative Reference: None

Profile G Normative Reference: None

Profile Q Normative Reference: None

Profile S Normative Reference: None

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 responds 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.

6 Test Cases for Imaging

6.1 Get Imaging Capabilities Test Cases

6.1.1 Feature Level Requirement:

Validated Feature: Get Imaging Capabilities (GetImagingCapabilities)

Check Condition based on Device Features: Imaging Service is supported by Device.

Required Number of Devices: 1

6.1.2 Expected Scenarios Under Test:

1. Client connects to Device to retrieve a imaging capabilities.
2. Client is considered as supporting Get Imaging Capabilities if the following conditions are met:
    - Client is able to retrieve a imaging capabilities using **GetCapabilities** operation OR **GetServiceCapabilities** operation (Imaging Service) OR supports get_services_capabilities.get_services feature.
3. Client is considered as NOT supporting Get Imaging Capabilities if ANY of the following is TRUE:
    - No valid responses for **GetCapabilities** request if detected AND Device supportes GetCapabilities feature OR
    - No valid responses for **GetServiceCapabilities** request (Imaging Service) if detected AND Device supportes GetServices feature
    - No valid responses for **GetCapabilities** request AND no valid responses for **GetServiceCapabilities** request (Imaging Service) AND get_services_capabilities.get_services feature is not supported by Client.

6.1.3 GET CAPABILITIES

Test Label: Get Imaging Capabilities - Get Capabilities

Test Case ID: GETIMAGINGCAPABILITIES-1

Feature Under Test: Get Imaging Capabilities using Get Capabilities (GetImagingCapabilities_GetImgCapabilities)

Test Purpose: To verify that imaging capabilities provided by Device is received by Client using the **GetCapabilities** operation.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with **GetCapabilities** operation with **tds:Category** element equal to "All" OR "Imaging" OR without any **tds:Category** element present.
- Device supports Imaging Service.

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

1. Client invokes **GetCapabilities** request message with **tds:Category** element equal to "All" OR "Imaging" OR without any **tds:Category** element to retrieve imaging capabilities from the Device.
2. Device responds with code HTTP 200 OK and **GetCapabilitiesResponse** message.

Test Result:

PASS -

- Client **GetCapabilities** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **GetCapabilities** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **tds:GetCapabilities** AND
  - [S2] IF it contains any **tds:Category** element THEN it contains **tds:Category** element equal to "All" OR "Imaging" AND
- Device response on the **GetCapabilities** request fulfills the following requirements:
  - [S3] It has HTTP 200 response code AND
  - [S4] **soapenv:Body** element has child element **tds:GetCapabilitiesResponse**.

FAIL -

- The Client failed PASS criteria.

6.1.4 GET SERVICE CAPABILITIES

Test Label: Get Imaging Capabilities - Get Service Capabilities

Test Case ID: GETIMAGINGCAPABILITIES-2

Feature Under Test: Get Imaging Capabilities using Get Service Capabilities (GetImagingCapabilities_GetImgServiceCapabilities)

Test Purpose: To verify that imaging capabilities provided by Device is received by Client using the **GetServiceCapabilities** operation.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with **GetServiceCapabilities** operation for Imaging Service present.
- Device supports Imaging Service.

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

1. Client invokes **GetServiceCapabilities** request message to retrieve imaging capabilities from the Device.
2. Device responds with code HTTP 200 OK and **GetServiceCapabilitiesResponse** message.

Test Result:

PASS -

- Client **GetServiceCapabilities** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **GetServiceCapabilities** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **timg:GetServiceCapabilities** AND
- Device response on the **GetServiceCapabilities** request fulfills the following requirements:
  - [S2] It has HTTP 200 response code AND
  - [S3] **soapenv:Body** element has child element **timg:GetServiceCapabilitiesResponse**.

FAIL -

- The Client failed PASS criteria.

7 Test Cases for OSD for Media

7.1 Get OSD Configuration Test Cases

7.1.1 Feature Level Requirement:

Validated Feature: Get OSD Configuration (GetOSD)

Check Condition based on Device Features: TO BE DISCUSSED

Required Number of Devices: 1

7.1.2 Expected Scenarios Under Test:

1. Client connects to Device to retrieve a OSD configuration.

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

   - Client is able to retrieve a OSD configuration using **GetOSD** operation.

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

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

7.1.3 GET OSD

Test Label: Get OSD - Get OSD

Test Case ID: GETOSD-1

Feature Under Test: Get OSD (GetOSD_GetOsd)

Test Purpose: To verify that OSD list for Device is received by Client using the **GetOSD** operation.

