ONVIF™

Profile T Client Test Specification

Version 17.06

June 2017

REVISION HISTORY

Vers.	Date	Description
17.06	Jun 15, 2017	Links in Normative references section were updated.
17.06	Jun 13, 2017	The following test cases added accoring to #201: Get Snapshot Uri Using Media2.
17.06	Jun 9, 2017	The following test cases added accoring to #201: Analytics Profile Configuration Using Media2 Test Cases Video Source Mode Test Cases Audio Source Configuration Using Media2 Test Cases Audio Output Configuration Using Media2 Test Cases Audio Decoder Configuration Using Media2 Test Cases List Video Source Configurations Using Media2 Test Cases OSD Configuration Using Media2 Test Cases
17.06	Jun 07, 2017	The following Device IO test cases moved into ONVIF Device IO Client Test Specification accoring to #194: Relay Outputs Using Device IO Get Digital Inputs
17.06	Jun 06, 2017	The following PTZ test cases moved into ONVIF PTZ Client Test Specification accoring to #194: PTZ Using Media2 Absolute Positioning PTZ Using Media2 Continuous Positioning PTZ - Set Preset Annex A.1 Get default PTZ space of PTZ Configuration corresponding to Move Operation
17.06	Feb 06, 2017	• Initial version: General parts added
17.06	Mar 17, 2017	• Media2 Video Streaming Test Cases added • Media2 Video Encoder Configuration Test Cases added
17.06	Mar 20, 2017	• Media2 Audio Encoder Configuration Test Cases added
17.06	Mar 21, 2017	• Media2 Media Streaming Test Cases added
17.06	Mar 23, 2017	• Relay Outputs Using DeviceIO Test Cases added
17.06	Mar 24, 2017	• Digital Inputs Test Cases added • Audio Backchannel Streaming Using Media2 Test Cases added
17.06	Mar 27, 2017	• PTZ Using Media2 Absolute Positioning Test Cases added

17.06	Mar 28, 2017	• Get Audio Output Configurations Using Media2 Test Cases added
17.06	Mar 29, 2017	• Audio Output Profile Configuration Using Media2 added
17.06	Apr 03, 2017	• PTZ Using Media2 Continuous Positioning Test Cases added
17.06	Apr 04, 2017	• PTZ - Set Preset Test Cases added • Audio Profile Configuration Using Media2 Test Cases added
17.06	May 29, 2017	• Audio Profile Configuration Using Media Test Cases updated according to #191 • Audio Output Profile Configuration Using Media2 Test Cases updated according to #193 • Get Audio Output Configurations Using Media2 Test Cases removed according to #193

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 testing Profile T features of a Client application e.g. Video Streaming, Video Encoder Configuration, Audio Streaming, Configuration of Audio Profile, Audio Source Configuration, Audio Encoder Configuration, Audio Output Streaming, Configuration of Audio Output Profile, Audio Output Configuration, Metadata Configuration, Relay Outputs, Digital Inputs, and Tampering. 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 Profile T Client Test Specification defines and regulates the conformance testing procedure for the ONVIF conformant Clients in the scope of Profile T features. Conformance testing is meant to be black-box network traces analysis and verification. The objective of this specification is to provide the test cases to test individual requirements of ONVIF Clients in the scope of Profile T features according to ONVIF Profile Specifications.

The principal intended purposes are:

- Provide self-assessment tool for implementations.
- Provide comprehensive test suite coverage for Profile T features.

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

- 3rd parties Client use cases
- Non-functional (performance and regression) testing and analysis.
- SOAP Implementation Interoperability test i.e. Web Services Interoperability Basic Profile version 2.0 (WS-I BP2.0).
- Network protocol implementation Conformance test for HTTPS and HTTP protocols.

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

1.2 Get Profiles Using Media2

Get Profiles Using Media2 section specifies Client ability to retrieve a list of Media2 profiles from Device.

1.3 Get Stream Uri Using Media2

Get Stream Uri Using Media2 section specifies Client ability to retrieve a Media2 stream URI from Device.

1.4 Media Streaming Using Media2

Media Streaming Using Media2 section defines different streaming options based on RTP protocol which are required for all types of streams of video, audio and metadata. Media control is done using RTSP protocol.

1.5 Video Streaming Using Media2

Media Streaming Using Media2 section specifies Client ability to establish specific video streams in H264 and H265 video formats.

1.6 Video Encoder Configuration Using Media2

Video Encoder Configuration Using Media2 section specifies modification of video encoder configurations on Device.

1.7 Audio Encoder Configuration Using Media2

Audio Encoder Configuration Using Media2 section specifies listing and modification of audio encoder configurations on Device.

1.8 Audio Streaming Using Media2

Audio Streaming Using Media2 section specifies Client ability to establish specific audio streams in G.711 and AAC audio formats.

1.9 Audio Profile Configuration Using Media2

Audio Profile Configuration Using Media2 section specifies Client ability to configure or create media profile with audio source configuration and to add audio encoder configuration to a media profile.

1.10 Audio Backchannel Streaming Using Media2

Audio Backchannel Streaming Using Media2 section specifies Client ability to stream audio for backchannel to Device.

1.11 Audio Output Profile Configuration Using Media2

Audio Output Profile Configuration Using Media2 section specifies Client ability to configure or create media profile with audio output configuration and to add audio decoder configuration to a media profile.

1.12 Analytics Profile Configuration Using Media2

Analytics Profile Configuration Using Media2 section specifies Client ability to add analytics configuration to a media profile.

1.13 Video Source Mode

Video Source Mode section specifies Client ability to request the information for current video source mode and settable video source modes and to change current video source mode on device.

1.14 Audio Source Configuration Using Media2

Audio Source Configuration Using Media2 section specifies listing and modification of audio source configurations on Device.

1.15 Audio Output Configuration Using Media2

Audio Output Configuration Using Media2 section specifies listing and modification of audio output configurations on Device.

1.16 Audio Decoder Configuration Using Media2

Audio Decoder Configuration Using Media2 section specifies listing and modification of audio decoder configurations on Device.

1.17 Get Snapshot Uri Using Media2

Get Snapshot Uri Using Media2 section specifies Client ability to obtain a JPEG snapshot from the device.

2 Normative references

- ONVIF Conformance Process Specification:

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

- ONVIF Profile Policy:

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

- ONVIF Core Specifications:

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

- ONVIF Core Client Test Specification:

  https://www.onvif.org/profiles/conformance/client-test/

- ONVIF Profile T Specification:

  https://www.onvif.org/profiles/profile-t/

- ONVIF Media2 Service Specification:

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

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

  http://www.iso.org/directives

- ISO 16484-5:2014-09 Annex P:

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

- WS-BaseNotification:

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

- W3C SOAP 1.2, Part 1, Messaging Framework:

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

- W3C XML Schema Part 1: Structures Second Edition:

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

- W3C XML Schema Part 2: Datatypes Second Edition:

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

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.

Profile	See ONVIF Profile Policy.
Profile T	The Profile T Specification.
ONVIF Device	Computer appliance or software program that exposes one or multiple ONVIF Web Services.
ONVIF Client	Computer appliance or software program that uses ONVIF Web Services.
Conversation	A Conversation is all exchanges between two MAC addresses that contains SOAP request and response.
Network	A network is an interconnected group of devices communicating using the Internet protocol.
Network Trace Capture file	Data file created by a network protocol analyzer software (such as Wireshark). Contains network packets data recorded during a live network communications.
SOAP	SOAP is a lightweight protocol intended for exchanging structured information in a decentralized, distributed environment. It uses XML technologies to define an extensible messaging framework providing a message construct that can be exchanged over a variety of underlying protocols.
Client Test Tool	ONVIF Client Test Tool that tests ONVIF Client implementation towards the ONVIF Test Specification set.
Configuration Entity	A network video device media abstract component that produces or consumes a media stream on the network, i.e. video and/or audio stream.
Configuration Entity	A network video device media abstract component that produces or consumes a media stream on the network, i.e. video and/or audio stream.
Digital PTZ	Function that diminishes or crops an image to adjust the image position and ratio.
GZIP	GNU data format for lossless compression.
Media Profile	Maps a video and audio sources and outputs encoders as well as PTZ and analytics configurations.

Metadata	All streaming data except video and audio, including video analytics results, PTZ position data and other metadata (such as textual data from POS applications).
Reference Token	Token provided by the device to uniquely reference an instance of a physicalIO, configuration or profile.
Video Analytics	Algorithms or programs used to analyze video data and to generate data describing object location and behaviour.

3.3 Abbreviations

This section describes abbreviations used in this document.

RTCP	RTP Control Protocol.
RTP	Realtime Transport Protocol.
RTSP	Real Time Streaming Protocol.
TCP	Transmission Control Protocol.
UDP	User Datagram Protocol.
XML	eXtensible Markup Language.
HTTP	Hyper Text Transport Protocol.
HTTPS	Hyper Text Transport Protocol over Secure Socket Layer.
URI	Uniform Resource Identifier.
WSDL	Web Services Description Language.
XML	eXtensible Markup Language.

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]
tt	http://www.onvif.org/ver10/schema	ONVIF XML schema descriptions
tr2	http://www.onvif.org/ver20/media/wsdl	The namespace for the WSDL Media2 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 compliant to Profile T can configure, request, and control streaming of video, audio, and audio output data over an IP network from an ONVIF Device compliant to the Profile T. The client can also retrieve and receive standardized Tampering related events.

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 Normative Reference
- Expected Scenarios Under Test
- List of Test Cases

4.1.1 Feature Level Normative Reference

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

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

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

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

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 compatibility with the Profile T, 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 Get Profiles Using Media2 Test Cases

5.1 Feature Level Normative Reference:

Validated Feature: GetProfiles

Profile T Requirement: Mandatory

5.2 Expected Scenarios Under Test:

1. Client connects to Device to retrive media profiles from the DUT.
2. Client is considered as supporting Get Profiles if the following conditions are met:
   - Client is able to retrieve media profiles using **GetProfiles** operation (Media2 Service).
3. Client is considered as NOT supporting Get Profiles if ANY of the following is TRUE:
   - No valid response to **GetProfiles** request (Media2 Service).

5.3 GET PROFILES USING MEDIA2

Test Label: GetProfiles

Test Case ID: MEDIA2_GETPROFILES-1

Profile T Normative Reference: Mandatory

Feature Under Test: Get Profiles

Test Purpose: To verify that media profiles provided by Device are received by Client using the **GetProfiles** operation.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with **GetProfiles** operation for Media2 Service present.
- Device supports Media2 Service (Media2Service).

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

1. Client invokes **GetProfiles** request message to retrieve a media profile or a list of media profiles from the Device.

2. Device responds with code HTTP 200 OK and **GetProfilesResponse** message.

Test Result:

PASS -

- Client **GetProfiles** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **GetProfiles** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **tr2:GetProfiles** AND
- Device response on the **GetProfiles** request fulfills the following requirements:
  - [S2] It has HTTP 200 response code AND
  - [S3] **soapenv:Body** element has child element **tr2:GetProfilesResponse**.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: Media2_GetProfiles.Media2_GetProfilesRequest

6 Get Stream Uri Using Media2 Test Cases

6.1 Feature Level Normative Reference:

Validated Feature: GetStreamUri

Profile T Requirement: Mandatory

6.2 Expected Scenarios Under Test:

1. Client connects to Device to retrive stream uri.
2. Client is considered as supporting Get Stream Uri if the following conditions are met:
   - Client is able to get the stream URI for the selected media profile using **GetStreamURI** operation (Media2 Service).
3. Client is considered as NOT supporting Get Stream Uri if ANY of the following is TRUE:
   - No valid response to **GetStreamURI** request (Media2 Service).

6.3 GET STREAM URI USING MEDIA2

Test Label: GetStreamUri

Test Case ID: MEDIA2_GETSTREAMURI-1

Profile T Normative Reference: Mandatory

Feature Under Test: Get Stream URI

Test Purpose: To verify that stream URI provided by Device is received by Client using the **GetStreamUri** operation.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with **GetStreamUri** operation for Media2 Service present.
- Device supports Media2 Service (Media2Service).

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

1. Client invokes **GetStreamUri** request message to retrieve a stream URI from the Device.

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

Test Result:

PASS -

- Client **GetStreamUri** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **GetStreamUri** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **tr2:GetStreamUri** AND
- Device response on the **GetStreamUri** request fulfills the following requirements:
  - [S2] It has HTTP 200 response code AND
  - [S3] **soapenv:Body** element has child element **tr2:GetStreamUriResponse**.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: media2_get_stream_uri.get_stream_uri

7 Media Streaming Using Media2 Test Cases

7.1 Feature Level Requirement:

Validated Feature: MediaStreaming

Profile T Requirement: Mandatory

7.2 Expected Scenarios Under Test:

1. Client connects to Device to initiate Media Streaming.
2. Client is considered as supporting Media Streaming if the following conditions are met:
   - Device supports media2_get_profiles.get_profiles feature AND
   - Device supports media2_get_stream_uri.get_stream_uri feature AND
   - Stream was successfully established by Client using UDP protocol OR HTTP protocol.
   - Stream was successfully established by Client using RTSP protocol (if supported).
3. Client is considered as NOT supporting Media Streaming if the following is TRUE:
   - Device does not support media2_get_profiles.get_profiles feature OR
   - Device does not support media2_get_stream_uri.get_stream_uri feature OR
   - Client is unable to establish stream using UDP protocol OR HTTP protocol OR
   - Client is unable to establish stream using RTSP protocol if detected.

7.3 STREAMING OVER RTSP USING MEDIA2

Test Label: Media Streaming - RTSP

Test Case ID: MEDIA2_MEDIASTREAMING-1

Profile T Normative Reference: Optional

Feature Under Test: Media Streaming using Media2

Test Purpose: To verify that stream over RTSP protocol was successfully established between Client and Device using RTSP commands and then successfully stopped.

Pre-Requisite:

- Device supports RTSP streaming for Media2 Service (Media2_RTPRTSPTCP).
- The Network Trace Capture files contains at least one Conversation between Client and Device with RTSP SETUP request with transport parameter as "RTP/AVP/TCP" and which does not contain Require header with "onvif-replay" value and which is not tunneled in HTTP present.
- The Network Trace Capture files contains at least one Conversation between Client and Device with **GetStreamUri** for Media2 Service with **rt2:Protocol** element value equals to RTSP.

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

1. Client invokes **GetStreamUri** request message for media profile with Transport element with "RTSP" value.
2. Device responds with code HTTP 200 OK and **GetStreamUriResponse** message.
3. Client invokes **RTSP DESCRIBE** request to retrieve media stream description.
4. Device responds with code RTSP 200 OK.
5. Client invokes **RTSP SETUP** request with **Transport** tag in RTSP header that contains "RTP/AVP/TCP" and without "onvif-replay" Require header to set media session parameters.
6. Device responds with code RTSP 200 OK.
7. Client invokes **RTSP PLAY** request without "onvif-replay" Require header to start media stream.
8. Device responds with code RTSP 200 OK.
9. Client invokes **RTSP TEARDOWN** request to terminate the RTSP session.
10. If Device sends response to RTSP TEARDOWN, it has code RTSP 200 OK or RTSP 454.

Test Result:

PASS -

- Client **RTSP SETUP** request in Test Procedure fulfills the following requirements:
  - [S1] It contains **Transport** request header field with value is equal to "RTP/AVP/TCP" (transport=RTP, profile=AVP, lower-transport=TCP) (see [RFC 2326]) AND
  - [S2] It does not contain **Require** request header field with value is equal to "onvif-replay" AND

- [S3] It is not tunneled in HTTP AND

- Device response on the **RTSP SETUP** request fulfills the following requirements:
  - [S4] It has RTSP 200 response code AND
- There is Client **RTSP DESCRIBE** request in Test Procedure fulfills the following requirements:
  - [S5] It invoked for the same Device as for the Client **RTSP SETUP** request AND
  - [S6] It invoked before the Client **RTSP SETUP** request AND
  - [S7] It is not tunneled in HTTP AND
- Device response on the **RTSP DESCRIBE** request fulfills the following requirements:
  - [S8] SDP packet contains media type with Control URL that was used to send **RTSP SETUP** (see [RFC 2326, C.1.1 Control URL]) AND
  - [S9] It has RTSP 200 response code AND
- There is a Client **GetStreamUri** request in Test Procedure fulfills the following requirements:
  - [S10] It invoked for the same Device as for the Client **RTSP DESCRIBE** request AND
  - [S11] It invoked before the Client **RTSP DESCRIBE** request AND
  - [S12] **tr2:GetStreamUri/tr2:Protocol** element value is equal to "RTSP"
- Device response on the **GetStreamUri** request to Media2 Service fulfills the following requirements:
  - [S13] It has HTTP 200 response code AND
  - [S14] It contains **tr2: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 fulfills the following requirements:
  - [S15] It is invoked for the same RTSP session as the Client **RTSP SETUP** request AND
  - [S16] It invoked after the Client **RTSP SETUP** request AND
  - [S17] RTSP address that was used to send it is correspond to any media Control URL or session Control URL or Content-Base URL from SDP packet (see [RFC 2326, C.1.1 Control URL]) AND

- [S18] It does not contain **Require** request header field with value is equal to "onvif-replay" AND
- [S19] It is not tunneled in HTTP AND

- Device response on the **RTSP PLAY** request fulfills the following requirements:
  - [S20] It has RTSP 200 response code AND
- There is Client **RTSP TEARDOWN** request in Test Procedure fulfills the following requirements:
  - [S21] It invoked for the same RTSP session as the Client **RTSP SETUP** request AND
  - [S22] It invoked after the Client **RTSP PLAY** request AND
  - [S23] RTSP address that was used to send it is correspond to any media Control URL or session Control URL or Content-Base URL from SDP packet (see [RFC 2326, C.1.1 Control URL]) AND
  - [S24] It is not tunneled in HTTP AND
- If there is Device response on the **RTSP TEARDOWN** request then it fulfills the following requirements:
  - [S25] It has RTSP 200 response code.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: media2_media_streaming.RTP_RTSP_TCP

7.4 STREAMING OVER UDP USING MEDIA2

Test Label: Media Streaming - UDP

Test Case ID: MEDIA2_MEDIASTREAMING-2

Profile T Normative Reference: Mandatory

Feature Under Test: Media Streaming using Media2

Test Purpose: To verify that stream over UDP protocol was successfully established between Client and Device using RTSP commands and then successfully stopped.

Pre-Requisite:

- Device supports Media2 Real Time Streaming (Media2_RealTimeStreaming).
- The Network Trace Capture files contains at least one Conversation between Client and Device with RTSP SETUP request with transport parameter as "RTP/AVP/UDP" or "RTP/AVP" and which does not contain Require header with "onvif-replay" value present.
- The Network Trace Capture files contains at least one Conversation between Client and Device with **GetStreamUri** for Media2 Service with **rt2:Protocol** element value equals to "RtspUnicast" or "RtspMulticast".

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

1. Client invokes **GetStreamUri** request message for media profile with Transport element with "RtspUnicast" value or "RtspMulticast" value.
2. Device responds with code HTTP 200 OK and **GetStreamUriResponse** message.
3. Client invokes **RTSP DESCRIBE** request to retrieve media stream description.
4. Device responds with code RTSP 200 OK.
5. Client invokes **RTSP SETUP** request with **Transport** tag in RTSP header that contains "RTP/AVP/UDP" or "RTP/AVP" and without "onvif-replay" Require header to set media session parameters.
6. Device responds with code RTSP 200 OK.
7. Client invokes **RTSP PLAY** request without "onvif-replay" Require header to start media stream.
8. Device responds with code RTSP 200 OK.
9. Client invokes **RTSP TEARDOWN** request to terminate the RTSP session.
10. If Device sends response to RTSP TEARDOWN, it has code RTSP 200 OK or RTSP 454.

Test Result:

PASS -

- Client **RTSP SETUP** request in Test Procedure fulfills the following requirements:
  - [S1] It contains **Transport** request header field with value is equal to "RTP/AVP/UDP" OR "RTP/AVP" (transport=RTP, profile=AVP, lower-transport=TCP or skipped) (see [RFC 2326]) AND
  - [S2] It does not contain **Require** request header field with value is equal to "onvif-replay" AND

- Device response on the **RTSP SETUP** request fulfills the following requirements:
  - [S3] It has RTSP 200 response code AND
- There is Client **RTSP DESCRIBE** request in Test Procedure fulfills the following requirements:
  - [S4] It invoked for the same Device as for the Client **RTSP SETUP** request AND
  - [S5] It invoked before the Client **RTSP SETUP** request AND
- Device response on the **RTSP DESCRIBE** request fulfills the following requirements:
  - [S6] SDP packet contains media type with Control URL that was used to send **RTSP SETUP** (see [RFC 2326, C.1.1 Control URL]) AND
  - [S7] It has RTSP 200 response code AND
- There is a Device **GetStreamUri** request in Test Procedure fulfills the following requirements:
  - [S8] It invoked for the same Device as for the Client **RTSP DESCRIBE** request AND
  - [S9] It invoked before the Client **RTSP DESCRIBE** request AND
  - [S10] **tr2:GetStreamUri/tr2:Protocol** element value is equal to "RtspUnicast" or "RtspMulticast"
- Device response on the **GetStreamUri** request to Media2 Service fulfills the following requirements:
  - [S11] It contains **tr2:Uri** element which value is equal to RTSP address that was used to send the **RTSP DESCRIBE** request AND
  - [S12] It has HTTP 200 response code AND
- There is Client **RTSP PLAY** request in Test Procedure fulfills the following requirements:
  - [S13] It invoked for the same RTSP session as the Client **RTSP SETUP** request AND
  - [S14] It invoked after the Client **RTSP SETUP** request AND
  - [S15] RTSP address that was used to send it is correspond to any media Control URL or session Control URL or Content-Base URL from SDP packet (see [RFC 2326, C.1.1 Control URL]) AND
  - [S16] It does not contain **Require** request header field with value is equal to "onvif-replay" AND
- Device response on the **RTSP PLAY** request fulfills the following requirements:

- [S17] It has RTSP 200 response code AND

- There is Client **RTSP TEARDOWN** request in Test Procedure fulfills the following requirements:
  - [S18] It invoked for the same RTSP session as the Client **RTSP SETUP** request AND
  - [S19] It invoked after the Client **RTSP PLAY** request AND
  - [S20] RTSP address that was used to send it is correspond to any media Control URL or session Control URL or Content-Base URL from SDP packet (see [RFC 2326, C.1.1 Control URL]) AND
- If there is Device response on the **RTSP TEARDOWN** request then it fulfills the following requirements:
  - [S21] It has RTSP 200 response code.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: media2_media_streaming.RTP_UDP

7.5 STREAMING OVER HTTP USING MEDIA2

Test Label: Media Streaming - HTTP

Test Case ID: MEDIA2_MEDIASTREAMING-3

Profile T Normative Reference: Mandatory

Feature Under Test: Media Streaming using Media2

Test Purpose: To verify that stream over HTTP protocol was successfully established between Client and Device using RTSP commands and then successfully stopped.

Pre-Requisite:

- Device supports HTTP streaming for Media2 Service (Media2_RTPRTSPHTTP).
- The Network Trace Capture files contains at least one Conversation between Client and Device with RTSP SETUP request with transport parameter as "RTP/AVP/TCP" and which does not contain Require header with "onvif-replay" value and which is tunneled in HTTP present.

- The Network Trace Capture files contains at least one Conversation between Client and Device with **GetStreamUri** for Media2 Service with **rt2:Protocol** element value equals to RtspOverHttp.

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

1. Client invokes **GetStreamUri** request message for media profile with Protocol element with "RtspOverHttp" value.
2. Device responds with code HTTP 200 OK and **GetStreamUriResponse** message.
3. Client invokes **RTSP DESCRIBE** request in HTTP tunnel to retrieve media stream description.
4. Device responds with code RTSP 200 OK.
5. Client invokes **RTSP SETUP** request without "onvif-replay" Require header in HTTP tunnel with **Transport** tag in RTSP header that contains "RTP/AVP/TCP" to set media session parameters.
6. Device responds with code RTSP 200 OK.
7. Client invokes **RTSP PLAY** request without "onvif-replay" Require header in HTTP tunnel to start media stream.
8. Device responds with code RTSP 200 OK.
9. Client invokes **RTSP TEARDOWN** request in HTTP tunnel to terminate the RTSP session.
10. If Device sends response to RTSP TEARDOWN, it has code RTSP 200 OK or RTSP 454.

Test Result:

PASS -

- Client **RTSP SETUP** request in Test Procedure fulfills the following requirements:
  - [S1] It contains **Transport** request header field with value is equal to "RTP/AVP/TCP" (transport=RTP, profile=AVP, lower-transport=TCP) (see [RFC 2326]) AND
  - [S2] It does not contain **Require** request header field with value is equal to "onvif-replay" AND
  - [S3] It is tunneled in HTTP AND
- Device response on the **RTSP SETUP** request fulfills the following requirements:

- [S4] It has RTSP 200 response code AND

- There is Client **RTSP DESCRIBE** request in Test Procedure fulfills the following requirements:
  - [S5] It invoked for the same Device as for the Client **RTSP SETUP** request AND
  - [S6] It invoked before the Client **RTSP SETUP** request AND
  - [S7] It is tunneled in HTTP AND
- Device response on the **RTSP DESCRIBE** request fulfills the following requirements:
  - [S8] SDP packet contains media type with Control URL that was used to send **RTSP SETUP** (see [RFC 2326, C.1.1 Control URL]) AND
  - [S9] It has RTSP 200 response code AND
- There is a Device **GetStreamUri** request in Test Procedure fulfills the following requirements:
  - [S10] It invoked for the same Device as for the Client **RTSP DESCRIBE** request AND
  - [S11] It invoked before the Client **RTSP DESCRIBE** request AND
  - [S12] **tr2:GetStreamUri/tr2:Protocol** element value is equal to "RtspOverHttp"
- Device response on the **GetStreamUri** request to Media2 Service fulfills the following requirements:
  - [S13] It has HTTP 200 response code AND
  - [S14] It contains **tr2: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 fulfills the following requirements:
  - [S15] It invoked for the same RTSP session as the Client **RTSP SETUP** request AND
  - [S16] It invoked after the Client **RTSP SETUP** request AND
  - [S17] RTSP address that was used to send it is correspond to any media Control URL or session Control URL or Content-Base URL from SDP packet (see [RFC 2326, C.1.1 Control URL]) AND
  - [S18] It does not contain **Require** request header field with value is equal to "onvif-replay" AND
  - [S19] It is tunneled in HTTP AND

- Device response on the **RTSP PLAY** request fulfills the following requirements:
  - [S20] It has RTSP 200 response code AND
- There is Client **RTSP TEARDOWN** request in Test Procedure fulfills the following requirements:
  - [S21] It invoked for the same RTSP session as the Client **RTSP SETUP** request AND
  - [S22] It invoked after the Client **RTSP PLAY** request AND
  - [S23] RTSP address that was used to send it is correspond to any media Control URL or session Control URL or Content-Base URL from SDP packet (see [RFC 2326, C.1.1 Control URL]) AND
  - [S24] It is tunneled in HTTP AND
- If there is Device response on the **RTSP TEARDOWN** request then it fulfills the following requirements:
  - [S25] It has RTSP 200 response code.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: media2_media_streaming.RTP_RTSP_HTTP

8 Video Streaming Using Media2 Test Cases

8.1 Feature Level Normative Reference:

Validated Feature: VideoStreaming

Profile T Requirement: Mandatory

8.2 Expected Scenarios Under Test:

1. Client connects to Device to setup and control of video streaming.
2. Client is considered as supporting Video Streaming if the following conditions are met:
   - Client supports media2_get_profiles.get_profiles feature AND
   - Client supports media2_get_stream_uri.get_stream_uri feature AND
   - Client is able to receive a stream using Media2 and decode H.264 video using the selected Media Profile over RTSP AND
   - Client is able to receive a stream using Media2 and decode H.265 video using the selected Media Profile over RTSP.
3. Client is considered as NOT supporting Video Streaming if ANY of the following is TRUE:
   - Client does not support media2_get_profiles.get_profiles feature OR
   - Client does not support media2_get_stream_uri.get_stream_uri feature OR
   - Client is unable to initiate and retrieve video stream using Media2 with H.264 encoding type OR
   - Client is unable to initiate and retrieve video stream using Media2 with H.265 encoding type.