Pre-Requisite:

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

- Device supports Media Service.

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

1. Client invokes **GetOSD** request message to retrieve OSD configuration from the Device.
2. Device responds with code HTTP 200 OK and **GetOSDResponse** message.

Test Result:

PASS -

- Client **GetOSD** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **GetOSD** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **trt:GetOSD** AND
  - [S2] **trt:OSDToken** element has non-empty string value of specific OSD token AND
- Device response on the **GetOSD** request fulfills the following requirements:
  - [S3] It has HTTP 200 response code AND
  - [S4] **soapenv:Body** element has child element **trt:GetOSDResponse**.

FAIL -

- The Client failed PASS criteria.

7.2 Get OSD List Test Cases

7.2.1 Feature Level Requirement:

Validated Feature: Get OSD List (GetOSDs)

Check Condition based on Device Features: TO BE DISCUSSED

Required Number of Devices: 1

7.2.2 Expected Scenarios Under Test:

1. Client connects to Device to retrieve a OSD list.
2. Client is considered as supporting Get OSD List if the following conditions are met:
   - Client is able to retrieve a OSD list using **GetOSDs** operation.
3. Client is considered as NOT supporting Get OSD List if ANY of the following is TRUE:

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

7.2.3 GET OSDS

Test Label: Get OSDs - Get OSDs

Test Case ID: GETOSDS-1

Feature Under Test: Get OSDs (GetOSDs_GetOsds)

Test Purpose: To verify that OSD list for Device is received by Client using the **GetOSDs** operation.

Pre-Requisite:

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

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

1. Client invokes **GetOSDs** request message to retrieve OSD list from the Device.
2. Device responds with code HTTP 200 OK and **GetOSDsResponse** message.

Test Result:

PASS -

- Client **GetOSDs** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **GetOSDs** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **trt:GetOSDs** AND
  - If it contains **trt:ConfigurationToken** element then it fulfills the following requirements (else skip the check):
    - [S2] **trt:ConfigurationToken** element has non-empty string value of specific video source configuraton token AND
- Device response on the **GetOSDs** request fulfills the following requirements:
  - [S3] It has HTTP 200 response code AND
  - [S4] **soapenv:Body** element has child element **trt:GetOSDsResponse**.

FAIL -

- The Client failed PASS criteria.

7.3 OSD Configuration Test Cases

7.3.1 Feature Level Requirement:

Validated Feature: OSD Configuration (SetOSD)

Check Condition based on Device Features: TO BE DISCUSSED

Required Number of Devices: 1

7.3.2 Expected Scenarios Under Test:

1. Client connects to Device to change OSD settings.
2. Client is considered as supporting OSD Configuration if the following conditions are met:
   - Client is able to retrieve a OSD options using **GetOSDOptions** operation AND
   - Client is able to change a OSD settings using **SetOSD** operation.
3. Client is considered as NOT supporting OSD Configuration if ANY of the following is TRUE:
   - No valid responses for **GetOSDOptions** request OR
   - No valid responses for **SetOSD** request.

7.3.3 GET OSD OPTIONS

Test Label: OSD Configuration - Get OSD Options

Test Case ID: SETOSD-1

Feature Under Test: Get OSD Options (SetOSD_GetOsdOptions)

Test Purpose: To verify that OSD options for Device is received by Client using the **GetOSDOptions** operation.

Pre-Requisite:

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

- Device supports Media Service.

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

1. Client invokes **GetOSDOptions** request message to retrieve OSD options for specified Video Source Configuration from the Device.
2. Device responds with code HTTP 200 OK and **GetOSDOptionsResponse** message.

Test Result:

PASS -

- Client **GetOSDOptions** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **GetOSDOptions** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **trt:GetOSDOptions** AND
  - [S2] **trt:ConfigurationToken** element has non-empty string value of specific video source configuraton token AND
- Device response on the **GetOSDOptions** request fulfills the following requirements:
  - [S3] It has HTTP 200 response code AND
  - [S4] **soapenv:Body** element has child element **trt:GetOSDOptionsResponse**.

FAIL -

- The Client failed PASS criteria.

7.3.4 SET OSD

Test Label: OSD Configuration - Set OSD

Test Case ID: SETOSD-2

Feature Under Test: Set OSD (SetOSD_SetOsd)

Test Purpose: To verify that Client is able to change OSD settings on Device using the **SetOSD** operation.

Pre-Requisite:

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

- Device supports Media Service.