8.3 H264 VIDEO STREAMING USING MEDIA2

Test Label: Video Streaming - H264

Test Case ID: MEDIA2_VIDEOSTREAMING-1

Profile S Normative Reference: Mandatory

Feature Under Test: Video Streaming

Test Purpose: To verify that the Client is able to initiate and retrieve a video stream with H264 encoding type.

Pre-Requisite:

- Device supports H264 encoding for Video Streaming for Media2 Service (Media2_H264).
- The Network Trace Capture files contains at least one Conversation between Client and Device with Video Streaming of H264 encoding type.
- The Network Trace Capture files contains at least one Conversation between Client and Device with **GetStreamUri** for Media2 Service.
- Device supports H264 encoding for Video Streaming using Media2.

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

1. Client invokes **GetStreamUri** request message for media profile that contains Video Source Configuration and Video Encoder Configuration with H264 Encoding value. GetStreamUri request is set for RtspUnicast OR RtspMulticast OR RTSP OR RtspOverHttp transport.
2. Device responds with code HTTP 200 OK and **GetStreamUriResponse** message.
3. Client invokes **RTSP DESCRIBE** request to retrieve media stream description.
4. Device responds with code RTSP 200 OK and SDP information with Media Type: "video" and with encoding name "H264".
5. Client invokes **RTSP SETUP** request without "onvif-replay" Require header and with transport parameter element to to set media session parameters for H264 video streaming.
6. Device responds with code RTSP 200 OK.
7. Client invokes **RTSP PLAY** request without "onvif-replay" Require header to start media stream.
8. Device responds with code RTSP 200 OK.
9. Client invokes **RTSP TEARDOWN** request to terminate the RTSP session.
10. If Device sends response to RTSP TEARDOWN, it has code RTSP 200 OK or RTSP 454.

Test Result:

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

PASS -

- There is Client **RTSP DESCRIBE** request in Test Procedure
- Device response on the **RTSP DESCRIBE** request fulfills the following requirements:
  - [S1] It has RTSP 200 response code AND
  - [S2] SDP packet contains media type "video" (m=video) with sessions attribute "rtpmap" with encoding name "H264" (see [RFC 3984], item 8.2.1. Mapping of MIME Parameters to SDP) AND
- There is Client **RTSP SETUP** request in Test Procedure fulfills the following requirements:
  - [S3] It invoked for the same Device as for the Client **RTSP DESCRIBE** request AND
  - [S4] It invoked after the Client **RTSP DESCRIBE** request AND
  - [S5] RTSP address that was used to send **RTSP SETUP** is correspond to corresponding media Control URL from SDP packet (see [RFC 2326, C.1.1 Control URL]) AND
  - [S6] It does not contain **Require** request header field with value is equal to "onvif-replay" AND
- Device response on the **RTSP SETUP** request fulfills the following requirements:
  - [S7] It has RTSP 200 response code AND
- There is a Device response on the **GetStreamUri** request for Media2 Service in Test Procedure fulfills the following requirements:
  - [S8] It has HTTP 200 response code AND
  - [S9] It received for the same Device as for the Client **RTSP DESCRIBE** request AND
  - [S10] It received before the Client **RTSP DESCRIBE** request AND
  - [S11] It contains **tr2: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 fulfills the following requirements:
  - [S12] It invoked for the same RTSP session as the Client **RTSP SETUP** request AND
  - [S13] It invoked after the Client **RTSP SETUP** request AND
  - [S14] RTSP address that was used to send it is correspond to corresponding media Control URL or session Control URL or Content-Base URL from SDP packet (see [RFC 2326, C.1.1 Control URL]) AND

- [S15] It does not contain **Require** request header field with value is equal to "onvif-replay" AND

- Device response on the **RTSP PLAY** request fulfills the following requirements:
  - [S16] It has RTSP 200 response code AND
- There is Client **RTSP TEARDOWN** request in Test Procedure fulfills the following requirements:
  - [S17] It invoked for the same RTSP session as the Client **RTSP SETUP** request AND
  - [S18] It invoked after the Client **RTSP PLAY** request AND
  - [S19] RTSP address that was used to send it is correspond to corresponding media Control URL or session Control URL or Content-Base URL from SDP packet (see [RFC 2326, C.1.1 Control URL]) AND
- If there is Device response on the **RTSP TEARDOWN** request then it fulfills the following requirements:
  - [S20] It has RTSP 200 response code.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: media2_video_streaming.h264

8.4 H265 VIDEO STREAMING USING MEDIA2

Test Label: Video Streaming - H265

Test Case ID: MEDIA2_VIDEOSTREAMING-2

Profile S Normative Reference: Mandatory

Feature Under Test: Video Streaming

Test Purpose: To verify that the Client is able to initiate and retrieve a video stream with H265 encoding type.

Pre-Requisite:

- Device supports H265 encoding for Video Streaming for Media2 Service (Media2_H265).

- The Network Trace Capture files contains at least one Conversation between Client and Device with Video Streaming of H265 encoding type.
- The Network Trace Capture files contains at least one Conversation between Client and Device with **GetStreamUri** for Media2 Service.
- Device supports H265 encoding for Video Streaming using Media2.

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

1. Client invokes **GetStreamUri** request message for media profile that contains Video Source Configuration and Video Encoder Configuration with H265 Encoding value. GetStreamUri request is set for RtspUnicast OR RtspMulticast OR RTSP OR RtspOverHttp transport.
2. Device responds with code HTTP 200 OK and **GetStreamUriResponse** message.
3. Client invokes **RTSP DESCRIBE** request to retrieve media stream description.
4. Device responds with code RTSP 200 OK and SDP information with Media Type: "video" and with encoding name "H265".
5. Client invokes **RTSP SETUP** request without "onvif-replay" Require header and with transport parameter element to to set media session parameters for H265 video streaming.
6. Device responds with code RTSP 200 OK.
7. Client invokes **RTSP PLAY** request without "onvif-replay" Require header to start media stream.
8. Device responds with code RTSP 200 OK.
9. Client invokes **RTSP TEARDOWN** request to terminate the RTSP session.
10. If Device sends response to RTSP TEARDOWN, it has code RTSP 200 OK or RTSP 454.

Test Result:

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

PASS -

- There is Client **RTSP DESCRIBE** request in Test Procedure
- Device response on the **RTSP DESCRIBE** request fulfills the following requirements:
  - [S1] It has RTSP 200 response code AND

- [S2] SDP packet contains media type "video" (m=video) with sessions attribute "rtpmap" with encoding name "H264" (see [RFC 3984], item 8.2.1. Mapping of MIME Parameters to SDP) AND

- There is Client **RTSP SETUP** request in Test Procedure fulfills the following requirements:
  - [S3] It invoked for the same Device as for the Client **RTSP DESCRIBE** request AND
  - [S4] It invoked after the Client **RTSP DESCRIBE** request AND
  - [S5] RTSP address that was used to send **RTSP SETUP** is correspond to corresponding media Control URL from SDP packet (see [RFC 2326, C.1.1 Control URL]) AND
  - [S6] It does not contain **Require** request header field with value is equal to "onvif-replay" AND
- Device response on the **RTSP SETUP** request fulfills the following requirements:
  - [S7] It has RTSP 200 response code AND
- There is a Device response on the **GetStreamUri** request to Media2 Service in Test Procedure fulfills the following requirements:
  - [S8] It has HTTP 200 response code AND
  - [S9] It received for the same Device as for the Client **RTSP DESCRIBE** request AND
  - [S10] It received before the Client **RTSP DESCRIBE** request AND
  - [S11] It contains **tr2: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 fulfills the following requirements:
  - [S12] It invoked for the same RTSP session as the Client **RTSP SETUP** request AND
  - [S13] It invoked after the Client **RTSP SETUP** request AND
  - [S14] RTSP address that was used to send it is correspond to corresponding media Control URL or session Control URL or Content-Base URL from SDP packet (see [RFC 2326, C.1.1 Control URL]) AND
  - [S15] It does not contain **Require** request header field with value is equal to "onvif-replay" AND
- Device response on the **RTSP PLAY** request fulfills the following requirements:

- [S16] It has RTSP 200 response code AND

- There is Client **RTSP TEARDOWN** request in Test Procedure fulfills the following requirements:
  - [S17] It invoked for the same RTSP session as the Client **RTSP SETUP** request AND
  - [S18] It invoked after the Client **RTSP PLAY** request AND
  - [S19] RTSP address that was used to send it is correspond to corresponding media Control URL or session Control URL or Content-Base URL from SDP packet (see [RFC 2326, C.1.1 Control URL]) AND
- If there is Device response on the **RTSP TEARDOWN** request then it fulfills the following requirements:
  - [S20] It has RTSP 200 response code.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: media2_video_streaming.h265

9 Video Encoder Configuration Using Media2 Test Cases

9.1 Feature Level Normative Reference:

Validated Feature: VideoEncoderConfiguration

Profile T Requirement: Mandatory

9.2 Expected Scenarios Under Test:

1. Client connects to Device to modify Video Encoder Configuration.
2. Client is considered as supporting Video Encoder Configuration if the following conditions are met:
   - Client is able to retrieve video encoder configuratoin options using **GetVideoEncoderConfigurationOptions** operation (Media2 Service) AND
   - Client is able to modify video encoder configuratoin using **SetVideoEncoderConfiguration** operation (Media2 Service) AND
3. Client is considered as NOT supporting Video Encoder Configuration if ANY of the following is TRUE:
   - No valid response to **GetVideoEncoderConfigurationOptions** request (Media2 Service) OR
   - No valid response to **SetVideoEncoderConfiguration** request (Media2 Service) OR

9.3 GET VIDEO ENCODER CONFIGURATION OPTIONS USING MEDIA2

Test Label: Video Encoder Configuration - Get Video Encoder Configuration Options

Test Case ID: MEDIA2_VIDEOENCODERCONFIGURATION-1

Profile T Normative Reference: Mandatory

Feature Under Test: Get Video Encoder Configuration Options

Test Purpose: To verify that video encoder configuration options provided by Device is received by Client using the **GetVideoEncoderConfigurationOptions** operation.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with **GetVideoEncoderConfigurationOptions** operation for Media2 Service present.
- Device supports Media2 Service (Media2Service).

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

1. Client invokes **GetVideoEncoderConfigurationOptions** request message to retrieve a video encoder configuration options from the Device.
2. Device responds with code HTTP 200 OK and **GetVideoEncoderConfigurationOptionsResponse** message.

Test Result:

PASS -

- Client **GetVideoEncoderConfigurationOptions** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **GetVideoEncoderConfigurationOptions** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **tr2:GetVideoEncoderConfigurationOptions** AND
- Device response on the **GetVideoEncoderConfigurations** request fulfills the following requirements:
  - [S2] It has HTTP 200 response code AND
  - [S3] **soapenv:Body** element has child element **tr2:GetVideoEncoderConfigurationOptionsResponse**.

FAIL -

- The Client failed PASS criteria.

Validated Feature List:
media2_video_encoder_configuration.get_video_encoder_configuration_options

9.4 SET VIDEO ENCODER CONFIGURATION USING MEDIA2

Test Label: Video Encoder Configuration - Set Video Encoder Configuration

Test Case ID: MEDIA2_VIDEOENCODERCONFIGURATION-2

Profile T Normative Reference: Mandatory

Feature Under Test: Set Video Encoder Configuration

Test Purpose: To verify that Client is able to change video encoder configuration provided by Device using the **SetVideoEncoderConfiguration** operation.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with **SetVideoEncoderConfiguration** operation for Media2 Service present.
- Device supports Media2 Service (Media2Service).

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

1. Client invokes **SetVideoEncoderConfiguration** request message to change a video encoder configuration on the Device.
2. Device responds with code HTTP 200 OK and **SetVideoEncoderConfigurationResponse** message.

Test Result:

PASS -

- Client **SetVideoEncoderConfiguration** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **SetVideoEncoderConfiguration** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **tr2:SetVideoEncoderConfiguration** AND
- Device response on the **SetVideoEncoderConfiguration** request fulfills the following requirements:
  - [S2] It has HTTP 200 response code AND
  - [S3] **soapenv:Body** element has child element **tr2:SetVideoEncoderConfigurationResponse**.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: media2_video_encoder_configuration.set_video_encoder_configuration

10 Audio Encoder Configuration Using Media2 Test Cases

10.1 Feature Level Normative Reference:

Validated Feature: AudioEncoderConfiguration

Profile T Requirement: Mandatory

10.2 Expected Scenarios Under Test:

1. Client connects to Device to modify Audio Encoder Configuration.
2. Client is considered as supporting Audio Encoder Configuration if the following conditions are met:
   - Client is able to retrieve audio encoder configuratoins using **GetAudioEncoderConfigurations** operation (Media2 Service) AND
   - Client is able to retrieve audio encoder configuratoin options using **GetAudioEncoderConfigurationOptions** operation (Media2 Service) AND
   - Client is able to modify audio encoder configuratoin using **SetAudioEncoderConfiguration** operation (Media2 Service) AND
3. Client is considered as NOT supporting Audio Encoder Configuration if ANY of the following is TRUE:
   - No valid response to **GetAudioEncoderConfigurations** request (Media2 Service) OR
   - No valid response to **GetAudioEncoderConfigurationOptions** request (Media2 Service) OR
   - No valid response to **SetAudioEncoderConfiguration** request (Media2 Service) OR

10.3 GET AUDIO ENCODER CONFIGURATIONS USING MEDIA2

Test Label: Audio Encoder Configuration - Get Audio Encoder Configurations

Test Case ID: MEDIA2_AUDIOENCODERCONFIGURATION-1

Profile T Normative Reference: Mandatory

Feature Under Test: GetAudioEncoderConfigurations

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

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with **GetAudioEncoderConfigurations** operation for Media2 Service present.
- Device supports Media2 Service (Media2Service).
- Device supports Audio feature for Media2 Service (Media2_Audio).

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

1. Client invokes **GetAudioEncoderConfigurations** request message to retrieve an audio encoder configuration or a list of audio encoder configurations from the Device.
2. Device responds with code HTTP 200 OK and **GetAudioEncoderConfigurationsResponse** message.

Test Result:

PASS -

- Client **GetAudioEncoderConfigurations** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **GetAudioEncoderConfigurations** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **tr2:GetAudioEncoderConfigurations** AND
- Device response on the **GetAudioEncoderConfigurations** request fulfills the following requirements:
  - [S2] It has HTTP 200 response code AND
  - [S3] **soapenv:Body** element has child element **tr2:GetAudioEncoderConfigurationsResponse**.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: media2_audio_encoder_configuration.get_audio_encoder_configurations

10.4 GET AUDIO ENCODER CONFIGURATION OPTIONS USING MEDIA2

Test Label: Audio Encoder Configuration - Get Audio Encoder Configuration Options

Test Case ID: MEDIA2_AUDIOENCODERCONFIGURATION-2

Profile T Normative Reference: Mandatory

Feature Under Test: GetAudioEncoderConfigurationOptions

Test Purpose: To verify that audio encoder configuration options provided by Device is received by Client using the **GetAudioEncoderConfigurationOptions** operation.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with **GetAudioEncoderConfigurationOptions** operation for Media2 Service present.
- Device supports Media2 Service (Media2Service).
- Device supports Audio feature for Media2 Service (Media2_Audio).

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

1. Client invokes **GetAudioEncoderConfigurationOptions** request message to retrieve an audio encoder configuration options from the Device.
2. Device responds with code HTTP 200 OK and **GetAudioEncoderConfigurationOptionsResponse** message.

Test Result:

PASS -

- Client **GetAudioEncoderConfigurationOptions** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **GetAudioEncoderConfigurationOptions** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **tr2:GetAudioEncoderConfigurationOptions** AND
- Device response on the **GetAudioEncoderConfigurations** request fulfills the following requirements:

- [S2] It has HTTP 200 response code AND
- [S3] **soapenv:Body** element has child element **tr2:GetAudioEncoderConfigurationOptionsResponse**.

FAIL -

- The Client failed PASS criteria.

Validated Feature List:
media2_audio_encoder_configuration.get_audio_encoder_configuration_options

10.5 SET AUDIO ENCODER CONFIGURATION USING MEDIA2

Test Label: Audio Encoder Configuration - Set Audio Encoder Configuration

Test Case ID: MEDIA2_AUDIOENCODERCONFIGURATION-3

Profile T Normative Reference: Mandatory

Feature Under Test: SetAudioEncoderConfiguration

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

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with **SetAudioEncoderConfiguration** operation for Media2 Service present.
- Device supports Media2 Service (Media2Service).
- Device supports Audio feature for Media2 Service (Media2_Audio).

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

1. Client invokes **SetAudioEncoderConfiguration** request message to change an audio encoder configuration on the Device.
2. Device responds with code HTTP 200 OK and **SetAudioEncoderConfigurationResponse** message.

Test Result:

PASS -

- Client **SetAudioEncoderConfiguration** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **SetAudioEncoderConfiguration** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **tr2:SetAudioEncoderConfiguration** AND
- Device response on the **SetAudioEncoderConfiguration** request fulfills the following requirements:
  - [S2] It has HTTP 200 response code AND
  - [S3] **soapenv:Body** element has child element **tr2:SetAudioEncoderConfigurationResponse**.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: media2_audio_encoder_configuration.set_audio_encoder_configuration

11 Audio Streaming Using Media2 Test Cases

11.1 Feature Level Requirement:

Validated Feature: AudioStreaming

Profile T Requirement: Mandatory

11.2 Expected Scenarios Under Test:

1. Client connects to Device to configure a media profile and initiate Audio Streaming with G.711 encoding type and AAC encoding type.
2. Client is considered as supporting Audio Streaming if the following conditions are met:
   - Client supports media2_get_profiles.get_profiles feature AND
   - Client supports media2_get_stream_uri.get_stream_uri feature AND
   - Client is able to initiate and retrieve audio stream using Media2 with G.711 encoding type AND
   - Client is able to initiate and retrieve audio stream using Media2 with AAC encoding type.
3. Client is considered as NOT supporting Audio Streaming if ANY of the following is TRUE:
   - Client does not support media2_get_profiles.get_profiles feature OR
   - Client does not support media2_get_stream_uri.get_stream_uri feature OR
   - Client is unable to initiate and retrieve audio stream using Media2 with G.711 encoding type OR
   - Client is unable to initiate and retrieve audio stream using Media2 with AAC encoding type.

11.3 G.711 AUDIO STREAMING USING MEDIA2

Test Label: Audio Streaming using Media2 - G.711

Test Case ID: MEDIA2_AUDIOSTREAMING-1

Profile T Normative Reference: Mandatory

Feature Under Test: AudioStreaming

Test Purpose: To verify that the Client is able to initiate and retrieve audio stream with G.711 encoding type.

Pre-Requisite:

- The Network Trace Capture files contains at least one conversation between Client and Device with Audio Streaming of G.711 encoding type.
- The Network Trace Capture files contains at least one Conversation between Client and Device with **GetStreamUri** for Media2 Service.
- Device supports G.711 encoding for Audio streaming using Media2 (Media2_G711).

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

1. Client invokes **GetStreamUri** request message for media profile that contains Audio Source Configuration and Audio Encoder Configuration with G711 Encoding value. GetStreamUri request is set for 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.
4. Device responds with code RTSP 200 OK and SDP information with Media Type: "audio" and with encoding name "PCMU" or with payload type number "0".
5. Client invokes **RTSP SETUP** request without "onvif-replay" Require header and with transport parameter element to set media session parameters for G711 audio streaming.
6. Device responds with code RTSP 200 OK.
7. Client invokes **RTSP PLAY** request without "onvif-replay" Require header to start media stream.
8. Device responds with code RTSP 200 OK.
9. Client invokes **RTSP TEARDOWN** request to terminate the RTSP session.
10. If Device sends response to RTSP TEARDOWN, it has code RTSP 200 OK or RTSP 454.

Test Result:

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

PASS -

- There is Client **RTSP DESCRIBE** request in Test Procedure
- Device response on the **RTSP DESCRIBE** request fulfills the following requirements:
  - [S1] It has RTSP 200 response code AND
  - [S2] IF SDP packet contains media type "audio" (m=audio) without session attribute "sendonly" (a=sendonly) AND with sessions attribute "rtpmap" THEN encoding name is "PCMU"
  - [S3] ELSE IF SDP packet contains media type "audio" (m=audio) without session attribute "sendonly" (a=sendonly) AND without sessions attribute "rtpmap" THEN payload type number is "0" (see [RFC 3551]) AND
- There is Client **RTSP SETUP** request in Test Procedure fulfills the following requirements:
  - [S4] It invoked for the same Device as for the Client **RTSP DESCRIBE** request AND
  - [S5] It invoked after the Client **RTSP DESCRIBE** request AND
  - [S6] RTSP address that was used to send **RTSP SETUP** is correspond to corresponding media Control URL from SDP packet (see [RFC 2326, C.1.1 Control URL]) AND
  - [S7] It does not contain **Require** request header field with value is equal to "onvif-replay" AND
- Device response on the **RTSP SETUP** request fulfills the following requirements:
  - [S8] It has RTSP 200 response code AND
- There is a Device response on the **GetStreamUri** request to Media2 Service in Test Procedure fulfills the following requirements:
  - [S9] It has HTTP 200 response code AND
  - [S10] It received for the same Device as for the Client **RTSP DESCRIBE** request AND
  - [S11] It received before the Client **RTSP DESCRIBE** request AND
  - [S12] It contains **tr2: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 fulfills the following requirements:
  - [S13] It invoked for the same Device as for the Client **RTSP SETUP** request AND
  - [S14] It invoked after the Client **RTSP SETUP** request AND

- [S15] RTSP address that was used to send it is correspond to corresponding media Control URL or session Control URL or Content-Base URL from SDP packet (see [RFC 2326, C.1.1 Control URL]) AND
- [S16] It does not contain **Require** request header field with value is equal to "onvif-replay" AND

- Device response on the **RTSP PLAY** request fulfills the following requirements:
  - [S17] It has RTSP 200 response code AND
- There is Client **RTSP TEARDOWN** request in Test Procedure fulfills the following requirements:
  - [S18] It invoked for the same Device as for the Client **RTSP SETUP** request AND
  - [S19] It invoked after the Client **RTSP PLAY** request AND
  - [S20] RTSP address that was used to send it is correspond to corresponding media Control URL or session Control URL or Content-Base URL from SDP packet (see [RFC 2326, C.1.1 Control URL]) AND
- If there is Device response on the **RTSP TEARDOWN** request then it fulfills the following requirements:
  - [S21] It has RTSP 200 response code.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: media2_audio_streaming.g711

11.4 AAC AUDIO STREAMING USING MEDIA2

Test Label: Audio Streaming using Media2 - AAC

Test Case ID: MEDIA2_AUDIOSTREAMING-2

Profile T Normative Reference: Mandatory

Feature Under Test: AudioStreaming

Test Purpose: To verify that the Client is able to initiate and retrieve audio stream with AAC encoding type.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with Audio Streaming of AAC encoding type.
- The Network Trace Capture files contains at least one Conversation between Client and Device with **GetStreamUri** for Media2 Service.
- Device supports AAC encoding for Audio streaming using Media2 (Media2_AAC).

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

1. Client invokes **GetStreamUri** request message for media profile that contains Audio Source Configuration and Audio Encoder Configuration with AAC Encoding value. GetStreamUri request is set for 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.
4. Device responds with code RTSP 200 OK and SDP information with Media Type: "audio" and with encoding name "MPEG4-GENERIC".
5. Client invokes **RTSP SETUP** request without "onvif-replay" Require header and with transport parameter element to to set media session parameters for AAC audio streaming.
6. Device responds with code RTSP 200 OK.
7. Client invokes **RTSP PLAY** request without "onvif-replay" Require header to start media stream.
8. Device responds with code RTSP 200 OK.
9. Client invokes **RTSP TEARDOWN** request to terminate the RTSP session.
10. If Device sends response to RTSP TEARDOWN, it has code RTSP 200 OK or RTSP 454.

Test Result:

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

PASS -

- There is Client **RTSP DESCRIBE** request in Test Procedure
- Device response on the **RTSP DESCRIBE** request fulfills the following requirements:
  - [S1] It has RTSP 200 response code AND

- [S2] SDP packet contains media type "audio" (m=audio) without session attribute "sendonly" (a=sendonly) AND with sessions attribute "rtpmap" with encoding name "MPEG4-GENERIC" (see [RFC 3640]) AND

- There is Client **RTSP SETUP** request in Test Procedure fulfills the following requirements:
  - [S3] It invoked for the same Device as for the Client **RTSP DESCRIBE** request AND
  - [S4] It invoked after the Client **RTSP DESCRIBE** request AND
  - [S5] RTSP address that was used to send **RTSP SETUP** is correspond to corresponding media Control URL from SDP packet (see [RFC 2326, C.1.1 Control URL]) AND
  - [S6] It does not contain **Require** request header field with value is equal to "onvif-replay" AND
- Device response on the **RTSP SETUP** request fulfills the following requirements:
  - [S7] It has RTSP 200 response code AND
- There is a Device response on the **GetStreamUri** request to Media2 Service in Test Procedure fulfills the following requirements:
  - [S8] It has HTTP 200 response code AND
  - [S9] It received for the same Device as for the Client **RTSP DESCRIBE** request AND
  - [S10] It received before the Client **RTSP DESCRIBE** request AND
  - [S11] It contains **tr2: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 fulfills the following requirements:
  - [S12] It invoked for the same Device as for the Client **RTSP SETUP** request AND
  - [S13] It invoked after the Client **RTSP SETUP** request AND
  - [S14] RTSP address that was used to send it is correspond to corresponding media Control URL or session Control URL or Content-Base URL from SDP packet (see [RFC 2326, C.1.1 Control URL]) AND
  - [S15] It does not contain **Require** request header field with value is equal to "onvif-replay" AND
- Device response on the **RTSP PLAY** request fulfills the following requirements:

- [S16] It has RTSP 200 response code AND

- There is Client **RTSP TEARDOWN** request in Test Procedure fulfills the following requirements:
  - [S17] It invoked for the same Device as for the Client **RTSP SETUP** request AND
  - [S18] It invoked after the Client **RTSP PLAY** request AND
  - [S19] RTSP address that was used to send it is correspond to corresponding media Control URL or session Control URL or Content-Base URL from SDP packet (see [RFC 2326, C.1.1 Control URL]) AND
- If there is Device response on the **RTSP TEARDOWN** request then it fulfills the following requirements:
  - [S20] It has RTSP 200 response code.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: media2_audio_streaming.aac

12 Audio Profile Configuration Using Media2 Test Cases

12.1 Feature Level Requirement:

Validated Feature: Media2AudioProfileConfiguration

Profile T Requirement: Mandatory

12.2 Expected Scenarios Under Test:

1. Client connects to Device to add compatible audio source configuration and audio encoder configuration to a Media Profile.
2. Client is considered as supporting Audio Profile Configuration if the following conditions are met:
   - Client is able to add an audio source configuration to profile using EITHER **AddConfiguration** operation with **Type** element value is equal to **AudioSource** OR **CreateProfile** operation with **Type** element value is equal to **AudioSource**.
   - Client is able to retrieve audio encoder configurations compatible with media profile using **GetAudioEncoderConfigurations** operation with specified ProfileToken element.
   - Client is able to add an audio encoder configuration using **AddConfiguration** operation with Type element value is equal to AudioEncoder.
3. Client is considered as NOT supporting Audio Profile Configuration if ANY of the following is TRUE:
   - Client unable to add an audio source configuration to profile using **AddConfiguration** operation and **CreateProfile** operation OR
   - No valid responses for **CreateProfile** request **Type** element value is equal to **AudioSource** if deteced OR
   - No valid responses for **GetAudioEncoderConfigurations** request with ProfileToken element OR
   - No valid responses for **AddConfiguration** request with Type element value is equal to AudioEncoder OR
   - No valid responses for **AddConfiguration** request with Type element value is equal to AudioSource is detected.