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

1. Client invokes **GetOSDOptions** request message to retrieve OSD options for specified Video Source Configuration from the Device.
2. Device responds with code HTTP 200 OK and **GetOSDOptionsResponse** message.
3. Client invokes **SetOSD** request message to change OSD settings for specified OSD which are correspond to the recieved options on the Device.
4. Device responds with code HTTP 200 OK and **SetOSDResponse** message.

Test Result:

PASS -

- Client **SetOSD** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **SetOSD** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **trt:SetOSD** AND
- Device response on the **SetOSD** request fulfills the following requirements:
  - [S2] It has HTTP 200 response code AND
  - [S3] **soapenv:Body** element has child element **trt:SetOSDResponse** AND
- There is a Client **GetOSDOptions** request in Test Procedure fulfills the following requirements:
  - [S4] It invoked for the same Device as for the Client **SetOSD** request AND
  - [S5] It invoked before the Client **SetOSD** request AND
  - [S6] **trt:ConfigurationToken** element value is equal to **trt:OSD/tt:VideoSourceConfigurationToken** element from the **SetOSD** request AND
- Device response on the **GetOSDOptions** request fulfills the following requirements:
  - [S7] It has HTTP 200 response code.

FAIL -

- The Client failed PASS criteria.

8 Test Cases for Security Configuration

8.1 Enabled TLS Versions Configuration Test Cases

8.1.1 Feature Level Requirement:

Validated Feature: Enabled TLS Versions Configuration (EnabledTLSVersionsConfiguration)

Check Condition based on Device Features: Enabled TLS Versions (Security Configuration Service) is supported by the Device.

Required Number of Devices: 1

Profile A Requirement: None

Profile C Requirement: None

Profile G Requirement: None

Profile Q Requirement: None

Profile S Requirement: None

8.1.2 Expected Scenarios Under Test:

1. Client connects to Device configure enabled TLS versions on Device.

2. Client is considered as supporting Enabled TLS Versions Configuration if the following conditions are met:

   - Client is able to retrieve supported TLS versions using **GetServices** operation with IncludeCapability = true or using **GetServiceCapabilities** operation for Security Confguration Service if Device supports Enabled TLS Versions feature AND

     Client is able to setup enabled TLS versions using **SetEnabledTLSVersions** operation if Device supports Enabled TLS Versions feature.

3. Client is considered as NOT supporting Enabled TLS Versions Configuration if ANY of the following is TRUE:

   - No valid responses for **GetServices** request with IncludeCapability = true or for **GetServiceCapabilities** request for Security Confguration Service if detected if Device supports Enabled TLS Versions feature OR

- No valid responses for **SetEnabledTLSVersions** request if detected if Device supports Enabled TLS Versions feature OR
- No valid responses for **GetEnabledTLSVersions** request if detected if Device supports Enabled TLS Versions feature OR

8.1.3 Get Enabled TLS Versions

Test Case ID: ENABLEDTLSVERSIONSCONFIGURATION-1

Feature Under Test: Get Enabled TLS Versions (EnabledTLSVersionsConfiguration_GetEnabledTLSVersions)

Profile A Normative Reference: None

Profile C Normative Reference: None

Profile G Normative Reference: None

Profile Q Normative Reference: None

Profile S Normative Reference: None

Test Purpose: To verify that Client is able to get currently enabled TLS versions from Device using **GetEnabledTLSVersions** operation.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with **GetEnabledTLSVersions** operation present.
- Device supports Security Configuration Service.
- Device supports Enabled TLS Versions (Security Configuration Service).

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

1. Client invokes **GetEnabledTLSVersions** request message to get currently enabled TLS versions from Device.
2. Device responds with code HTTP 200 OK and **GetEnabledTLSVersionsResponse** message.

Test Result:

PASS -

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

- Client **GetEnabledTLSVersions** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **tas:GetEnabledTLSVersions** AND

- Device response on the **GetEnabledTLSVersions** request fulfills the following requirements:
  - [S2] It has HTTP 200 response code AND
  - [S3] **soapenv:Body** element has child element **tas:GetEnabledTLSVersionsResponse**.

FAIL -

- The Client failed PASS criteria.

8.1.4 Set Enabled TLS Versions

Test Case ID: ENABLEDTLSVERSIONSCONFIGURATION-2

Feature Under Test: Set Enabled TLS Versions (EnabledTLSVersionsConfiguration_SetEnabledTLSVersions)

Profile A Normative Reference: None

Profile C Normative Reference: None

Profile G Normative Reference: None

Profile Q Normative Reference: None

Profile S Normative Reference: None

Test Purpose: To verify that Client is able to setup enabled TLS versions on Device using **SetEnabledTLSVersions** operation.