12.3 ADD AUDIO SOURCE CONFIGURATION USING MEDIA2

Test Label: Add Audio Source Configuration

Test Case ID: MEDIA2_AUDIOPROFILECONFIGURATION-1

Profile T Normative Reference: Mandatory

Feature Under Test: AddConfiguration

Test Purpose: To verify that Client is able to add an audio source configuration to a media profile using the **GetAudioSourceConfigurations** and **AddConfiguration** operation.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with **AddConfiguration** operation with Type = **AudioSource** present.
- Device supports Media2 Audio feature (Media2_Audio).

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

1. Client invokes **GetAudioSourceConfigurations** request message with specified **ProfileToken** to retrieve compatible audio source configurations for specified media profile from the Device.
2. Device responds with code HTTP 200 OK and **GetAudioSourceConfigurationsResponse** message.
3. Client invokes **AddConfiguration** request message with Type element value is equal to **AudioSource** and with Configuration token that was recieved in **GetAudioSourceConfigurationsResponse** message for the same media profile to add an audio source configuration to specified media profile on the Device.
4. Device responds with code HTTP 200 OK and **AddConfigurationResponse** message.

Test Result:

PASS -

- Client **AddConfiguration** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **AddConfiguration** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **tr2:AddConfiguration** AND

- [S2] It has **tr2:Configuration/tr2:Type** element with value is equal to "AudioSource" AND

- Device response to the **AddConfiguration** request fulfills the following requirements:
  - [S3] It has HTTP 200 response code AND
  - [S4] **soapenv:Body** element has child element **tr2:AddConfigurationResponse**.
- There is Client **GetAudioSourceConfigurations** request in Test Procedure that fulfills the following requirements:
  - [S5] It is invoked before the Client **AddConfiguration** request AND
  - [S6] It has **tr2:ProfileToken** element with value is equal to **tr2:ProfileToken** element value from the **AddConfiguration** request AND
  - [S7] It is the last **GetAudioSourceConfigurations** request which corresponds to [S5], AND [S6] AND
- Device response to the **GetAudioSourceConfigurations** request fulfills the following requirements:
  - [S8] It has HTTP 200 response code AND
  - [S9] **soapenv:Body** element has child element **tr2:GetAudioSourceConfigurationsResponse** AND
  - [S10] It contains **tr2:Configurations** element with **@token** attribute value equal to **tr2:Configuration/tr2:Token** value for Configuration with **tr2:Configuration/tr2:Type** value is equal to AudioSource from the **AddConfiguration** request message.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: Media2AudioProfileConfiguration.AddAudioSourceConfiguration

12.4 CREATE MEDIA PROFILE WITH AUDIO SOURCE CONFIGURATION USING MEDIA2

Test Label: Create Media2 Profile with Audio Source Configuration

Test Case ID: MEDIA2_AUDIOPROFILECONFIGURATION-2

Profile T Normative Reference: Conditional

Feature Under Test: CreateProfile

Test Purpose: To verify that Client is able to create media profile with audio source configuration using the **CreateProfile** operation.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with **CreateProfile** operation with Type = **AudioSource** present.
- Device supports Media2 Audio feature (Media2_Audio).

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

1. Client invokes **CreateProfile** request message with **tr2:Configuration\tr2:Type** = **AudioSource** and with specified **tr2:Token** element for this Configuration to create profile with audio source configuration on the Device.
2. Device responds with code HTTP 200 OK and **CreateProfileResponse** message.

Test Result:

PASS -

- Client **CreateProfile** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **CreateProfile** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **tr2:CreateProfile** AND
  - [S2] It has **tr2:Configuration/tr2:Type** element with value is equal to **"AudioSource"** AND
  - [S3] **tr2:Configuration** element with **tr2:Type** value is equal to **"AudioSource"** has **tr2:Token** element AND
- Device response to the **CreateProfile** request fulfills the following requirements:
  - [S3] It has HTTP 200 response code AND
  - [S4] **soapenv:Body** element has child element **tr2:CreateProfileResponse**.

FAIL -

- The Client failed PASS criteria.

Validated Feature List:
Media2AudioProfileConfiguration.CreateProfileWithAudioSourceConfiguration

12.5 GET AUDIO ENCODER CONFIGURATIONS COMPATIBLE WITH PROFILE USING MEDIA2

Test Label: Get Audio Encoder Configurations

Test Case ID: MEDIA2_AUDIOPROFILECONFIGURATION-3

Profile T Normative Reference: Mandatory

Feature Under Test: Media2_GetCompatibleAudioEncoderConfigurations

Test Purpose: To verify that list of audio encoder configurations compatible with a media profile is received by Client using the **GetAudioEncoderConfigurations** operation.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with **GetAudioEncoderConfigurations** operation with specified **ProfileToken** element present.
- Device supports Media2 Audio feature (Media2_Audio).

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

1. Client invokes **GetAudioEncoderConfigurations** request message with **ProfileToken** element to retrieve a list of audio encoder configurations compatible with requested media profile from the Device.
2. Device responds with code HTTP 200 OK and **GetAudioEncoderConfigurationsResponse** message.

Test Result:

PASS -

- Client **GetAudioEncoderConfigurations** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **GetAudioEncoderConfigurations** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **tr2:GetAudioEncoderConfigurations** AND
  - [S2] It has **tr2:ProfileToken** element AND

- Device response to the **GetAudioEncoderConfigurations** request fulfills the following requirements:
  - [S3] It has HTTP 200 response code AND
  - [S4] **soapenv:Body** element has child element **tr2:GetAudioEncoderConfigurationsResponse**.

FAIL -

- The Client failed PASS criteria.

Validated Feature List:
Media2AudioProfileConfiguration.Media2_GetCompatibleAudioEncoderConfigurations

12.6 ADD AUDIO ENCODER CONFIGURATION USING MEDIA2

Test Label: Add Audio Encoder Configuration

Test Case ID: MEDIA2_AUDIOPROFILECONFIGURATION-4

Profile T Normative Reference: Mandatory

Feature Under Test: AddConfiguration

Test Purpose: To verify that Client is able to add an audio encoder configuration to a media profile using the **AddConfiguration** operation.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with **AddConfiguration** operation with Type value is equal to "VideoEncoder" present.
- Device supports Media2 Audio feature (Media2_Audio).

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

1. Client invokes **GetAudioEncoderConfigurations** request message with specified **ProfileToken** to retrieve compatible audio encoder configurations for specified media profile from the Device.
2. Device responds with code HTTP 200 OK and **GetAudioEncoderConfigurationsResponse** message.
3. Client invokes **AddConfiguration** request message with Type element value is equal to AudioEncoder and with Configuration token that was recieved in

GetAudioEncoderConfigurationsResponse message for the same media profile to add an audio encoder configuration to specified media profile on the Device.

4. Device responds with code HTTP 200 OK and **AddConfigurationResponse** message.

Test Result:

PASS -

- Client **AddConfiguration** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **AddConfiguration** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **tr2:AddConfiguration** AND
  - [S2] It has **tr2:Configuration/tr2:Type** element with value is equal to "AudioEncoder" AND
- Device response to the **AddConfiguration** request fulfills the following requirements:
  - [S3] It has HTTP 200 response code AND
  - [S4] **soapenv:Body** element has child element **tr2:AddConfigurationResponse**.
- There is Client **GetAudioEncoderConfigurations** request in Test Procedure that fulfills the following requirements:
  - [S5] It is invoked before the Client **AddConfiguration** request AND
  - [S6] It has **tr2:ProfileToken** element with value is equal to **tr2:ProfileToken** element value from the **AddConfiguration** request AND
  - [S7] It is the last **GetAudioEncoderConfigurations** request which corresponds to [S5], AND [S6] AND
- Device response to the **GetAudioEncoderConfigurations** request fulfills the following requirements:
  - [S8] It has HTTP 200 response code AND
  - [S9] **soapenv:Body** element has child element **tr2:GetAudioEncoderConfigurationsResponse** AND
  - [S10] It contains **tr2:Configurations** element with **@token** attribute value equal to **tr2:Configuration/tr2:Token** value for Configuration with **tr2:Configuration/tr2:Type** value is equal to AudioEncoder from the **AddConfiguration** request message.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: Media2AudioProfileConfiguration.AddAudioEncoderConfiguration

13 Audio Backchannel Streaming Using Media2 Test Cases

13.1 Feature Level Requirement:

Validated Feature: AudioBackchannelStreaming

Profile A Requirement: None

Profile C Requirement: None

Profile G Requirement: None

Profile Q Requirement: None

Profile S Requirement: None

Profile T Requirement: Mandatory

13.2 Expected Scenarios Under Test:

1. Client connects to Device to stream audio for backchannel using Media2.
2. Client is considered as supporting Audio Backchannel Streaming if the following conditions are met:
   - Client is able to stream audio for backchannel using **G.711** AND
   - Client is able to stream audio for backchannel using **AAC** if supported.
3. Client is considered as NOT supporting Audio Backchannel Streaming if ANY of the following is TRUE:
   - No G.711 Audio Backchannel Streaming attempts were found OR
   - G.711 Audio Backchannel Streaming attempts have failed OR
   - Detected AAC Audio Backchannel Streaming attempts have failed.

13.3 G.711 AUDIO BACKCHANNEL STREAMING USING MEDIA2

Test Label: Audio Backchannel Streaming Using Media2 - G.711

Test Case ID: MEDIA2_AUDIOBACKCHANNELSTREAMING-1

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

Profile T Normative Reference: Mandatory

Feature Under Test: Audio Backchannel Streaming

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

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with audio backchannel streaming using Media2 with G.711 encoding.
- The Network Trace Capture files contains at least one Conversation between Client and Device with **GetStreamUri** for Media2 Service.
- Device supports G.711 encoding for Media2 Audio Outputs (Media2_AudioOutputG711).

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

1. Client invokes **GetStreamUri** request message to Media2 Service 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 or RTSP 454.

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

Test Result:

PASS -

- Client **RTSP DESCRIBE** request in Test Procedure fulfills the following requirements:
  - [S1] **Require** tag in RTSP header contains "www.onvif.org/ver20/backchannel" AND
- Device response to the **RTSP DESCRIBE** request fulfills the following requirements:
  - [S3] It has RTSP 200 response code AND
  - [S4] SDP packet contains media type "audio" (m=audio) with session attribute "sendonly" (a=sendonly) and sessions attribute "rtpmap" with encoding name "PCMU" AND
- There is 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 to Media2 Service 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 **tr2: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.

Validated Feature List: media2_audio_backchannel_streaming.g711

13.4 AAC AUDIO BACKCHANNEL STREAMING USING MEDIA2

Test Label: Audio Backchannel Streaming - AAC

Test Case ID: MEDIA2_AUDIOBACKCHANNELSTREAMING-2

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

Profile T Normative Reference: Conditional

Feature Under Test: Audio Backchannel Streaming

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

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with audio backchannel streaming using Media2 with AAC encoding.
- The Network Trace Capture files contains at least one Conversation between Client and Device with **GetStreamUri** for Media2 Service.
- Device supports AAC encoding for Media2 Audio Outputs (Media2_AudioOutputAAC).

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

1. Client invokes **GetStreamUri** request message to Media2 Service 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 or RTSP 454.

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

Test Result:

PASS -

- Client **RTSP DESCRIBE** request in Test Procedure fulfills the following requirements:
  - [S1] **Require** tag in RTSP header contains "www.onvif.org/ver20/backchannel" AND
- Device response to the **RTSP DESCRIBE** request fulfills the following requirements:
  - [S3] It has RTSP 200 response code AND
  - [S4] SDP packet contains media type "audio" (m=audio) with session attribute "sendonly" (a=sendonly) and sessions attribute "rtpmap" with encoding name "mpeg4-generic" AND
- There is 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 to Media2 Service 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 **tr2: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.

Validated Feature List: media2_audio_backchannel_streaming.aac

14 Audio Output Profile Configuration Using Media2 Test Cases

14.1 Feature Level Requirement:

Validated Feature: Media2AudioOutputProfileConfiguration

Profile T Requirement: Mandatory

14.2 Expected Scenarios Under Test:

1. Client connects to Device to add compatible audio output configuration and decoder configuration to a Media Profile.
2. Client is considered as supporting Audio Output Profile Configuration if the following conditions are met:
   - Client is able to add an audio output configuration to profile using EITHER **AddConfiguration** operation with **Type** element value is equal to **AudioOutput** OR **CreateProfile** operation with **Type** element value is equal to **AudioOutput**.
   - Client is able to retrieve audio decoder configurations compatible with media profile using **GetAudioDecoderConfigurations** operation with specified ProfileToken element.
   - Client is able to add an audio decoder configurations using **AddConfiguration** operation with Type element value is equal to AudioDecoder.
3. Client is considered as NOT supporting Audio Output Profile Configuration if ANY of the following is TRUE:
   - Client unable to add an audio output configuration to profile using **AddConfiguration** operation and **CreateProfile** operation OR
   - No valid responses for **CreateProfile** request **Type** element value is equal to **AudioOutput** if deteced OR
   - No valid responses for **GetAudioDecoderConfigurations** request with ProfileToken element OR
   - No valid responses for **AddConfiguration** request with Type element value is equal to AudioDecoder OR
   - No valid responses for **AddConfiguration** request with Type element value is equal to AudioOutput is detected.

14.3 ADD AUDIO OUTPUT CONFIGURATION USING MEDIA2

Test Label: Add Audio Output Configuration

Test Case ID: MEDIA2_AUDIOOUTPUTPROFILECONFIGURATION-1

Profile T Normative Reference: Mandatory

Feature Under Test: AddConfiguration

Test Purpose: To verify that Client is able to add an audio output configuration to a media profile using the **GetAudioOutputConfigurations** and **AddConfiguration** operation.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with **AddConfiguration** operation with Type = **AudioOutput** present.
- Device supports Media2 Audio Output feature (Media2_AudioOutput).

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

1. Client invokes **GetAudioOutputConfigurations** request message with specified **ProfileToken** to retrieve compatible audio output configurations for specified media profile from the Device.
2. Device responds with code HTTP 200 OK and **GetAudioOutputConfigurationsResponse** message.
3. Client invokes **AddConfiguration** request message with Type element value is equal to **AudioOutput** and with Configuration token that was recieved in **GetAudioOutputConfigurationsResponse** message for the same media profile to add an audio output configuration to specified media profile on the Device.
4. Device responds with code HTTP 200 OK and **AddConfigurationResponse** message.

Test Result:

PASS -

- Client **AddConfiguration** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **AddConfiguration** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **tr2:AddConfiguration** AND

- [S2] It has **tr2:Configuration/tr2:Type** element with value is equal to "AudioOutput" AND

- Device response to the **AddConfiguration** request fulfills the following requirements:
  - [S3] It has HTTP 200 response code AND
  - [S4] **soapenv:Body** element has child element **tr2:AddConfigurationResponse**.
- There is Client **GetAudioOutputConfigurations** request in Test Procedure that fulfills the following requirements:
  - [S5] It is invoked before the Client **AddConfiguration** request AND
  - [S6] It has **tr2:ProfileToken** element with value is equal to **tr2:ProfileToken** element value from the **AddConfiguration** request AND
  - [S7] It is the last **GetAudioOutputConfigurations** request which corresponds to [S5], AND [S6] AND
- Device response to the **GetAudioOutputConfigurations** request fulfills the following requirements:
  - [S8] It has HTTP 200 response code AND
  - [S9] **soapenv:Body** element has child element **tr2:GetAudioOutputConfigurationsResponse** AND
  - [S10] It contains **tr2:Configurations** element with **@token** attribute value equal to **tr2:Configuration/tr2:Token** value for Configuration with **tr2:Configuration/tr2:Type** value is equal to AudioOutput from the **AddConfiguration** request message.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: Media2AudioProfileConfiguration.AddAudioOutputConfiguration

14.4 CREATE MEDIA PROFILE WITH AUDIO OUTPUT CONFIGURATION USING MEDIA2

Test Label: Create Media2 Profile with Audio Output Configuration

Test Case ID: MEDIA2_AUDIOOUTPUTPROFILECONFIGURATION-2

Profile T Normative Reference: Conditional

Feature Under Test: CreateProfile

Test Purpose: To verify that Client is able to create media profile with audio output configuration using the **CreateProfile** operation.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with **CreateProfile** operation with Type = **AudioOutput** present.
- Device supports Media2 Audio Output feature (Media2_AudioOutput).

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

1. Client invokes **CreateProfile** request message with **tr2:Configuration\tr2:Type** = **AudioOutput** and with specified **tr2:Token** element for this Configuration to create profile with audio output configuration on the Device.
2. Device responds with code HTTP 200 OK and **CreateProfileResponse** message.

Test Result:

PASS -

- Client **CreateProfile** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **CreateProfile** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **tr2:CreateProfile** AND
  - [S2] It has **tr2:Configuration/tr2:Type** element with value is equal to **"AudioOutput"** AND
  - [S3] **tr2:Configuration** element with **tr2:Type** value is equal to **"AudioOutput"** has **tr2:Token** element AND
- Device response to the **CreateProfile** request fulfills the following requirements:
  - [S3] It has HTTP 200 response code AND
  - [S4] **soapenv:Body** element has child element **tr2:CreateProfileResponse**.

FAIL -

- The Client failed PASS criteria.

Validated Feature List:
Media2AudioProfileConfiguration.CreateProfileWithAudioOutputConfiguration

14.5 GET AUDIO DECODER CONFIGURATIONS COMPATIBLE WITH PROFILE USING MEDIA2

Test Label: Get Audio Decoder Configurations

Test Case ID: MEDIA2_AUDIOOUTPUTPROFILECONFIGURATION-3

Profile T Normative Reference: Mandatory

Feature Under Test: GetAudioDecoderConfigurationsUsingMedia2

Test Purpose: To verify that list of audio decoder configurations compatible with a media profile 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 Media2 Audio Output feature (Media2_AudioOutput).

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

1. Client invokes **GetAudioDecoderConfigurations** request message with **ProfileToken** element to retrieve a list of audio decoder configurations compatible with requested media profile 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 **tr2:GetAudioDecoderConfigurations** AND
  - [S2] It has **tr2:ProfileToken** element AND
- Device response to the **GetAudioDecoderConfigurations** request fulfills the following requirements:

- [S3] It has HTTP 200 response code AND
- [S4] **soapenv:Body** element has child element **tr2:GetAudioDecoderConfigurationsResponse**.

FAIL -

- The Client failed PASS criteria.

Validated Feature List:
Media2AudioOutputProfileConfiguration.GetCompatibleAudioDecoderConfigurations

14.6 ADD AUDIO DECODER CONFIGURATION USING MEDIA2

Test Label: Add Audio Decoder Configuration

Test Case ID: MEDIA2_AUDIOOUTPUTPROFILECONFIGURATION-4

Profile T Normative Reference: Mandatory

Feature Under Test: AddConfiguration

Test Purpose: To verify that Client is able to add an audio decoder configuration to a media profile using the **AddConfiguration** operation.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with **AddConfiguration** operation present.
- Device supports Media2 Audio Output feature (Media2_AudioOutput).

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

1. Client invokes **GetAudioDecoderConfigurations** request message with specified **ProfileToken** to retrieve compatible audio decoder configurations for specified media profile from the Device.
2. Device responds with code HTTP 200 OK and **GetAudioDecoderConfigurationsResponse** message.
3. Client invokes **AddConfiguration** request message with Type element value is equal to AudioDecoder and with Configuration token that was recieved in

GetAudioDecoderConfigurationsResponse message for the same media profile to add an audio decoder configuration to specified media profile on the Device.

4. Device responds with code HTTP 200 OK and **AddConfigurationResponse** message.

Test Result:

PASS -

- Client **AddConfiguration** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **AddConfiguration** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **tr2:AddConfiguration** AND
  - [S2] It has **tr2:Configuration/tr2:Type** element with value is equal to "AudioDecoder" AND
- Device response to the **AddConfiguration** request fulfills the following requirements:
  - [S3] It has HTTP 200 response code AND
  - [S4] **soapenv:Body** element has child element **tr2:AddConfigurationResponse**.
- There is Client **GetAudioDecoderConfigurations** request in Test Procedure that fulfills the following requirements:
  - [S5] It is invoked before the Client **AddConfiguration** request AND
  - [S6] It has **tr2:ProfileToken** element with value is equal to **tr2:ProfileToken** element value from the **AddConfiguration** request AND
  - [S7] It is the last **GetAudioDecoderConfigurations** request which corresponds to [S5], AND [S6] AND
- Device response to the **GetAudioDecoderConfigurations** request fulfills the following requirements:
  - [S8] It has HTTP 200 response code AND
  - [S9] **soapenv:Body** element has child element **tr2:GetAudioDecoderConfigurationsResponse** AND
  - [S10] It contains **tr2:Configurations** element with **@token** attribute value equal to **tr2:Configuration/tr2:Token** value for Configuration with **tr2:Configuration/tr2:Type** value is equal to AudioDecoder from the **AddConfiguration** request message.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: Media2AudioOutputProfileConfiguration.AddAudioDecoderConfiguration

15 Analytics Profile Configuration Using Media2 Test Cases

15.1 Feature Level Requirement:

Validated Feature: Media2AnalyticsProfileConfiguration

Profile T Requirement: Conditional

15.2 Expected Scenarios Under Test:

1. Client connects to Device to add compatible analytics configuration to a Media Profile.
2. Client is considered as supporting Analytics Profile Configuration if the following conditions are met:
   - Client is able to retrieve analytics configurations compatible with media profile using **GetAnalyticsConfigurations** operation with specified ProfileToken element.
   - Client is able to add an analytics configuration using **AddConfiguration** operation with Type element value is equal to **Analytics**.
3. Client is considered as NOT supporting Analytics Profile Configuration if ANY of the following is TRUE:
   - No valid responses for **GetAnalyticsConfigurations** request with ProfileToken element OR
   - No valid responses for **AddConfiguration** request with Type element value is equal to **Analytics**.

15.3 GET ANALYTICS CONFIGURATIONS COMPATIBLE WITH PROFILE USING MEDIA2

Test Label: Get Analytics Configurations

Test Case ID: MEDIA2_ANALYTICSPROFILECONFIGURATION-1

Profile T Normative Reference: Conditional

Feature Under Test: Media2_GetCompatibleAnalyticsConfigurations

Test Purpose: To verify that list of analytics configurations compatible with a media profile is received by Client using the **GetAnalyticsConfigurations** operation.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with **GetAnalyticsConfigurations** operation with specified **ProfileToken** element present.
- Device supports Media2 Analytics feature (Media2_Analytics).

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

1. Client invokes **GetAnalyticsConfigurations** request message with **ProfileToken** element to retrieve a list of analytics configurations compatible with requested media profile from the Device.
2. Device responds with code HTTP 200 OK and **GetAnalyticsConfigurationsResponse** message.

Test Result:

PASS -

- Client **GetAnalyticsConfigurations** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **GetAnalyticsConfigurations** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **tr2:GetAnalyticsConfigurations** AND
  - [S2] It has **tr2:ProfileToken** element AND
- Device response to the **GetAnalyticsConfigurations** request fulfills the following requirements:
  - [S3] It has HTTP 200 response code AND
  - [S4] **soapenv:Body** element has child element **tr2:GetAnalyticsConfigurationsResponse**.

FAIL -

- The Client failed PASS criteria.

Validated Feature List:
Media2AnalyticsProfileConfiguration.Media2_GetCompatibleAnalyticsConfigurations

15.4 ADD ANALYTICS CONFIGURATION USING MEDIA2

Test Label: Add Analytics Configuration

Test Case ID: MEDIA2_ANALYTICSPROFILECONFIGURATION-2

Profile T Normative Reference: Conditional

Feature Under Test: AddConfiguration

Test Purpose: To verify that Client is able to add an analytics configuration to a media profile using the **GetAnalyticsConfigurations** and **AddConfiguration** operation.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with **AddConfiguration** operation with Type = **Analytics** present.
- Device supports Media2 Analytics feature (Media2_Analytics).

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

1. Client invokes **GetAnalyticsConfigurations** request message with specified **ProfileToken** to retrieve compatible analytics configurations for specified media profile from the Device.
2. Device responds with code HTTP 200 OK and **GetAnalyticsConfigurationsResponse** message.
3. Client invokes **AddConfiguration** request message with Type element value is equal to **Analytics** and with Configuration token that was recieved in **GetAnalyticsConfigurationsResponse** message for the same media profile to add an analytics configuration to specified media profile on the Device.
4. Device responds with code HTTP 200 OK and **AddConfigurationResponse** message.