Pre-Requisite:

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

- Device supports Security Configuration Service.

- Device supports Enabled TLS Versions (Security Configuration Service).

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

1. Client invokes **GetServices** request message with IncludeCapability = true or **GetServiceCapabilities** request message for the Security Confguration Service request message to get supported TLS versions from a Device.

2. Device responds with code HTTP 200 OK and **GetServicesResponse** message or **GetServiceCapabilitiesResponse** message with Security Configuration Service capabilities.

3. Client invokes **SetEnabledTLSVersions** request message with non empty list to configure enabled TLS versions on a Device.

Test Result:

PASS -

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

- Client **SetEnabledTLSVersions** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **tas:SetEnabledTLSVersions** AND
  - [S2] **tas:Versions** element contqains at least one TLS version AND

- Device response on the **SetEnabledTLSVersions** request fulfills the following requirements:
  - [S3] It has HTTP 200 response code AND
  - [S4] **soapenv:Body** element has child element **tas:SetEnabledTLSVersionsResponse**.

- There is a Client **GetServices** request or **GetServiceCapabilities** request in Test Procedure that fulfills the following requirements:
  - [S5] It invoked before **SetEnabledTLSVersions** request AND
  - If **GetServices** was detected:
    - [S6] **soapenv:Body** element has child element **tds:GetServices** AND
    - [S7] **tds:IncludeCapability** element is equal to true AND
  - If **GetServiceCapabilities** was detected:
    - [S8] **soapenv:Body** element has child element **tas:GetServiceCapabilities** AND

- If **GetServices** was detected Device response on the **GetServices** request fulfills the following requirements:
  - [S9] It has HTTP 200 response code AND
  - [S10] **soapenv:Body** element has child element **tds:GetServicesResponse**.

- If **GetServiceCapabilities** was detected Device response on the **GetServiceCapabilities** request fulfills the following requirements:
  - [S11] It has HTTP 200 response code AND
  - [S12] **soapenv:Body** element has child element **tas:GetServiceCapabilitiesResponse**.

FAIL -

- The Client failed PASS criteria.

Annex A Test for Appendix A

A.1 Required Number of Devices Summary

Required number of devices and Device feature dependency used in this test specification are listed in the Table.

Table A.1. Required Number of Devices Summary

Feature ID	Feature Name	Required Number of Devices	Check Condition based on Device Features	Check Condition based on Device Features ID
tc.AudioBackchannelStreaming	Audio Backchannel Streaming	1	Audio Output (Media Service) is supported by Device.	AudioOutput
tc.GetAudioDecoderConfigurationsList	Get Audio Decoder Configurations List	1	Audio Output (Media Service) is supported by Device.	AudioOutput
tc.GetAudioOutputConfigurationsList	Get Audio Output Configurations List	1	Audio Output (Media Service) is supported by Device.	AudioOutput
tc.GetAudioOutputsList	Get Audio Outputs List	1	Audio Output (Media Service) is supported by Device.	AudioOutput
tc.GetAudioDecoderConfiguration	Get Audio Decoder Configuration	1	Audio Output (Media Service) is supported by Device.	AudioOutput
tc.GetAudioOutputConfiguration	Get Audio Output Configuration	1	Audio Output (Media Service) is supported by Device.	AudioOutput
tc.ProfileConfigurationForAudioBackchannel	Profile Configuration for Audio Backchannel	1	Audio Output (Media Service) is supported by Device.	AudioOutput

Feature ID	Feature Name	Required Number of Devices	Check Condition based on Device Features	Check Condition based on Device Features ID
tc.SetAudioDecoderConfiguration	Configure Audio Decoder Configuration	1	Audio Output (Media Service) is supported by Device.	AudioOutput
tc.SetAudioOutputConfiguration	Configure Audio Output Configuration	1	Audio Output (Media Service) is supported by Device.	AudioOutput
tc.GetImagingCapabilities	Get Imaging Capabilities	1	Imaging Service is supported by Device.	ImagingService
tc.GetOSD	Get OSD Configuration	1	TO BE DISCUSSED	TBD
tc.GetOSDs	Get OSD List	1	TO BE DISCUSSED	TBD
tc.SetOSD	OSD Configuration	1	TO BE DISCUSSED	TBD
tc.EnabledTLSVersionsConfiguration	Enabled TLS Versions Configuration	1	Enabled TLS Versions (Security Configuration Service) is supported by the Device.	EnabledTLSVersions