Test Result:

PASS -

- Client **AddConfiguration** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **AddConfiguration** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **tr2:AddConfiguration** AND
  - [S2] It has **tr2:Configuration/tr2:Type** element with value is equal to **Analytics** AND

- Device response to the **AddConfiguration** request fulfills the following requirements:
  - [S3] It has HTTP 200 response code AND
  - [S4] **soapenv:Body** element has child element **tr2:AddConfigurationResponse**.
- There is Client **GetAnalyticsConfigurations** request in Test Procedure that fulfills the following requirements:
  - [S5] It is invoked before the Client **AddConfiguration** request AND
  - [S6] It has **tr2:ProfileToken** element with value is equal to **tr2:ProfileToken** element value from the **AddConfiguration** request AND
  - [S7] It is the last **GetAnalyticsConfigurations** request which corresponds to [S5], AND [S6] AND
- Device response to the **GetAnalyticsConfigurations** request fulfills the following requirements:
  - [S8] It has HTTP 200 response code AND
  - [S9] **soapenv:Body** element has child element **tr2:GetAnalyticsConfigurationsResponse** AND
  - [S10] It contains **tr2:Configurations** element with **@token** attribute value equal to **tr2:Configuration/tr2:Token** value for Configuration with **tr2:Configuration/tr2:Type** value is equal to **Analytics** from the **AddConfiguration** request message.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: Media2AnalyticsProfileConfiguration.AddAnalyticsConfiguration

16 Video Source Mode Test Cases

16.1 Feature Level Normative Reference:

Validated Feature: VideoSourceMode

Profile T Requirement: Conditional

16.2 Expected Scenarios Under Test:

1. Client connects to Device to request the information for current video source mode and settable video source modes of specified video source using **GetVideoSourceModes** operation.
2. Client changes current video source mode using the **SetVideoSourceMode** operation.
3. Client is considered as supporting Video Source Mode if the following conditions are met:
   - Client is able to retrieve current video source mode and settable video source modes using **GetVideoSourceModes** operation AND
   - Client is able to modify source mode using **SetVideoSourceMode** operation.
4. Client is considered as NOT supporting Video Source Mode if ANY of the following is TRUE:
   - No valid response to **GetVideoSourceModes** request OR
   - No valid response to **SetVideoSourceMode** request.

16.3 GET VIDEO SOURCE MODES

Test Label: Get Video Source Modes

Test Case ID: VIDEOSOURCEMODE-1

Profile T Normative Reference: Conditional

Feature Under Test: GetVideoSourceModes

Test Purpose: To verify that video source modes provided by Device is received by Client using the **GetVideoSourceModes** operation.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with **GetVideoSourceModes** operation for Media2 Service present.
- Device supports Media2 Service (Media2Service).
- Device supports Video Source Mode feature (Media2_VideoSourceMode).

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

1. Client invokes **GetVideoSourceModes** request message to retrieve a video source modes from the Device.
2. Device responds with code HTTP 200 OK and **GetVideoSourceModesResponse** message.

Test Result:

PASS -

- Client **GetVideoSourceModes** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **GetVideoSourceModes** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **tr2:GetVideoSourceModes** AND
- Device response on the **GetVideoSourceModes** request fulfills the following requirements:
  - [S2] It has HTTP 200 response code AND
  - [S3] **soapenv:Body** element has child element **tr2:GetVideoSourceModesResponse**.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: video_source_mode.get_video_source_modes

16.4 SET VIDEO SOURCE MODE

Test Label: Set Video Source Mode

Test Case ID: VIDEOSOURCEMODE-2

Profile T Normative Reference: Conditional

Feature Under Test: SetVideoSourceMode

Test Purpose: To verify that current video source mode can be changed by Client using the **SetVideoSourceMode** operation.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with **SetVideoSourceMode** operation for Media2 Service present.
- Device supports Media2 Service (Media2Service).
- Device supports Video Source Mode feature (Media2_VideoSourceMode).

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

1. Client invokes **SetVideoSourceMode** request message to change a video source mode on the Device.
2. Device responds with code HTTP 200 OK and **SetVideoSourceModeResponse** message.

Test Result:

PASS -

- Client **SetVideoSourceMode** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **SetVideoSourceMode** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **tr2:SetVideoSourceMode** AND
- Device response on the **SetVideoSourceMode** request fulfills the following requirements:
  - [S2] It has HTTP 200 response code AND
  - [S3] **soapenv:Body** element has child element **tr2:SetVideoSourceModeResponse**.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: video_source_mode.set_video_source_mode

17 Audio Source Configuration Using Media2 Test Cases

17.1 Feature Level Normative Reference:

Validated Feature: AudioSourceConfiguration

Profile T Requirement: Conditional

17.2 Expected Scenarios Under Test:

1. Client connects to Device to modify Audio Source Configuration.
2. Client is considered as supporting Audio Source Configuration if the following conditions are met:
   - Client is able to retrieve audio source configuratoins using **GetAudioSourceConfigurations** operation (Media2 Service) AND
   - Client is able to retrieve audio source configuratoin options using **GetAudioSourceConfigurationOptions** operation (Media2 Service) AND
   - Client is able to modify audio source configuratoin using **SetAudioSourceConfiguration** operation (Media2 Service) AND
3. Client is considered as NOT supporting Audio Source Configuration if ANY of the following is TRUE:
   - No valid response to **GetAudioSourceConfigurations** request (Media2 Service) OR
   - No valid response to **GetAudioSourceConfigurationOptions** request (Media2 Service) OR
   - No valid response to **SetAudioSourceConfiguration** request (Media2 Service) OR

17.3 GET AUDIO SOURCE CONFIGURATIONS USING MEDIA2

Test Label: Audio Source Configuration - Get Audio Source Configurations

Test Case ID: MEDIA2_AUDIOSOURCECONFIGURATION-1

Profile T Normative Reference: Conditional

Feature Under Test: GetAudioSourceConfigurations

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

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with **GetAudioSourceConfigurations** operation for Media2 Service present.
- Device supports Media2 Service (Media2Service).
- Device supports Audio feature for Media2 Service (Media2_Audio).

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

1. Client invokes **GetAudioSourceConfigurations** request message to retrieve an audio source configuration or a list of audio source configurations from the Device.
2. Device responds with code HTTP 200 OK and **GetAudioSourceConfigurationsResponse** message.

Test Result:

PASS -

- Client **GetAudioSourceConfigurations** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **GetAudioSourceConfigurations** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **tr2:GetAudioSourceConfigurations** AND
- Device response on the **GetAudioSourceConfigurations** request fulfills the following requirements:
  - [S2] It has HTTP 200 response code AND
  - [S3] **soapenv:Body** element has child element **tr2:GetAudioSourceConfigurationsResponse**.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: media2_audio_source_configuration.get_audio_source_configurations

17.4 GET AUDIO SOURCE CONFIGURATION OPTIONS USING MEDIA2

Test Label: Audio Source Configuration - Get Audio Source Configuration Options

Test Case ID: MEDIA2_AUDIOSOURCECONFIGURATION-2

Profile T Normative Reference: Conditional

Feature Under Test: GetAudioSourceConfigurationOptions

Test Purpose: To verify that audio source configuration options provided by Device is received by Client using the **GetAudioSourceConfigurationOptions** operation.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with **GetAudioSourceConfigurationOptions** operation for Media2 Service present.
- Device supports Media2 Service (Media2Service).
- Device supports Audio feature for Media2 Service (Media2_Audio).

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

1. Client invokes **GetAudioSourceConfigurationOptions** request message to retrieve an audio source configuration options from the Device.
2. Device responds with code HTTP 200 OK and **GetAudioSourceConfigurationOptionsResponse** message.

Test Result:

PASS -

- Client **GetAudioSourceConfigurationOptions** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **GetAudioSourceConfigurationOptions** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **tr2:GetAudioSourceConfigurationOptions** AND
- Device response on the **GetAudioSourceConfigurations** request fulfills the following requirements:

- [S2] It has HTTP 200 response code AND
- [S3] **soapenv:Body** element has child element **tr2:GetAudioSourceConfigurationOptionsResponse**.

FAIL -

- The Client failed PASS criteria.

Validated Feature List:
media2_audio_source_configuration.get_audio_source_configuration_options

17.5 SET AUDIO SOURCE CONFIGURATION USING MEDIA2

Test Label: Audio Source Configuration - Set Audio Source Configuration

Test Case ID: MEDIA2_AUDIOSOURCECONFIGURATION-3

Profile T Normative Reference: Conditional

Feature Under Test: SetAudioSourceConfiguration

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

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with **SetAudioSourceConfiguration** operation for Media2 Service present.
- Device supports Media2 Service (Media2Service).
- Device supports Audio feature for Media2 Service (Media2_Audio).

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

1. Client invokes **SetAudioSourceConfiguration** request message to change an audio source configuration on the Device.
2. Device responds with code HTTP 200 OK and **SetAudioSourceConfigurationResponse** message.

Test Result:

PASS -

- Client **SetAudioSourceConfiguration** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **SetAudioSourceConfiguration** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **tr2:SetAudioSourceConfiguration** AND
- Device response on the **SetAudioSourceConfiguration** request fulfills the following requirements:
  - [S2] It has HTTP 200 response code AND
  - [S3] **soapenv:Body** element has child element **tr2:SetAudioSourceConfigurationResponse**.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: media2_audio_source_configuration.set_audio_source_configuration

18 Audio Output Configuration Using Media2 Test Cases

18.1 Feature Level Normative Reference:

Validated Feature: AudioOutputConfiguration

Profile T Requirement: Conditional

18.2 Expected Scenarios Under Test:

1. Client connects to Device to modify Audio Output Configuration.
2. Client is considered as supporting Audio Output Configuration if the following conditions are met:
   - Client is able to retrieve audio output configuratoins using **GetAudioOutputConfigurations** operation (Media2 Service) AND
   - Client is able to retrieve audio output configuratoin options using **GetAudioOutputConfigurationOptions** operation (Media2 Service) AND
   - Client is able to modify audio output configuratoin using **SetAudioOutputConfiguration** operation (Media2 Service) AND
3. Client is considered as NOT supporting Audio Output Configuration if ANY of the following is TRUE:
   - No valid response to **GetAudioOutputConfigurations** request (Media2 Service) OR
   - No valid response to **GetAudioOutputConfigurationOptions** request (Media2 Service) OR
   - No valid response to **SetAudioOutputConfiguration** request (Media2 Service) OR

18.3 GET AUDIO OUTPUT CONFIGURATIONS USING MEDIA2

Test Label: Audio Output Configuration - Get Audio Output Configurations

Test Case ID: MEDIA2_AUDIOOUTPUTCONFIGURATION-1

Profile T Normative Reference: Conditional

Feature Under Test: GetAudioOutputConfigurations

Test Purpose: To verify that audio output configuration 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 for Media2 Service present.
- Device supports Media2 Service (Media2Service).
- Device supports Audio outputs feature for Media2 Service (Media2_AudioOutput).

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

1. Client invokes **GetAudioOutputConfigurations** request message to retrieve an audio output configuration or a list of 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 **tr2:GetAudioOutputConfigurations** AND
- Device response on the **GetAudioOutputConfigurations** request fulfills the following requirements:
  - [S2] It has HTTP 200 response code AND
  - [S3] **soapenv:Body** element has child element **tr2:GetAudioOutputConfigurationsResponse**.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: media2_audio_output_configuration.get_audio_output_configurations

18.4 GET AUDIO OUTPUT CONFIGURATION OPTIONS USING MEDIA2

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

Test Case ID: MEDIA2_AUDIOOUTPUTCONFIGURATION-2

Profile T Normative Reference: Conditional

Feature Under Test: GetAudioOutputConfigurationOptions

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

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with **GetAudioOutputConfigurationOptions** operation for Media2 Service present.
- Device supports Media2 Service (Media2Service).
- Device supports Audio outputs feature for Media2 Service (Media2_AudioOutput).

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

1. Client invokes **GetAudioOutputConfigurationOptions** request message to retrieve an audio output configuration options from 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 **tr2:GetAudioOutputConfigurationOptions** AND
- Device response on the **GetAudioOutputConfigurations** request fulfills the following requirements:

- [S2] It has HTTP 200 response code AND
- [S3] **soapenv:Body** element has child element **tr2:GetAudioOutputConfigurationOptionsResponse**.

FAIL -

- The Client failed PASS criteria.

Validated Feature List:
media2_audio_output_configuration.get_audio_output_configuration_options

18.5 SET AUDIO OUTPUT CONFIGURATION USING MEDIA2

Test Label: Audio Output Configuration - Set Audio Output Configuration

Test Case ID: MEDIA2_AUDIOOUTPUTCONFIGURATION-3

Profile T Normative Reference: Conditional

Feature Under Test: SetAudioOutputConfiguration

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 for Media2 Service present.
- Device supports Media2 Service (Media2Service).
- Device supports Audio outputs feature for Media2 Service (Media2_AudioOutput).

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

1. Client invokes **SetAudioOutputConfiguration** request message to change an 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 **tr2:SetAudioOutputConfiguration** AND
- Device response on the **SetAudioOutputConfiguration** request fulfills the following requirements:
  - [S2] It has HTTP 200 response code AND
  - [S3] **soapenv:Body** element has child element **tr2:SetAudioOutputConfigurationResponse**.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: media2_audio_output_configuration.set_audio_output_configuration

19 Audio Decoder Configuration Using Media2 Test Cases

19.1 Feature Level Normative Reference:

Validated Feature: AudioDecoderConfiguration

Profile T Requirement: Conditional

19.2 Expected Scenarios Under Test:

1. Client connects to Device to modify Audio Decoder Configuration.
2. Client is considered as supporting Audio Decoder Configuration if the following conditions are met:
   - Client is able to retrieve audio decoder configuratoins using **GetAudioDecoderConfigurations** operation (Media2 Service) AND
   - Client is able to retrieve audio decoder configuratoin options using **GetAudioDecoderConfigurationOptions** operation (Media2 Service) AND
   - Client is able to modify audio decoder configuratoin using **SetAudioDecoderConfiguration** operation (Media2 Service) AND
3. Client is considered as NOT supporting Audio Decoder Configuration if ANY of the following is TRUE:
   - No valid response to **GetAudioDecoderConfigurations** request (Media2 Service) OR
   - No valid response to **GetAudioDecoderConfigurationOptions** request (Media2 Service) OR
   - No valid response to **SetAudioDecoderConfiguration** request (Media2 Service) OR

19.3 GET AUDIO DECODER CONFIGURATIONS USING MEDIA2

Test Label: Audio Decoder Configuration - Get Audio Decoder Configurations

Test Case ID: MEDIA2_AUDIODECODERCONFIGURATION-1

Profile T Normative Reference: Conditional

Feature Under Test: GetAudioDecoderConfigurations

Test Purpose: To verify that audio decoder configuration 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 for Media2 Service present.
- Device supports Media2 Service (Media2Service).
- Device supports Audio outputs feature for Media2 Service (Media2_AudioOutput).

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

1. Client invokes **GetAudioDecoderConfigurations** request message to retrieve an audio decoder configuration or a list of 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 **tr2:GetAudioDecoderConfigurations** AND
- Device response on the **GetAudioDecoderConfigurations** request fulfills the following requirements:
  - [S2] It has HTTP 200 response code AND
  - [S3] **soapenv:Body** element has child element **tr2:GetAudioDecoderConfigurationsResponse**.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: media2_audio_decoder_configuration.get_audio_decoder_configurations

19.4 GET AUDIO DECODER CONFIGURATION OPTIONS USING MEDIA2

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

Test Case ID: MEDIA2_AUDIODECODERCONFIGURATION-2

Profile T Normative Reference: Conditional

Feature Under Test: GetAudioDecoderConfigurationOptions

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

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with **GetAudioDecoderConfigurationOptions** operation for Media2 Service present.
- Device supports Media2 Service (Media2Service).
- Device supports Audio outputs feature for Media2 Service (Media2_AudioOutput).

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

1. Client invokes **GetAudioDecoderConfigurationOptions** request message to retrieve an audio decoder configuration options from 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 **tr2:GetAudioDecoderConfigurationOptions** AND
- Device response on the **GetAudioDecoderConfigurationOptions** request fulfills the following requirements:

- [S2] It has HTTP 200 response code AND
- [S3] **soapenv:Body** element has child element **tr2:GetAudioDecoderConfigurationOptionsResponse**.

FAIL -

- The Client failed PASS criteria.

Validated Feature List:
media2_audio_decoder_configuration.get_audio_decoder_configuration_options

19.5 SET AUDIO DECODER CONFIGURATION USING MEDIA2

Test Label: Audio Decoder Configuration - Set Audio Decoder Configuration

Test Case ID: MEDIA2_AUDIODECODERCONFIGURATION-3

Profile T Normative Reference: Conditional

Feature Under Test: SetAudioDecoderConfiguration

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 for Media2 Service present.
- Device supports Media2 Service (Media2Service).
- Device supports Audio outputs feature for Media2 Service (Media2_AudioOutput).

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

1. Client invokes **SetAudioDecoderConfiguration** request message to change an 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 **tr2:SetAudioDecoderConfiguration** AND
- Device response on the **SetAudioDecoderConfiguration** request fulfills the following requirements:
  - [S2] It has HTTP 200 response code AND
  - [S3] **soapenv:Body** element has child element **tr2:SetAudioDecoderConfigurationResponse**.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: media2_audio_decoder_configuration.set_audio_decoder_configuration

20 List Video Source Configurations Using Media2 Test Cases

20.1 Feature Level Normative Reference:

Validated Feature: GetVideoSourceConfigurations

Profile T Requirement: Conditional

20.2 Expected Scenarios Under Test:

1. Client connects to Device to retrieve all Video Source Configurations availible on device.
2. Client is considered as supporting List Video Source Configurations if the following conditions are met:
   - Client is able to retrieve full list of video source configuratoins using **GetVideoSourceConfigurations** operation with skipped ConfigurationToken and ProfileToken in request (Media2 Service).
3. Client is considered as NOT supporting List Video Source Configurations if ANY of the following is TRUE:
   - No valid response to **GetVideoSourceConfigurations** request that does not contain ConfigurationToken and ProfileToken elements (Media2 Service) OR

20.3 LIST VIDEO SOURCE CONFIGURATIONS USING MEDIA2

Test Label: Video Source Configuration - Get Video Source Configurations

Test Case ID: MEDIA2_LISTVIDEOSOURCECONFIGURATIONS-1

Profile T Normative Reference: Conditional

Feature Under Test: Get Video Source Configurations

Test Purpose: To verify that video source configurations provided by Device is received by Client using the **GetVideoSourceConfigurations** operation.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with **GetVideoSourceConfigurations** operation with skipped ConfigurationToken element and with skipped ProfileToken element present.
- Device supports Media2 Service (Media2Service).

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

1. Client invokes **GetVideoSourceConfigurations** request message with skipped ConfigurationToken element and with skipped ProfileToken element to retrieve a list of video source configurations from the Device.
2. Device responds with code HTTP 200 OK and **GetVideoSourceConfigurationsResponse** message.

Test Result:

PASS -

- Client **GetVideoSourceConfigurations** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **GetVideoSourceConfigurations** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **tr2:GetVideoSourceConfigurations** AND
  - [S2] **tr2:GetVideoSourceConfigurations** element does not have child element **tr2:ConfigurationToken** AND
  - [S3] **tr2:GetVideoSourceConfigurations** element does not have child element **tr2:ProfileToken** AND
- Device response on the **GetVideoSourceConfigurations** request fulfills the following requirements:
  - [S4] It has HTTP 200 response code AND
  - [S5] **soapenv:Body** element has child element **tr2:GetVideoSourceConfigurationsResponse**.

FAIL -

- The Client failed PASS criteria.

Validated Feature List:
media2_list_video_source_configurations.list_video_source_configurations

21 OSD Configuration Using Media2 Test Cases

21.1 Feature Level Normative Reference:

Validated Feature: OSD Configuration

Profile T Requirement: Conditional

21.2 Expected Scenarios Under Test:

1. Client connects to Device to list OSDs, create OSD, and modify OSD on the device.
2. Client is considered as supporting OSD Configuration if the following conditions are met:
   - Client supports **List video source configurations** feature AND
   - Client is able to retrieve OSD configuratoins using **GetOSDs** operation (Media2 Service) AND
   - Client is able to create OSD text configurations using **CreateOSD** operation with Type = **Text** (Media2 Service) AND
   - If supported, Client is able to create OSD image configurations using **CreateOSD** operation with Type = **Image** (Media2 Service) AND
   - Client is able to retrieve OSD options using **GetOSDOptions** operation (Media2 Service).
   - Client is able to modify OSD using **SetOSD** operation (Media2 Service).
3. Client is considered as NOT supporting OSD Configuration if ANY of the following is TRUE:
   - Client does not support **List video source configurations** feature OR
   - No valid response to **GetOSDs** request (Media2 Service) OR
   - No valid response to **CreateOSD** operation with Type = **Text** (Media2 Service) OR
   - No valid response to **CreateOSD** operation with Type = **Image** if detected (Media2 Service) OR
   - No valid response to **GetOSDOptions** operation (Media2 Service) OR
   - No valid response to **SetOSD** operation (Media2 Service).

21.3 GET OSD CONFIGURATIONS USING MEDIA2

Test Label: OSD Configuration - Get OSDs

Test Case ID: MEDIA2_OSDCONFIGURATION-1

Profile T Normative Reference: Conditional

Feature Under Test: GetOSDs

Test Purpose: To verify that existing OSD configurations 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 with skipped **OSDToken** element for Media2 Service present.
- Device supports Media2 Service (Media2Service).
- Device supports Media2 OSD feature (Media2_OSD).

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

1. Client invokes **GetOSDs** request message to retrieve OSD configurations 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 **tr2:GetOSDs** AND
  - [S2] **tr2:GetOSDs** element does not contain child element **tr2:OSDToken** 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 **tr2:GetOSDsResponse**.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: media2_osd_configuration.get_osds

21.4 CREATE TEXT OSD USING MEDIA2

Test Label: OSD Configuration - Create OSD

Test Case ID: MEDIA2_OSDCONFIGURATION-2

Profile T Normative Reference: Conditional

Feature Under Test: CreateOSD

Test Purpose: To verify that Client is able to create text OSD using the **CreateOSD** operation with Type value is equal to **Text**

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with **CreateOSD** operation for Media2 Service with Type value is equal to **Text** present.
- Device supports Media2 Service (Media2Service).
- Device supports Media2 OSD feature (Media2_OSD).

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

1. Client invokes **CreateOSD** with Type = **Text** request message to create text OSD on the Device.
2. Device responds with code HTTP 200 OK and **CreateOSDResponse** message.

Test Result:

PASS -

- Client **CreateOSD** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **CreateOSD** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **tr2:CreateOSD** AND
  - [S2] **tr2:CreateOSD/tr2:OSD/tt:Type** element value is equal to **Text** AND

- Device response on the **CreateOSD** request fulfills the following requirements:
  - [S3] It has HTTP 200 response code AND
  - [S4] **soapenv:Body** element has child element **tr2:CreateOSDResponse**.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: media2_osd_configuration.create_text_osd

21.5 CREATE IMAGE OSD USING MEDIA2

Test Label: OSD Configuration - Create OSD

Test Case ID: MEDIA2_OSDCONFIGURATION-3

Profile T Normative Reference: Conditional

Feature Under Test: CreateOSD

Test Purpose: To verify that Client is able to create image OSD using the **CreateOSD** operation with Type value is equal to **Image**

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with **CreateOSD** operation for Media2 Service with Type value is equal to **Image** present.
- Device supports Media2 Service (Media2Service).
- Device supports Media2 OSD feature (Media2_OSD).

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

1. Client invokes **CreateOSD** with Type = **Image** request message to create image OSD on the Device.
2. Device responds with code HTTP 200 OK and **CreateOSDResponse** message.

Test Result:

PASS -

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

- Client **CreateOSD** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **tr2:CreateOSD** AND
  - [S2] **tr2:CreateOSD/tr2:OSD/tt:Type** element value is equal to **Image** AND
- Device response on the **CreateOSD** request fulfills the following requirements:
  - [S3] It has HTTP 200 response code AND
  - [S4] **soapenv:Body** element has child element **tr2:CreateOSDResponse**.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: media2_osd_configuration.create_image_osd

21.6 GET OSD OPTIONS USING MEDIA2

Test Label: OSD Configuration - Get OSD Options

Test Case ID: MEDIA2_OSDCONFIGURATION-4

Profile T Normative Reference: Conditional

Feature Under Test: GetOSDOptions

Test Purpose: To verify that OSD options provided by 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 for Media2 Service present.
- Device supports Media2 Service (Media2Service).
- Device supports Media2 OSD feature (Media2_OSD).

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

1. Client invokes **GetOSDOptions** request message to retrieve an OSD options 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 **tr2:GetOSDOptions** AND
- Device response on the **GetOSDOptions** request fulfills the following requirements:
  - [S2] It has HTTP 200 response code AND
  - [S3] **soapenv:Body** element has child element **tr2:GetOSDOptionsResponse**.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: media2_osd_configuration.get_osd_options

21.7 SET OSD USING MEDIA2

Test Label: OSD Configuration - Set OSD Configuration

Test Case ID: MEDIA2_OSDCONFIGURATION-5

Profile T Normative Reference: Conditional

Feature Under Test: SetOSD

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

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with **SetOSD** operation for Media2 Service present.
- Device supports Media2 Service (Media2Service).
- Device supports Media2 OSD feature (Media2_OSD).

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

1. Client invokes **SetOSD** request message to change an OSD configuration on the Device.
2. 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 **tr2: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 **tr2:SetOSDResponse**.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: media2_osd_configuration.set_osd

22 Get Snapshot Uri Using Media2 Test Cases

22.1 Feature Level Normative Reference:

Validated Feature: GetSnapshotUri

Profile T Requirement: Conditional

22.2 Expected Scenarios Under Test:

1. Client connects to Device to obtain a JPEG snapshot uri from the device using **GetSnapshotUri** operation.
2. Client gets JPEG images from the device using **HTTP GET** sent to the Uri provided by the Device in GetSnapshotUriResponse.
3. Client is considered as supporting Snapshot Uri if the following conditions are met:
   - Client is able to retrieve JPEG snapshot URI using **GetSnapshotUri** operation (Media2 Service) AND
   - Client is able to retrieve JPEG images using **HTTP GET**.
4. Client is considered as NOT supporting JPEG snapshot URI if ANY of the following is TRUE:
   - No valid response to **GetSnapshotUri** request (Media2 Service).
   - No valid response for **HTTP GET** request the Uri provided by the Device in GetSnapshotUriResponse.

22.3 GET SNAPSHOT URI USING MEDIA2

Test Label: SnapshotUri - Get Snapshot Uri

Test Case ID: MEDIA2_GETSNAPSHOTURI-1

Profile T Normative Reference: Conditional

Feature Under Test: Get Snapshot Uri

Test Purpose: To verify that snapshot URI provided by Device is received by Client using the **GetSnapshotUri** operation.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with **GetSnapshotUri** operation present.
- Device supports SnapshotUri feature for Media2 Service (Media2_SnapshotUri).

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

1. Client invokes **GetSnapshotUri** request message to retrieve snapshot Uri from the Device.
2. Device responds with code HTTP 200 OK and **GetSnapshotUriResponse** message.
3. Client invokes **HTTP GET** request to snapshot Uri.
4. Client responds with code HTTP 200 OK.

Test Result:

PASS -

- Client **GetSnapshotUri** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **GetSnapshotUri** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **tr2:GetSnapshotUri** AND
- Device response on the **GetSnapshotUri** request fulfills the following requirements:
  - [S2] It has HTTP 200 response code AND
  - [S3] **soapenv:Body** element has child element **tr2:GetSnapshotUriResponse** AND
- There is **HTTP GET** request in Test Procedure that fulfills the following requirements:
  - [S4] It invoked to address which equal to **tr2:GetSnapshotUriResponse/tr2:Uri** value from the Device response to GetSnapshotUri request AND
  - [S5] It invoked after the Client **GetSnapshotUri** request AND
- Device response on the **HTTP GET** request fulfills the following requirements:
  - [S6] It has **HTTP 200** response code.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: media2_snapshot_uri.get_snapshot_uri